



HAL
open science

Une approche logicielle du traitement de la dyslexie : étude de modèles et applications

Geoffrey Garcia

► **To cite this version:**

Geoffrey Garcia. Une approche logicielle du traitement de la dyslexie : étude de modèles et applications. Médecine humaine et pathologie. Université Blaise Pascal - Clermont-Ferrand II, 2015. Français. NNT : 2015CLF22634 . tel-01330746

HAL Id: tel-01330746

<https://theses.hal.science/tel-01330746>

Submitted on 13 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Blaise Pascal – Clermont-Ferrand II
U.F.R. Sciences et Technologies

N° D.U. : 2634

EDSPIC : 728

Ecole Doctorale N°70 : Sciences Pour l'Ingénieur de Clermont-Ferrand

Thèse

présentée par :

Geoffrey GARCIA

pour obtenir le grade de :

Docteur d'Université

Spécialité : Informatique

soutenue publiquement le 07 décembre 2015

Une approche logicielle du traitement de la dyslexie

Étude de modèles et applications

Directeur de thèse :

M. Vincent BARRA

Professeur

CNRS UMR 6158

Président du jury :

M. David HILL

Professeur

CNRS UMR 6158

Rapporteurs :

M. Olivier BOISSIER

Professeur

CNRS UMR 5516

M. Jean ECALLE

Professeur

Laboratoire EMC - Lyon 2

*Aux mémoires de mon grand-père,
mon arrière grand-mère
et mon oncle.*

Remerciements

Je tiens tout d'abord à remercier Monsieur Olivier Boissier, Professeur à l'École des Mines de Saint-Étienne, et Monsieur Jean Ecalle, Professeur à l'Université Lumière Lyon 2, qui m'ont fait l'honneur d'accepter de rapporter mon travail de thèse. La rédaction de cet imposant manuscrit fut longue et je ne doute pas que sa relecture a dû l'être également.

Je remercie sincèrement Monsieur David Hill, Professeur à L'Université Blaise Pascal à Clermont-Ferrand, pour avoir présidé ce jury. Les nombreux conseils qu'il m'a prodigués pendant et après mes recherches m'ont d'ailleurs été très précieux.

Je remercie aussi chaleureusement mon directeur de thèse, Monsieur Vincent Barra, Professeur à l'Université Blaise Pascal et Directeur de l'Institut Supérieur d'Informatique, de Modélisation et de leurs Applications à Clermont-Ferrand, qui m'a soutenu malgré les multiples difficultés rencontrées pendant mon parcours doctoral.

J'adresse également mes profonds remerciements aux doctorants, Mario Laurent, Anastasiya Shtiliyanova et Franck Dernoncourt, dont les nombreuses collaborations m'ont permis d'avancer significativement et plus efficacement dans mes recherches.

Sans rentrer dans une longue énumération qui me mènerait forcément à oublier quelqu'un, je souhaite témoigner de ma plus grande gratitude à l'ensemble des chercheurs et professionnels de santé qui ont pris le temps de me recevoir et d'échanger avec moi sur la problématique de ma thèse. En particulier, un grand merci aux membres de l'association lyonnaise $E=mc^{dys}$ qui m'ont reçu à plusieurs reprises dans leurs locaux, ainsi qu'à José Alba et Delphin Theallier, de l'entreprise clermontoise 2MAtech, et Dominique Gérard, de l'association Cap'tronic, pour leur aide précieuse sur les problématiques de matériel lié à mon travail.

Un grand merci également aux parents d'enfants dyslexiques qui ont pris la peine de répondre à mon enquête sur la prise en charge de leur(s) enfant(s).

Je souhaite remercier du fond du cœur mes amis proches, Annabelle, Maxime, les deux Nicolas et Sarah, pour leur aide et leur patience dans l'épreuve que fut ces années de thèse. Ce sont dans les périodes difficiles que l'on reconnaît ses vrais amis et vous avez su être à la hauteur.

Pour finir, un immense merci à ma famille et ma belle-famille qui m'ont toujours soutenu et affectueusement encouragé. Particulièrement, ma chère et tendre Anne-Paule sans l'appui de laquelle ce travail n'aurait pu voir le jour. Merci infiniment pour tout !

Résumé

Les troubles neuropsychologiques sont très répandus et posent de réels problèmes de santé publique. En particulier, dans notre société moderne où la communication écrite est omniprésente, la dyslexie peut s'avérer excessivement handicapante. On remarque néanmoins que le diagnostic et la remédiation de cette pathologie restent délicats et manquent d'uniformisation. Ceci semble malheureusement inhérent à la caractérisation clinique par exclusion de la dyslexie, à la multitude de praticiens différents impliqués dans une telle prise en charge ainsi qu'au manque d'objectivité de certaines méthodes existantes.

A ce titre, nous avons décidé d'investiguer les possibilités offertes par l'informatique actuelle pour surmonter ces barrières. Effectivement, nous avons supposé que la démocratisation des systèmes informatiques et leur puissance de calcul pourraient en faire un outil de choix pour pallier les difficultés rencontrées lors de la prise en charge de la dyslexie. Cette recherche nous a ainsi mené à étudier les techniques, aussi bien logicielles que matérielles, pouvant conduire au développement d'un système bon marché et évolutif qui serait capable d'assister un changement bénéfique et progressif des pratiques qui entourent cette pathologie. Avec ce projet, nous nous plaçons définitivement dans un courant innovant au service de la qualité des soins et des aides apportées aux personnes souffrant d'un handicap.

Notre travail a ainsi consisté à identifier différents axes d'amélioration que l'utilisation de l'outil informatique rend possible. Chacun de ces axes a alors pu faire l'objet de recherches exhaustives, de modélisations et de développements de prototypes. Nous avons également réfléchi à la méthodologie à mettre en œuvre pour concevoir un tel système dans sa globalité. En particulier, nos réflexions et ces différents accomplissements nous ont permis de définir un framework logiciel propice à l'implémentation d'une plate-forme logicielle que nous avons appelée la PAMMA. Cette plate-forme devrait théoriquement pouvoir disposer de tous les outils permettant le développement souple et efficace d'applications médicales intégrant des processus métiers. Il est ainsi attendu de ce système qu'il permette le développement d'applications, pour la prise en charge des patients dyslexiques, conduisant à un diagnostic plus rapide et plus précis ainsi qu'à une remédiation plus adaptée et plus efficace.

De notre effort d'innovation ressortent des perspectives encourageantes. Cependant, ce type d'initiative ne peut se concrétiser qu'autour de collaborations pluridisciplinaires disposant de nombreux moyens fonctionnels, techniques et financiers. La constitution d'un tel consortium semble donc être la prochaine étape nécessaire à l'obtention des financements pour réaliser un premier prototype fonctionnel de la PAMMA, ainsi que de premières applications. Des études cliniques pourront être alors menées pour prouver indubitablement l'efficacité d'une telle approche dans le cadre de la prise en charge de la dyslexie, ainsi qu'éventuellement d'autres troubles neuropsychologiques.

Mots clés : modélisation, dyslexie, génie logiciel, intégration de processus métiers, aide à la décision, framework multi-agents, informatique médicale, suivi oculaire, interface multimodale.

Abstract

Neuropsychological disorders are widespread and generate real public health problems. In particular in our modern society, where written communication is ubiquitous, dyslexia can be extremely disabling. Nevertheless we can note that the diagnosis and remediation of this pathology are fastidious and lack of standardization. Unfortunately it seems inherent to the clinical characterization of dyslexia by exclusion, to the multitude of different practitioners involved in such treatment and to the lack of objectivity of some existing methods.

In this respect, we decided to investigate the possibilities offered by modern computing to overcome these barriers. Indeed we have assumed that the democratization of computer systems and their computing power could make of them a perfect tool to alleviate the difficulties encountered in the treatment of dyslexia. This research has led us to study the techniques software as well as hardware, which can conduct to the development of an inexpensive and scalable system able to attend a beneficial and progressive changing of practices in this pathology field. With this project we put ourselves definitely in an innovative stream serving quality of care and aid provided to people with disabilities.

Our work has been identifying different improvement areas that the use of computers enables. Then each of these areas could then be the subject of extensive research, modeling and prototype developments. We also considered the methodology for designing this kind of system as a whole. In particular our thoughts and these accomplishments have allowed us to define a software framework suitable for implementing a software platform that we called the PAMMA. This platform should theoretically have access to all the tools required for the flexible and efficient development of medical applications integrating business processes. In this way it is expected that this system allows the development of applications for caring dyslexic patients thus leading to a faster and more accurate diagnosis and a more appropriate and effective remediation.

Of our innovation efforts emerge encouraging perspectives. However such initiatives can only be achieved within multidisciplinary collaborations with many functional, technical and financial means. Creating such a consortium seems to be the next required step to get a funding necessary for realizing a first functional prototype of the PAMMA, as well as its first applications. Some clinical studies may be conducted to prove undoubtedly the effectiveness of such an approach for treating dyslexia and eventually other neuropsychological disorders.

Key words: modeling, dyslexia, software engineering, integration of business processes, decision support, multi-agent framework, medical informatics, eye tracking, multimodal interface.

Sigles et abréviations

ACL : Agent Communication Language

ACNS TS1 : American Clinical Neurophysiology Society Technical Standard 1

AGR : Agent/Group/Role

AORML : Agent-Object Relationship Modeling Language

API : Application Programming Interface

ARIS : Architecture of Integrated Information Systems

ASEME : Agent SystEms Methodology

AUML : Agent UML

B.A.L.E. : Batterie Analytique de Langage Ecrit

BCI : Brain-Computer Interface

BP4WS : Business Process Execution Language for Web Services

BPMN : Business Process Modeling Notation

CAVE : Computer Assisted Virtual Environment

CEN : Comité Européen de Normalisation

CHU : Centre Hospitalier Universitaire

CIM : Computation-Independent Model

CISAD : Cooperative Information System Analysis and Design

CMS : Content Management System

COOL : Coordination Language

CoOL : Cooperation Language

CPDL : Communication Protocol Description Language

CU : Cas d'Utilisation

DAC : Diagramme d'ACTivité

DCL : Diagramme de CLasse

DCU : Diagramme des Cas d'Utilisation

DET : Diagrammes d'Etat-Transition

DICOM : Digital Imaging and Communications in Medicine

DIP : Designing Interaction Protocols

DLL : Dynamic Link Library

DMP : Dossier Médical Personnel

DSE : Diagrammes de SEquence

CORBA : Common Object Request Broker Architecture

DGPM : DILEMMA Generic Protocol Model

DRC : Dual Route Cascade

EDIFACT : Electronic Data Interchange For Administration, Commerce and Transport

EEG : Électroencéphalographie

EMR : Eletronic Medical Record

ERP : Event-Related Potential

E/S : Entrées/Sorties

ET : Eye-Tracking

EVA : Empan Visuo-Attentionnel

FIPA : Foundation for Intelligent Physical Agents

FC : Fonctions Complémentaires

FP : Fonctions Principales

FTP : File Transfer Protocol

GLARE : GuideLine Acquisition, Representation and Execution

GLIF : GuideLine Interchange Format

GMT : Guideline Markup Tool

GUI : Graphical User Interface

HAS : Haute Autorité de Santé

HMD : Head-Mounted Display

ICE : Integrated Clinical Environment

IHE : Integrating the Healthcare Environment

IHM : Interface Homme-Machine (ou Interaction Homme-Machine)

IHTSDO : International Health Terminology Standards Development Organisation

IL : Interaction Language

IR : InfraRouge

IRM : Imagerie par Résonance Magnétique

JADE : Java Agent DEvelopment framework

KIF : Knowledge Interchange Format

LCC : Lightweight Coordination Calculus

LDAP : Lightweight Directory Access Protocol
KQML : Knowledge Query and Manipulation Language
MA : Multi-Agents
MAS-ML : Multi-Agent Systems Modeling Language
MDA : Model Driven Architecture
MEG : Magnétoencéphalographie
MMN : MisMatch Negativity
MPI : Master Patient Index
MSC : Message Sequence Charts
MTM : Multiple-Trace Memory
MVC : Modèle-Vue-Contrôleur
MVP : Modèle-Vue-Présentateur
OMT : Object Modeling Technique
OO : Orienté(e) Objet
OOSE : Object-Oriented Software Engineering
OSSAD : Office Support System Analysis and Design
PAC : Présentation-Abstraction-Contrôle
PAM : Plateforme d'Applications Médicales
PAMMA : Plateforme d'Applications Médicales Multi-Agents
PDL : Protocol Description Language
PE : Potentiel Evoqué
PEA : Potentiel Evoqué Auditif
PEV : Potentiel Evoqué Visuel
PIM : Platform-Independent Model
PNML : Petri Net Markup Language
POO : Programmation Orientée Objet
PRODIGY : Prescribing RatiOnally with Decision-support In General-practice study
PSM : Platform-Specific Model
PUI : Perceptual User Interface
QI : Quotient Intellectuel
RANSAC : RANdom SAmple Consensus
RBAC : Role-Based Access-Control

RGB : Red Green Blue
RUP : Rational Unified Process
SAGE : Standards-based sharable Active Guideline Environment
SDK : Software Development Kit
SDL : Specification and Description Language
SDM-XML : Study Design Model in XML
SMA : Systèmes Multi-Agents
SOAP : Simple Object Access Protocol
SSL : Secure Sockets Layer
TDA/H : Trouble du Déficit de l'Attention avec Hyperactivité
TSL : Trouble Spécifique du Langage
SI : Système d'Information
SQL : Structured Query Language
TMS : Transcranial Magnetic Stimulation
UI : User Interface
UML : Unified Modeling Language
UP : Unified Process
VR : Virtual Reality
WPDL : Workflow Process Definition Language
WSDL : Web Services Description Language
WYSIWYG : What You See Is What You Get
XML : eXtensible Markup Language
XPDL : XML Process Definition Language
YAWL : Yet Another Workflow Language

Table des matières

Remerciements	I
Résumé	II
Abstract	III
Sigles et abréviations	IV
Table des matières	VIII
Table des figures	XIV
Table des tableaux	XXIII
Introduction	1
La problématique	3
I) La dyslexie	4
A) Aspects théoriques	4
1) La lecture et son apprentissage	4
2) Les différents types de dyslexies développementales.....	7
a) La dyslexie phonologique (ou profonde)	8
b) La dyslexie dyséidétique (ou de surface)	8
c) La dyslexie mixte	8
3) Définition formelle et caractérisation de la dyslexie développementale.....	9
4) La comorbidité.....	13
5) Les possibilités de remédiation.....	14
B) Aspects pratiques	15
1) Au niveau de l'éducation.....	15
a) La détection et le dépistage	15
b) Les aménagements et la compensation en milieu scolaire	17
2) Au niveau des soins médicaux et paramédicaux	18
a) Le diagnostic et la prise en charge	18
b) Les outils de diagnostic et de remédiation	20
C) Les différents problèmes posés	22
II) Les axes d'amélioration technique de la prise en charge	24
A) Le principe général du système	24
B) L'automatisation des tâches de bas niveau	30
1) Garantir la reproductibilité et la qualité des pratiques	31

2)	Recueillir et agréger systématiquement les données.....	32
3)	Proposer de l'aide à la décision.....	34
C)	Le support au travail collaboratif	35
1)	Permettre le télétravail et la télémédecine	37
2)	Faciliter le développement et le déploiement de tests informatisés.....	38
3)	Mutualiser nativement des données multimodales	39
4)	Proposer un outil de prise en charge et de suivi pluridisciplinaire.....	40
5)	Impliquer le patient dans une remédiation centrée autour de lui	41
III)	La division en sous-problèmes	43
IV)	Les objectifs fixés	46
L'ingénierie logicielle		48
I)	Le choix d'un modèle de développement	53
II)	Le choix des modèles d'analyse.....	59
A)	Les interfaces utilisateur	61
1)	L'évolution des interfaces utilisateur.....	62
2)	Les différents types d'interfaces utilisateur modernes	63
a)	Les interfaces graphiques.....	65
b)	Les interfaces perceptuelles.....	65
c)	Les interfaces multimodales	65
3)	L'utilisabilité et les fonctionnalités	66
B)	L'architecture du système	69
1)	Le design pattern Modèle-Vue-Contrôleur.....	69
2)	Les systèmes multi-agents	71
a)	Les différents types d'agents et d'organisations d'agents.....	73
b)	La communication entre agents.....	75
c)	De la microarchitecture à la macro-architecture.....	79
3)	La perspective du méta-système	83
a)	Modéliser les interactions avec les utilisateurs	86
b)	Modéliser la gestion des données	89
C)	La modélisation globale de notre solution.....	92
Les développements		100
I)	L'application de l'UP7: un modèle générique de prise en charge de la dyslexie	102
A)	La modélisation métier	105
1)	L'élaboration du schéma de contexte du domaine	106
2)	L'élaboration du diagramme d'activité.....	109

3)	L'élaboration du diagramme de classe métier.....	116
B)	Les exigences fonctionnelles	121
1)	L'élaboration du diagramme des cas d'utilisation système.....	122
2)	L'élaboration des diagrammes de séquence système	126
3)	L'élaboration du schéma de navigation générale	134
C)	L'analyse des cas d'utilisation	136
1)	L'élaboration du diagramme des cas d'utilisation	137
2)	La description des cas d'utilisation	140
3)	L'élaboration des diagrammes de séquence	150
4)	L'élaboration des diagrammes d'état-transition	174
5)	L'élaboration des interfaces utilisateur	179
6)	L'élaboration des diagrammes de classe	199
D)	La synthèse de l'analyse.....	205
1)	L'élaboration du diagramme de classe récapitulatif.....	206
2)	L'élaboration de la matrice de validation	208
E)	La conception	210
1)	La réalisation des choix techniques	211
2)	L'élaboration des diagrammes de séquence technique	213
3)	L'élaboration des diagrammes de classe techniques	229
4)	L'élaboration du diagramme de paquetage.....	236
F)	L'implémentation et les tests.....	238
1)	L'implémentation	238
2)	Les tests.....	239
II)	L'intégration horizontale : support des processus métiers collaboratifs	241
A)	L'intégration des données.....	242
1)	La sécurité	243
2)	La standardisation	245
B)	L'intégration d'applications	247
C)	L'intégration des processus métiers	249
1)	La spécification.....	252
a)	Le niveau 1 de l'éditeur	253
b)	Le niveau 2 de l'éditeur	256
c)	Le niveau 3 de l'éditeur	259
d)	Les relations entre les niveaux de l'éditeur	266
2)	L'encodage	267

3)	L'exécution	268
a)	Le chargement/instanciation d'une application	269
b)	L'exécution de l'application par les agents	271
c)	Le fonctionnement de services indispensables de la PAMMA	273
d)	La destruction des agents/applications	275
D)	L'utilisation du framework au travers de l'éditeur	276
1)	Le processus de développement d'applications	276
2)	L'architecture détaillée de la PAMMA	280
III)	L'intégration métier verticale : support de la multimodalité	285
A)	Le suivi oculaire	288
1)	Le principe de fonctionnement	288
a)	L'œil et la vision	288
b)	Les techniques existantes	292
1.	<i>Les méthodes de détection basées sur l'apparence</i>	<i>294</i>
2.	<i>Les méthodes de détection basées sur la forme</i>	<i>294</i>
3.	<i>Les méthodes de détection basées sur les caractéristiques oculaires</i>	<i>295</i>
4.	<i>Les méthodes d'estimation du regard</i>	<i>295</i>
2)	Les utilisations possibles	296
a)	Un appareil de mesures pour l'analyse de la perception visuelle	297
b)	Un dispositif d'IHM	299
3)	La conception de notre dispositif de suivi oculaire	300
a)	Un aperçu du fonctionnement général	301
b)	L'acquisition des données	304
1.	<i>La détection de l'œil de l'utilisateur</i>	<i>305</i>
a.	<i>La segmentation en quatre niveaux de gris</i>	<i>307</i>
b.	<i>L'identification de la pupille</i>	<i>309</i>
c.	<i>L'extraction des caractéristiques oculaires</i>	<i>323</i>
2.	<i>La détection de la tête de l'utilisateur</i>	<i>331</i>
3.	<i>L'estimation du regard de l'utilisateur</i>	<i>337</i>
a.	<i>Le calibrage</i>	<i>337</i>
b.	<i>La détermination du point de regard</i>	<i>342</i>
c)	Le traitement des données	343
1.	<i>Le filtrage du bruit</i>	<i>345</i>
2.	<i>L'identification des mouvements oculaires</i>	<i>346</i>
d)	La cartographie, l'analyse et l'interprétation des données	354

4) Vue d'ensemble du processus de suivi oculaire	356
B) L'électroencéphalographie.....	359
1) Le principe de fonctionnement.....	359
a) L'activité électrophysiologique	359
b) Les fondamentaux de la technique	360
2) Les utilisations possibles	362
a) Un appareil d'analyse de l'activité cognitive des dyslexiques	364
b) Un dispositif d'IHM.....	364
3) La conception de notre dispositif d'analyse de données EEG	365
4) Vue d'ensemble du processus de traitement des données EEG	368
C) L'intégration de données multimodales	370
1) La fusion de données.....	370
2) L'aide à la décision	371
L'analyse des résultats et les perspectives.....	375
I) Le bilan des objectifs initialement fixés.....	376
A) L'objectif MODEL1	376
B) L'objectif PROTO1.....	377
C) L'objectif MODEL2	379
D) L'objectif PROTO2.....	379
E) L'objectif PROTO3.....	381
II) Les développements à venir.....	385
A) L'implémentation du noyau de la PAMMA.....	386
B) La conception d'applications pour la PAMMA	390
1) La modélisation des processus métiers	394
2) L'approche ludique et immersive.....	399
C) La réalisation d'un casque « tout-en-un ».....	404
Conclusion	408
Bibliographie	410
Annexes.....	480
I) Les ressources utilisées pour cette recherche	481
II) Enquête auprès des parents d'enfants dyslexiques.....	482
A) Le questionnaire.....	482
B) L'analyse des résultats.....	491
C) Le tableau de recensement.....	515

III)	Éléments des différents prototypes	529
A)	Quelques fichiers d'entête	529
1)	DLL_constants.h	529
2)	IHardwareReader.h	531
3)	EYE_constants.h	531
4)	EYEReaders.h	533
5)	EEG_constants.h.....	534
6)	EEGReader.h.....	536
B)	Les listes de mots du test de lecture de mots fréquents et peu fréquents	537
IV)	Le prototypage de l'interface multimodale	539
A)	Le cahier des charges fonctionnel du suivi oculaire.....	539
B)	Premiers résultats	551

Table des figures

Figure 1 : Schéma du modèle de lecture à double voie (d'après (Coltheart et al., 2001)).....	6
Figure 2 : Représentation des aires du langage (à partir d'une source (Wikimedia, 2015))...	12
Figure 3 : Représentation du parcours de formation des élèves en difficultés scolaires graves et persistantes, avec ou sans handicap (source (M.D.P.H. de Guadeloupe, 2007)).....	16
Figure 4 : Représentation du parcours de soins au sein du réseau de santé Dys/10 (source (E=mc ^{dys} , 2013)).....	19
Figure 5 : Représentation du principe d'intégration verticale et horizontale dans notre contexte d'application.....	25
Figure 6 : Représentation des différents niveaux des processus métiers (à partir de (Ko, 2009))	26
Figure 7 : Représentation de l'intégration à la communication d'un SI dans le domaine de la santé.....	28
Figure 8 : Schéma des éléments et interfaces du modèle de référence des workflows (d'après (Hollingsworth, 1995))	36
Figure 9 : Schéma d'ébauche de l'architecture informatique du dispositif.....	44
Figure 10 : Représentation de la relation entre méta-modèle, modèle et instance de modèle	52
Figure 11 : Schéma d'ensemble de la démarche UP7 (d'après (Gabay & Gabay, 2008)).....	57
Figure 12 : Représentation de la relation entre UP, UP7 et notre processus de développement	58
Figure 13 : Ivan Sutherland utilisant Sketchpad (source (Sutherland, 2003)).....	63
Figure 14 : Schéma général des interactions modales en IHM.....	64
Figure 15 : Diagramme UML du méta-modèle pour représenter des modalités d'IHM (d'après (Obrenović & Starčević, 2004))	68
Figure 16 : Schéma des interactions au sein du modèle MVC.....	70
Figure 17 : Schéma de la microarchitecture du module de communication entre agent.....	78
Figure 18 : Diagramme des classes constituant un agent.....	79
Figure 19 : Schéma de la microarchitecture globale d'un agent	80
Figure 20 : Schéma des interactions au sein du modèle MVC composé d'agents	81
Figure 21 : Diagramme UML du méta-modèle pour représenter des SMA (d'après (Ferber et al., 2004)).....	82
Figure 22 : Schéma des interactions dans un framework composé d'agents reposant sur MVC	84
Figure 23 : Représentation d'agents d'interactions multimodales	87

Figure 24 : Représentation d'agents de notre framework impliqués dans les interactions avec les utilisateurs	88
Figure 25 : Schématisation de la classification des différentes données du framework	90
Figure 26 : Représentation d'agents d'abstractions de données	91
Figure 27 : Représentation d'agents de notre framework impliqués dans la gestion des données du domaine.....	92
Figure 28 : Représentation de la relation entre le langage UML, nos diagrammes UML et notre PAM	93
Figure 29 : Représentation de l'encapsulation des outils proposés.....	93
Figure 30 : Schéma d'ensemble d'un système supportant la modélisation MVC	94
Figure 31 : Schéma d'ensemble d'un système multi-agents supportant la modélisation MVC	95
Figure 32 : Schéma global d'ébauche de l'architecture multi-agents du dispositif.....	97
Figure 33 : Représentation de la constitution d'un réseau de PAMMA.....	98
Figure 34 : Représentation des différents types de documentation accompagnant le logiciel	100
Figure 35 : Représentation de la relation entre le framework de la PAMMA, les workflows et les applications médicales.....	101
Figure 36 : Schéma de la séquence d'activités lors d'une phase d'UP7 (d'après (Gabay & Gabay, 2008))	103
Figure 37 : Schéma de contexte fonctionnel	107
Figure 38 : Schéma de contexte structurel	107
Figure 39 : Diagramme d'environnement des cas d'utilisation	108
Figure 40 : Diagramme d'activité de la partie d'analyse et validation de théorie	110
Figure 41 : Diagramme d'activité du sous-système d'aide diagnostique	111
Figure 42 : Diagramme d'activité dans l'environnement dynamique de remédiation	113
Figure 43 : Représentation (a) d'un utilisateur et (b) d'un utilisateur authentifié	114
Figure 44 : Diagramme de classe des profils d'utilisateurs.....	115
Figure 45 : Diagramme de classe métier (serveur)	118
Figure 46 : Diagramme de classe métier (client)	120
Figure 47 : Diagramme des cas d'utilisation système de la partie d'analyse et validation de théorie.....	123
Figure 48 : Diagramme des cas d'utilisation système du sous-système d'aide diagnostique	124
Figure 49 : Diagramme des cas d'utilisation système dans l'environnement dynamique de remédiation.....	125

Figure 50 : Diagramme de séquence du cas d'utilisation « Définir des processus de soins »	127
Figure 51 : Diagramme de séquence du cas d'utilisation « Définir des épreuves »	127
Figure 52 : Diagramme de séquence du cas d'utilisation « Réaliser un bilan »	128
Figure 53 : Diagramme de séquence du cas d'utilisation « Passer les tests »	129
Figure 54 : Diagramme de séquence du cas d'utilisation « Analyse et conclusion »	130
Figure 55 : Diagramme de séquence du cas d'utilisation « Calibrage de la simulation »	130
Figure 56 : Diagramme de séquence du cas d'utilisation « Planifier le programme de remédiation »	131
Figure 57 : Diagramme de séquence du cas d'utilisation « Suivre le programme »	132
Figure 58 : Diagramme de séquence du cas d'utilisation « Simuler le suivi du programme »	132
Figure 59 : Diagramme de séquence du cas d'utilisation « Analyse et ajustement »	133
Figure 60 : Diagramme de séquence du cas d'utilisation « Recalibrage de la simulation » ..	133
Figure 61 : Schéma de navigation générale du client	134
Figure 62 : Schéma de navigation générale du serveur	135
Figure 63 : Diagramme des cas d'utilisation de gestion des profils	138
Figure 64 : Diagramme des cas d'utilisation des connexions client-serveur	139
Figure 65 : Diagramme de séquence de la précondition aux cas d'utilisation « Installation et activation du système »	151
Figure 66 : Diagramme de séquence de la précondition aux cas d'utilisation « Authentification de l'utilisateur »	152
Figure 67 : Diagramme de séquence de la précondition aux cas d'utilisation « Accéder à un dossier patient »	153
Figure 68 : Diagramme de séquence du cas d'utilisation 1 « Définir des processus de soins »	154
Figure 69 : Diagramme de séquence du cas d'utilisation 2 « Définir des épreuves »	155
Figure 70 : Diagramme de séquence du cas d'utilisation 3 « Réaliser un bilan »	157
Figure 71 : Diagramme de séquence du cas d'utilisation 4 « Passer les tests »	158
Figure 72 : Diagramme de séquence du cas d'utilisation 5 « Analyse et conclusion »	159
Figure 73 : Diagramme de séquence du cas d'utilisation 6 « Calibrage de la simulation » ..	160
Figure 74 : Diagramme de séquence du cas d'utilisation 7 « Planifier le programme de remédiation »	161
Figure 75 : Diagramme de séquence du cas d'utilisation 8 « Suivre le programme »	162
Figure 76 : Diagramme de séquence du cas d'utilisation 9 « Simuler le suivi du programme »	163

Figure 77 : Diagramme de séquence du cas d'utilisation 10 « Analyser les résultats »	164
Figure 78 : Diagramme de séquence du cas d'utilisation 11 « Ajuster la remédiation »	164
Figure 79 : Diagramme de séquence du cas d'utilisation 12 « Recalibrage de la simulation »	165
Figure 80 : Diagramme de séquence de la fonction « Valider une action sur un profil »	166
Figure 81 : Diagramme de séquence du cas d'utilisation 13 « Création d'un profil »	166
Figure 82 : Diagramme de séquence du cas d'utilisation 14 « Modification d'un profil »	167
Figure 83 : Diagramme de séquence du cas d'utilisation 15 « Suppression d'un profil »	168
Figure 84 : Diagramme de séquence du cas d'utilisation 16 « Import/Export d'un profil »	169
Figure 85 : Diagramme de séquence du cas d'utilisation 17 « Connexion/Déconnexion d'un profil »	170
Figure 86 : Diagramme de séquence du cas d'utilisation 18 « Consultation d'un dossier patient »	171
Figure 87 : Diagramme de séquence du cas d'utilisation 19 « Connexion client-serveur »	172
Figure 88 : Diagramme de séquence du cas d'utilisation 20 « Déconnexion client-serveur »	173
Figure 89 : Diagramme d'état-transition d'une PAMMA	175
Figure 90 : Diagramme d'état-transition d'un utilisateur	175
Figure 91 : Diagramme d'état-transition d'un profil	176
Figure 92 : Diagramme d'état-transition d'une connexion entre PAMMA	176
Figure 93 : Diagramme d'état-transition d'un dossier patient	177
Figure 94 : Diagramme d'état-transition d'un modèle de simulation	177
Figure 95 : Diagramme d'état-transition d'une application de la PAMMA	178
Figure 96 : Prototype d'IHM pour l'activation du système	181
Figure 97 : Prototype d'IHM pour la gestion des profils	182
Figure 98 : Prototype d'IHM pour l'utilisation d'un profil	182
Figure 99 : Prototype d'IHM pour la création d'un profil	183
Figure 100 : Prototype d'IHM pour la modification d'un profil	183
Figure 101 : Prototype d'IHM pour la suppression d'un profil	184
Figure 102 : Prototype d'IHM pour l'importation d'un profil	184
Figure 103 : Prototype d'IHM pour l'exportation d'un profil	185
Figure 104 : Prototype d'IHM de l'écran d'accueil du serveur	186
Figure 105 : Prototype d'IHM de l'écran d'accueil du client	186
Figure 106 : Prototype d'IHM de la déconnexion du profil en cours d'utilisation	187
Figure 107 : Prototype d'IHM de la fermeture de l'application	187

Figure 108 : Prototype d'IHM de gestion des connexions entre PAMMA.....	187
Figure 109 : Prototype d'IHM de demande de connexion entre PAMMA	188
Figure 110 : Prototype d'IHM de clôture d'une connexion entre PAMMA.....	189
Figure 111 : Prototype d'IHM pour l'accès à un dossier patient	190
Figure 112 : Prototype d'IHM pour la consultation d'un dossier patient.....	191
Figure 113 : Prototype d'IHM pour la réalisation de bilans.....	192
Figure 114 : Prototype d'IHM pour passer les tests d'un bilan	193
Figure 115 : Prototype d'IHM pour l'analyse et la conclusion d'un bilan.....	194
Figure 116 : Prototype d'IHM pour la planification d'un programme de remédiation.....	195
Figure 117 : Prototype d'IHM pour la réalisation des exercices d'un programme de remédiation.....	196
Figure 118 : Prototype d'IHM pour la simulation des exercices d'un programme de remédiation.....	197
Figure 119 : Prototype d'IHM pour l'analyser des résultats d'un programme de remédiation	198
Figure 120 : Diagramme de classe de la précondition aux cas d'utilisation « Installation et activation du système »	200
Figure 121 : Diagramme de classe de la précondition aux cas d'utilisation « Authentification de l'utilisateur ».....	200
Figure 122 : Diagramme de classe de la précondition aux cas d'utilisation « Accéder à un dossier patient » et du cas d'utilisation 18 « Consultation d'un dossier patient »	201
Figure 123 : Diagramme de classe des cas d'utilisation 1 « Définir des processus de soins » et 2 « Définir des épreuves ».....	201
Figure 124 : Diagramme de classe des cas d'utilisation de 3 « Réaliser un bilan » à 6 « Calibrage de la simulation »	202
Figure 125 : Diagramme de classe des cas d'utilisation de 7 « Planifier le programme de remédiation » à 12 « Recalibrage de la simulation ».....	203
Figure 126 : Diagramme de classe de la précondition aux cas d'utilisation « Valider une action sur un profil » et des cas d'utilisation de 13 « Création d'un profil » à 17 « Connexion/Déconnexion d'un profil »	204
Figure 127 : Diagramme de classe des cas d'utilisation 19 « Connexion client-serveur » et 20 « Déconnexion client-serveur »	204
Figure 128 : Diagramme de classe récapitulatif de l'ensemble du système	207
Figure 129 : Architecture globale du dispositif.....	212
Figure 130 : Diagramme de séquence technique de la précondition aux cas d'utilisation « Installation et activation du système »	214
Figure 131 : Diagramme de séquence technique de la précondition aux cas d'utilisation « Authentification de l'utilisateur »	215

Figure 132 : Diagramme de séquence technique de la précondition aux cas d'utilisation « Accéder à un dossier patient ».....	216
Figure 133 : Diagramme de séquence technique de la précondition aux cas d'utilisation « Valider une action sur un profil ».....	217
Figure 134 : Diagramme de séquence technique de gestion des profils (cas d'utilisation de 13 « Création d'un profil » à 17 « Connexion/Déconnexion d'un profil »).....	218
Figure 135 : Diagramme de séquence technique de l'utilisation de l'accueil du serveur.....	219
Figure 136 : Diagramme de séquence technique de l'utilisation de l'accueil du client.....	220
Figure 137 : Diagramme de séquence technique de l'utilisation de l'écran de gestion des connexions (cas d'utilisation 19 « Connexion client-serveur » et 20 « Déconnexion client-serveur »)	221
Figure 138 : Diagramme de séquence technique de la consultation d'un dossier patient (cas d'utilisation 6 « Calibrage de la simulation », 12 « Recalibrage de la simulation » et 18 « Consultation d'un dossier patient »).....	222
Figure 139 : Diagramme de séquence technique d'utilisation de l'aide diagnostique (cas d'utilisation de 3 « Réaliser un bilan » à 5 « Analyse et conclusion »)	223
Figure 140 : Diagramme de séquence technique d'utilisation du suivi de remédiation (cas d'utilisation de 7 « Planifier le programme de remédiation » à 11 « Ajuster la remédiation »)	224
Figure 141 : Diagramme de séquence technique d'un service de pages blanches.....	226
Figure 142 : Diagramme de séquence technique d'un service de pages jaunes.....	227
Figure 143 : Diagramme de séquence technique d'un médiateur/facilitateur.....	228
Figure 144 : Diagramme de classe technique de la précondition aux cas d'utilisation « Installation et activation du système »	230
Figure 145 : Diagramme de classe technique de la précondition aux cas d'utilisation « Authentification de l'utilisateur »	230
Figure 146 : Diagramme de classe technique de gestion et d'utilisation des profils (précondition aux cas d'utilisation « Valider une action sur un profil » et cas d'utilisation de 13 « Création d'un profil » à 17 « Connexion/Déconnexion d'un profil »).....	231
Figure 147 : Diagramme de classe technique de l'utilisation de l'accueil du serveur.....	232
Figure 148 : Diagramme de classe technique de l'utilisation de l'accueil du client.....	232
Figure 149 : Diagramme de classe technique de gestion des connexions (cas d'utilisation 19 « Connexion client-serveur » et 20 « Déconnexion client-serveur »).....	233
Figure 150 : Diagramme de classe technique de la consultation d'un dossier patient (précondition aux cas d'utilisation « Accéder à un dossier patient », cas d'utilisation 6 « Calibrage de la simulation », 12 « Recalibrage de la simulation » et 18 « Consultation d'un dossier patient »)	233
Figure 151 : Diagramme de classe technique d'utilisation de l'aide diagnostique cas d'utilisation de 3 « Réaliser un bilan » à 5 « Analyse et conclusion »)	234

Figure 152 : Diagramme de classe technique d'utilisation du suivi de remédiation (cas d'utilisation de 7 « Planifier le programme de remédiation » à 11 « Ajuster la remédiation »)	235
Figure 153 : Diagramme de classe technique des services de pages blanches, pages jaunes et médiateur/facilitateur.....	235
Figure 154 : Diagramme de paquetage.....	237
Figure 155 : Schématisation des relations entre organisation, processus métiers et informations (d'après (Hollingsworth, 2004)).....	241
Figure 156 : Représentation d'un modèle d'implémentation de workflows en milieu clinique	251
Figure 157 : Diagramme des types de classes des couches du domaine métier du modèle MVC composé d'agents.....	254
Figure 158 : Notation graphique des différents types d'envoi de messages	255
Figure 159 : Exemple de protocole d'interaction	256
Figure 160 : Exemple d'une organisation d'agents.....	257
Figure 161 : Notations graphiques des agents au niveau 2 de l'éditeur	258
Figure 162 : Notation graphique d'un lien de filiation entre agents	258
Figure 163 : Diagramme des classes composant les processus.....	264
Figure 164 : Schéma des relations entre les trois niveaux conceptuels de l'éditeur	265
Figure 165 : Diagramme d'états-transitions des activités	268
Figure 166 : Ordinogramme du principe d'instanciation des applications.....	270
Figure 167 : Relations entre des entités de niveaux conceptuels différents	271
Figure 168 : Ordinogramme du principe d'exécution d'un agent	272
Figure 169 : Système d'adressage de la PAMMA	274
Figure 170 : Exemple de fragment du tableau noir/blanc des états des protocoles	275
Figure 171 : Représentation du principe d'utilisation du framework	277
Figure 172 : Prototype de l'IHM de l'éditeur	278
Figure 173 : Représentation de l'installation d'un service λ au niveau 3 de l'éditeur	279
Figure 174 : Représentation d'un SMA assurant un service λ au niveau 2 de l'éditeur.....	279
Figure 175 : Représentation de l'enregistrement d'un service λ au niveau 1 de l'éditeur ...	280
Figure 176 : Représentation abrégée d'un service λ et d'un manager de services au niveau 1 de l'éditeur	280
Figure 177 : Prototype de l'IHM du moniteur de workflows.....	282
Figure 178 : Schéma global de l'architecture détaillée de la PAMMA	284
Figure 179 : Schématisation des principaux composants de la partie tangible de notre interface multimodale.....	286

Figure 180 : Schéma d'un globe oculaire humain (source (Wikimedia, 2015)).....	289
Figure 181 : La différence entre l'axe visuel et l'axe optique de l'œil humain (à partir d'une source (Wikimedia, 2015))	290
Figure 182 : Champ visuel humain dans le plan horizontal (source (Wikimedia, 2015))	291
Figure 183 : Diagramme de classe de la description de la modalité de suivi oculaire	297
Figure 184 : Schéma général du dispositif de suivi oculaire et des repères manipulés	302
Figure 185 : Schéma des composants et flux de notre système de suivi du regard.....	303
Figure 186 : La courbe $\Gamma(t)$ (a) en évolution dans \mathbb{R}^2 (b) puis discrétisée dans \mathbb{Z}^2	313
Figure 187 : Enveloppes des voisinages géodésiques d'un point/pixel x (a) 4-connecté et (b) 8-connecté.....	316
Figure 188 : Représentation d'une image numérique sous forme de graphe.....	317
Figure 189 : Représentation (a) des quatre parcours élémentaires d'un point/pixel et (b) des rotations permettant un de ces parcours	319
Figure 190 : Représentation d'une ellipse et de ses paramètres dans le plan (d'après (Ahn & Rauh, 1999))	324
Figure 191 : Représentation de l'estimation du vecteur de regard par simple connaissance des caractéristiques de la tête de l'utilisateur	332
Figure 192 : Photographie de la Creative Interactive Gesture Camera (Senz3D) (source (Creative Technology Ltd., 2015))	333
Figure 193 : Représentation de la position estimée du nasion (en vert) à partir de points de repère (en rouge) (à partir d'une source (Intel, 2015))	334
Figure 194 : Représentation des angles d'Euler utilisés pour le positionnement de la tête de l'utilisateur (source (Intel, 2015)).....	335
Figure 195 : Schéma détaillé de l'obtention de la position et de l'orientation de la tête.....	336
Figure 196 : Schématisation de l'installation de la caméra distante sur l'écran	338
Figure 197 : Schéma des relations entre repères dans le modèle pinhole	339
Figure 198 : Schéma de la détermination géométrique de l'angle α	340
Figure 199 : Schéma de l'ensemble des traitements multi-agents permettant le suivi oculaire	358
Figure 200 : Exemple de tracé mettant en évidence plusieurs composantes des PE (source (Wikipedia, 2015))	360
Figure 201 : Emplacement et nomenclature des électrodes intermédiaires à 10%, tel que normalisé par l'American Clinical Neurophysiology Society (American Clinical Neurophysiology Society, 2015).....	361
Figure 202 : Diagramme de classe de la description de la modalité EEG	363
Figure 203 : Photographie de l'Emotiv Epoc® (source (Emotiv Inc., 2015))	366
Figure 204 : Schéma du processus de traitement des données issues de l'EEG	369

Figure 205 : Schématisation de la classification des critères d'évaluation impliqués dans la prise de décision.....	373
Figure 206 : Schéma de l'architecture logicielle des deux premiers prototypes	377
Figure 207 : Impression d'écran du prototype réalisé dans le cadre de l'objectif PROTO1 ..	378
Figure 208 : Impression d'écran du prototype réalisé dans le cadre de l'objectif PROTO2 ..	381
Figure 209 : Schéma des relations entre objets constitutifs des scènes sous Unity	382
Figure 210 : Impression d'écran du prototype réalisé dans le cadre de l'objectif PROTO3 ..	383
Figure 211 : Schéma de l'architecture logicielle envisagée pour l'intégration des trois prototypes.....	383
Figure 212 : Chronogramme des activités de développement de la méthode Scrum	388
Figure 213 : Détail de l'activité de gestion d'un sprint	389
Figure 214 : Schéma de la conception d'une gestion de processus métiers (d'après (van der Aalst, ter Hofstede, & Weske, 2003)).....	396
Figure 215 : Schématisation du processus d'informatisation des processus cliniques (d'après (Ciccarese et al., 2005))	397
Figure 216 : « Carte du game design » de J. Schell (Schell, 2010)	401
Figure 217 : Représentation du rapport entre contrôle de l'expérimentation en psychologie par rapport à sa validité écologique (d'après (Blascovich et al., 2002; Loomis et al., 1999))	403
Figure 218 : Représentation fonctionnelle du besoin fondamental de l'interface multimodale	404
Figure 219 : Diagramme des interacteurs de l'interface multimodale.....	405
Figure 220 : Les différents niveaux d'abstraction traités dans cette thèse.....	409

Table des tableaux

Tableau 1 : Récapitulatif des problèmes identifiés	23
Tableau 2 : Récapitulatif des objectifs fixés.....	47
Tableau 3 : Relations entre les concepts liés aux langages d'interactions.....	62
Tableau 4 : Les différents types de médiateurs/facilitateurs utilisés pour les SMA	76
Tableau 5 : Description des agents de haut niveau impliqués dans le système	96
Tableau 6 : Fiche Guide 1 pour UP7 (d'après (Gabay & Gabay, 2008)).....	106
Tableau 7 : Fiche Guide 2 pour UP7 (d'après (Gabay & Gabay, 2008)).....	109
Tableau 8 : Liste de valeurs importantes des profils d'utilisateurs	114
Tableau 9 : Fiche Guide 3 pour UP7 (d'après (Gabay & Gabay, 2008)).....	116
Tableau 10 : Fiche Guide 4 pour UP7 (d'après (Gabay & Gabay, 2008)).....	122
Tableau 11 : Fiche Guide 5 pour UP7 (d'après (Gabay & Gabay, 2008)).....	126
Tableau 12 : Fiche Guide 6 pour UP7 (d'après (Gabay & Gabay, 2008)).....	134
Tableau 13 : Fiche Guide 7 pour UP7 (d'après (Gabay & Gabay, 2008)).....	137
Tableau 14 : Fiche Guide 8 pour UP7 (d'après (Gabay & Gabay, 2008)).....	140
Tableau 15 : Fiche Guide 9 pour UP7 (d'après (Gabay & Gabay, 2008)).....	150
Tableau 16 : Fiche Guide 10 pour UP7 (d'après (Gabay & Gabay, 2008)).....	174
Tableau 17 : Fiche Guide 11 pour UP7 (d'après (Gabay & Gabay, 2008)).....	179
Tableau 18 : Liste de caractéristiques d'utilisabilité fonctionnelle et de leurs mécaniques associées (d'après (Juristo et al., 2006))	180
Tableau 19 : Fiche Guide 12 pour UP7 (d'après (Gabay & Gabay, 2008)).....	199
Tableau 20 : Fiche Guide 13 pour UP7 (d'après (Gabay & Gabay, 2008)).....	206
Tableau 21 : Fiche Guide 14 pour UP7 (d'après (Gabay & Gabay, 2008)).....	208
Tableau 22 : Matrice de validation des cas d'utilisation	209
Tableau 23 : Fiche Guide 15 pour UP7 (d'après (Gabay & Gabay, 2008)).....	211
Tableau 24 : Fiche Guide 16 pour UP7 (d'après (Gabay & Gabay, 2008)).....	213
Tableau 25 : Fiche Guide 17 pour UP7 (d'après (Gabay & Gabay, 2008)).....	229
Tableau 26 : Fiche Guide 18 pour UP7 (d'après (Gabay & Gabay, 2008)).....	236
Tableau 27 : Relation entre les couches fonctionnelles du méta-modèle, les niveaux de l'éditeur et les modèles du framework MDA.....	253
Tableau 28 : Exemples d'éditeurs pour les langages de description de workflows présentés	262

Tableau 29 : Types des données gérées par le gestionnaire de protocoles du module de communication	273
Tableau 30 : Types des données gérées par le tableau noir/blanc des pages blanches.....	274
Tableau 31 : Types des données gérées par le tableau noir/blanc des pages jaunes.....	274
Tableau 32 : Critères retenus pour la classification des mouvements oculaires	348
Tableau 33 : Valeurs attendues des fonctions d'état en fonction des mouvements oculaires	349
Tableau 34 : Description de tous les agents impliqués dans le suivi oculaire.....	358
Tableau 35 : Fonctions des interacteurs de l'interface multimodale	406

Introduction

Les troubles cognitifs spécifiques concernent une part importante de la population mondiale. En France, les troubles spécifiques du développement du langage, tels que la dyslexie/dysorthographe touchent environ 8% des personnes. Un diagnostic précis et précoce de ces déficits est donc primordial pour l'efficacité de la prise en charge de ces pathologies. Ainsi l'idée première de ce travail de thèse était d'apporter des outils permettant de diagnostiquer les troubles dyslexiques à l'aide de l'observation puis de l'analyse de l'activité cérébrale des patients, dans le but d'augmenter significativement l'efficacité et la rapidité de l'examen diagnostique.

Dans un autre domaine de la pathologie cérébrale, les maladies neuro-dégénératives entraînent aussi des troubles de la communication ; par exemple, la maladie de Parkinson (environ 100 000 patients en France) génère des troubles majeurs de l'écriture à la fois pour des raisons motrices (diminution du geste d'écriture) et pour des raisons cognitives (déficit de la capacité de « penser » l'action et de l'exprimer par des verbes). Insuffisamment reconnues, ces difficultés de communication vont poser des problèmes croissants de santé publique. Le langage écrit, bien que modifié par la technologie dans son mode de réalisation, le crayon étant remplacé par l'ordinateur, tient une place majeure dans la communication de chaque individu au sein des sociétés occidentales (notamment avec l'utilisation de plus en plus répandue d'Internet). Il est donc primordial pour les personnes souffrant d'un tel handicap de pouvoir bénéficier d'aménagements inclusifs et compensatoires. L'ordinateur tendant de plus en plus à prendre une place centrale dans nos vies, il paraît naturel de vouloir exploiter cette technologie afin de faire bénéficier ces personnes d'une assistance informatisée et personnalisée.

A l'heure actuelle, il existe un certain nombre de solutions de compensation et/ou de rééducation. Certaines de ces méthodes utilisent un support informatique. Ainsi, il existe actuellement des solutions portables permettant d'aider la personne handicapée à s'affranchir des tâches « de bas niveau » de la production du langage écrit, ou tout du moins d'en faciliter la réalisation. Cependant le paramétrage ainsi que l'évaluation de l'efficacité de ces approches se fondent sur l'observation des performances de l'individu ainsi que de son ressenti : l'adaptation de cette remédiation à un type de handicap et à chacune des personnes handicapées nécessite donc une mise au point délicate. Pour tenir compte de cette variabilité, cette aide informatique doit être fondée sur des connaissances de base relatives aux mécanismes cognitifs et neurophysiologiques qui sous-tendent les performances. A terme nous espérons que la solution incluant l'analyse de l'activité cérébrale, que nous allons appliquer aux personnes souffrant de dyslexie, soit adaptable à tout autre déficit neuropsychologique, qu'il soit acquis ou soit d'origine développementale ou dégénérative.

De façon générale, notre projet s'inscrit dans un objectif d'amélioration de la qualité de vie des personnes handicapées, en assistant le diagnostic (à ce titre nous nous appuyons sur des connaissances existantes et considérées comme validées scientifiquement et cliniquement) et en mettant en place une compensation personnalisée et évolutive. Un soin particulier devra donc être apporté au dispositif dans son ensemble afin qu'il réponde aux

besoins, tout en étant une réelle percée aussi bien technologique – car il intégrera les dernières avancées de la recherche – que pratique – puisqu’il suivra complètement le parcours de la personne.

Ainsi, nous souhaitons opter pour un environnement de remédiation multimodal et immersif, car ce genre d’approche a déjà fait ses preuves de part son côté réaliste. Un tel environnement peut par ailleurs facilement être rendu ludique afin d’obtenir un véritable engagement de la personne handicapée, en particulier s’il s’agit d’enfants. L’acquisition et la restitution de contenus multimodaux présentent elles aussi plusieurs avantages certains, comme la possibilité de recoupement/croisement de données, mais aussi la mise en place d’une pédagogie adaptée sollicitant plusieurs sens et mécanismes cognitifs. L’évolution de l’informatique nous permet maintenant de réaliser ces analyses croisées et multivariées quasiment en temps réel. Avec ce projet, nous nous plaçons définitivement dans un courant innovant au service de la qualité des soins et des aides apportées aux personnes souffrant d’un handicap.

Concrètement, nous présentons dans ce manuscrit la méthodologie qui nous a permis de modéliser l’architecture informatique, à la fois souple et robuste, d’un système visant à assister la communication lors de la prise en charge de personnes souffrant de troubles neuropsychologiques tels que la dyslexie. A ce titre, notre système doit pouvoir non-exhaustivement assister le chercheur dans l’élaboration/formalisation de protocoles, servir de système d’information pour le praticien, analyser de façon multimodale les capacités des patients et sous-tendre l’interaction entre praticiens et patients. Dans ce contexte hautement pluridisciplinaire de prise en charge médicale/paramédicale, notre solution a pour but d’offrir une médiation assistée par ordinateur de toutes les étapes rendues critiques par la profonde complexité de la communication humaine. En effet comme l’a dit Montaigne : « la parole est moitié à celui qui parle, moitié à celui qui l’écoute ». Notre système se doit donc d’intégrer ces deux versants de la communication en proposant pour chacun d’eux des éléments de conception fonctionnels et/ou techniques permettant d’aborder une phase de prototypage. Cette démarche d’ingénierie, primordiale dans l’optique d’une future industrialisation, constituera le fil directeur de cette thèse.

Le présent manuscrit va, de la sorte, être composé de quatre parties. Dans la première, intitulée **La problématique**, nous précisons le contexte dans lequel s’inscrit notre projet afin d’identifier précisément les problèmes auxquels la prise en charge de la dyslexie se heurte et quels types de solutions l’informatique moderne pourrait y apporter. Nous explicitons ensuite, dans la partie **L’ingénierie logicielle**, notre méthodologie de travail, aussi bien fonctionnellement que techniquement, dans le but de faire apparaître des méta-modèles pour notre dispositif. La partie **Les développements** présente une modélisation très complète du système et détaille l’implémentation de certains des composants impliqués dans l’interaction directe avec les utilisateurs. **L’analyse des résultats et les perspectives** termine ce manuscrit et propose, notamment, une projection de ce que système pourrait devenir.

La problématique

Le mot « dyslexie » est d'étymologie grecque : le préfixe « dys » vient de « dus » qui induit une notion de difficulté, et « lexie » de « lexis » qui signifie « mot ». La dyslexie désigne ainsi un trouble spécifique de la lecture qui peut s'avérer particulièrement handicapant. L'Organisation Mondiale de la Santé reconnaît d'ailleurs, depuis 1991, la dyslexie comme un trouble du développement des acquisitions scolaires.

Dans ce chapitre nous allons explorer ce qu'est la dyslexie et comment elle perturbe l'activité de lecture. Nous donnerons d'abord une description comportementale et physiologique de cette pathologie puis un aperçu de la prise en charge qui y est proposée. De cette étude se dégagera un certain nombre de problèmes. Ceci nous amènera à identifier des fonctionnalités, sous la forme d'outils informatiques, destinées à leur trouver des solutions concrètes (voir le chapitre **Les axes d'amélioration technique de la prise en charge**). Nous présenterons alors l'ébauche d'un dispositif destiné à rassembler tous ces outils. Finalement nous proposerons un plan d'action visant à évaluer les concepts que ce système intégrera (voir le chapitre **Les objectifs fixés**).

La mise en œuvre de ce plan d'action se fera au travers de la modélisation du système et la réalisation de prototypes dans les parties suivantes de ce document.

I) La dyslexie

Avant tout autre propos, il est important de signaler que nous nous concentrerons ici sur la dyslexie développementale (en particulier en langue française) que nous désignerons simplement par « dyslexie », bien que le terme puisse également désigner les troubles spécifiques de la lecture acquis par lésion cérébrale, appelées plus précisément dyslexies acquises, ou alexies (Ramus, 2012). Dans la présente partie, nous proposons de faire un bref état des lieux des connaissances et pratiques entourant la dyslexie dont nous souhaitons décrire clairement et simplement les caractéristiques. Pour cela nous nous intéresserons d'abord à la pratique de langue qu'est la lecture, dans un premier temps à travers la théorie concernant son acquisition, puis à l'aide de modèles tentant d'expliquer les traitements cognitifs qui sous-tendent cette activité. Ensuite, nous définirons la dyslexie développementale, la caractériserons cliniquement et traiterons de la prise en charge des enfants qui en sont affectés. Pour finir nous discuterons des perspectives et possibilités d'innovations que cet état des lieux laisse entrevoir dans ce domaine.

A) Aspects théoriques

Dans cette section nous allons d'abord présenter la lecture d'un point de vue cognitiviste, pour ensuite donner une définition clinique de la dyslexie. A cette description comportementale de la pathologie nous adjoindrons des éléments relatifs à la neurophysiologie de ce trouble. Pour finir nous évoquerons la comorbidité rencontrée chez les dyslexiques et les possibilités de remédiation offertes à ces derniers.

1) La lecture et son apprentissage

Avant de vouloir traiter de la dyslexie il est important de comprendre le développement normal des capacités de lecture. Pour cela il faut faire appel à la linguistique, c'est-à-dire l'étude scientifique des langues naturelles et du langage. On précisera que le langage est un ensemble discret de symboles dont la combinatoire vise à fournir des étiquettes à chacune des notions concevables par l'esprit humain, et à les organiser logiquement (Habib, Joannette, & Lecours, 2000). Son étude permet d'expliquer la structure, l'évolution et le fonctionnement des langues. Mais qu'est exactement une langue ? Du point de vue des linguistes, il s'agit d'un système de signes vocaux doublement articulés, qui est propre à une communauté linguistique donnée (Martinet, 1969; Mounin, 2004). La langue est ainsi la manifestation concrète du langage et est propre à un groupe d'individus. Elle est ainsi un code commun à plusieurs locuteurs, une abstraction, alors que la parole, que chacun peut utiliser d'une manière personnelle, matérialise cette abstraction. La parole permet la production de discours, des combinaisons d'informations linguistiques et situationnelles. Le discours participe à la communication entre humains, qui s'avère être par nature multimodale (Obrenović & Starčević, 2004; Oviatt, 2007). On remarquera par ailleurs que la maîtrise du langage n'est pas innée et, qu'à ce titre, elle requiert un apprentissage qui, en conditions normales, débute spontanément dès le plus jeune âge. C'est pourquoi il est

primordial d'étudier la dynamique de l'enfant en développement, plutôt que l'enfant plus âgé ou l'adulte (Goswami, 2003; Karmiloff-Smith, 1998). En effet du point de vue neuro-constructiviste (Westermann et al., 2007), la façon qu'a l'enfant de traiter des stimuli environnementaux est susceptible de changer à plusieurs reprises lors de son développement, conduisant à la formation progressive de représentations spécifiques à un domaine (Karmiloff-Smith, 1998). Ce qui est vrai dans le cadre du développement global de l'enfant l'est aussi dans celui plus restreint de l'apprentissage du langage ou de la lecture. De nombreuses études ont d'ailleurs montré que lorsque l'enfant commence à apprendre à lire, son traitement phonologique est changé à jamais, comme le rapportent Goswami, Wang et Cruz en 2010. Fondamentalement la lecture est la compréhension du discours écrit. Les processus auditifs et visuels de bas niveau devraient donc logiquement être impliqués dans la dyslexie (Goswami, 2003). On remarque, au niveau phonologique, que les enfants prennent conscience d'abord des syllabes, puis des amorces et rimes, et finalement des phonèmes d'une façon très similaire quelle que soit la langue (Ziegler & Goswami, 2006). Le décodage phonologique est une condition nécessaire à l'apprentissage de la lecture, qui cependant n'est pas suffisante (Castel, 2008). On peut donc supposer que ces processus de bas niveau sont indispensables et communs à l'acquisition des différentes langues. Goswami et al. en 2010 ont de plus montré que des déficits au niveau de ces traitements étaient partagés par les dyslexiques de différentes langues. Basée sur l'observation de nombreux déficits, la lecture est ainsi l'objet de plusieurs modèles cognitivistes tentant d'en décrire les processus de bas niveau. De facto ces modèles tentent de mettre en évidence des traitements neuropsychologiques susceptibles d'être lésés et de provoquer un déficit. Afin de mieux comprendre quelles mécaniques cognitives sont impliquées, nous allons maintenant nous pencher sur deux modèles de l'activité de lecture communément mis en avant dans la littérature. On pourra d'ailleurs noter que c'est en bruitant certains traitements (ou certaines connexions entre traitements) de ces modèles, qu'on tente de simuler les comportements pathologiques communément observés (Coltheart, Rastle, Perry, Langdon, & Ziegler, 2001; Cresswell, Monteith-Hodge, & Winfield, 1997).

Le premier modèle que nous allons considérer pose le postulat que le cerveau humain peut suivre deux cheminements différents pour traiter l'information lors de l'activité de lecture. Ces deux manières de lire sont appelées « voies de lecture ». La **Figure 1**, issue d'un article de Coltheart, Rastle, Perry, Langdon et Ziegler (2001), représente un modèle avec ces deux voies de lecture appelé « **Dual Route Cascade** ». La première voie possible, à gauche, correspond à la reconnaissance globale (aussi appelée par adressage ou encore, lexicale). Dans ce premier cas, les mots sont reconnus dans leur globalité et retrouvés directement dans le lexique mental où leur sens et leur phonétique leur sont associés. La deuxième voie, à droite, correspond à la voie d'assemblage (phonologique ou analytique). Dans ce deuxième cas, les mots sont décodés, syllabe par syllabe et sont convertis directement de la forme écrite à la forme phonétique grâce aux correspondances entre graphèmes et phonèmes. En condition de lecture les processus des deux voies susnommées seraient activés de façon concurrente (Ziegler, Perry, & Coltheart, 2003). Ainsi l'action de lecture serait assurée par la voie au travers de laquelle le traitement serait le plus rapide. L'idée principale de cette approche est d'obtenir des données sur les sous-composantes de la lecture et de perturber le modèle de différentes manières afin de refléter les déficits retrouvés chez les dyslexiques. Bien que datant de quelques années, le DRC est assez populaire et les processus cognitifs qu'il s'emploie à mettre en évidence semblent cohérents avec les déficits de lecture

observés (Ziegler, 2011). Pour preuve de l'intérêt que présente cette modélisation à double adressage, des versions améliorées, affinées, ont vu le jour depuis la publication du DRC originel (Perry, Ziegler, & Zorzi, 2007, 2010; Zorzi, 2010). Chez les personnes dyslexiques, l'utilisation d'une de ces voies, voire des deux, est supposée défaillante. Dans ce cas, y avoir recours devrait devenir très difficile ou entraîner des erreurs, permettant de la sorte d'expliquer les difficultés à réaliser des tâches de lecture. D'une manière générale, selon ce modèle, des difficultés à utiliser au moins une des voies de lecture vont se traduire par des troubles de l'apprentissage de la lecture.

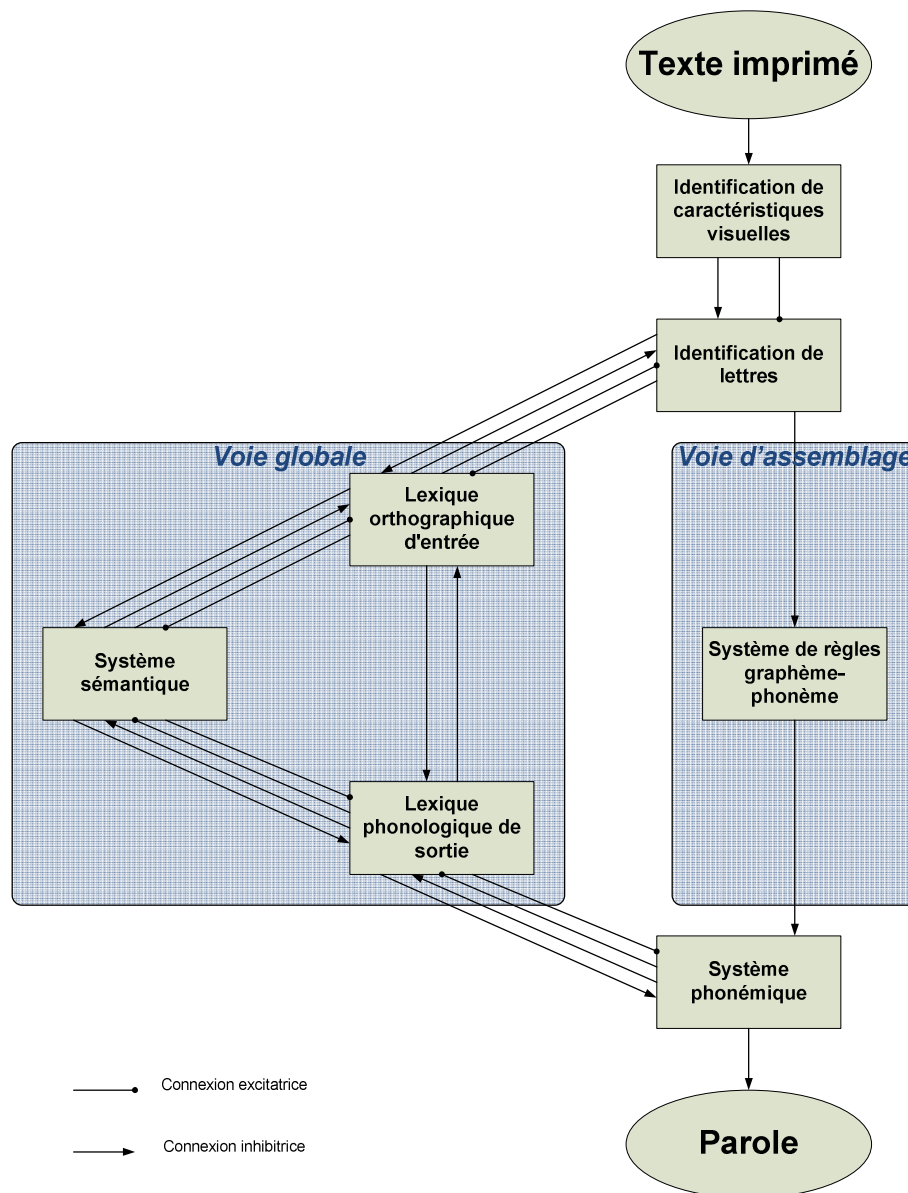


Figure 1 : Schéma du modèle de lecture à double voie (d'après (Coltheart et al., 2001))

Le second modèle que nous voulons traiter met en évidence un autre facteur pouvant jouer un rôle dans la lecture normale et son apprentissage : l'Empan Visuo-Attentionnel (Bosse & Valdois, 2009; Valdois, 2008, 2010). L'EVA correspond au nombre d'éléments visuels distincts qui peuvent être cognitivement traités en parallèle dans une configuration donnée. Il devrait donc occuper une place importante dans la lecture en particulier dans le cadre la

procédure de reconnaissance globale de mots (Valdois, 2010). En effet, la réussite de cette procédure semble logiquement être conditionnée par le nombre de lettres, de caractères, que le lecteur va pouvoir récupérer en une seule fixation. Le modèle destiné à représenter ces processus cognitifs de bas niveau est nommé « Multiple-Trace Memory Model » (Ans, Carbonnel, & Valdois, 1998). Il s'agit d'un modèle hybride qui postule l'existence de deux procédures de lecture, comme pour le DRC (Coltheart et al., 2001), qui cependant ne posséderait pas de composant de conversion graphème-phonème propre au traitement des pseudo-mots. On précisera que ces derniers sont des chaînes de caractères ressemblant à des mots réels mais ne portant aucun sens. Il s'appuie sur le fait qu'un composant visuo-attentionnel jouerait un rôle majeur dans la lecture experte et l'apprentissage de la lecture. Ainsi, un trouble de l'EVA va engendrer une perturbation de la composante orthographique de la lecture et peut être rencontré chez les enfants atteints de dyslexie de surface (Valdois, 2008, 2010). Pour procéder au diagnostic des troubles de la lecture, on notera que des méthodes destinées aux cliniciens intègrent d'ores et déjà ces considérations (voir par exemple (Jacquier-Roux, Lequette, Pouget, Valdois, & Zorman, 2010)).

De tels modèles n'apportent aucune solution concrète au problème des déficits en lecture, mais ils permettent, au travers de simulations, de mieux comprendre les processus impliqués dans la lecture et son apprentissage (Snowling, 2001; Ziegler, 2011; Ziegler et al., 2008). Il en découle que les compétences potentiellement impliquées par ces processus cognitifs peuvent faire l'objet d'évaluations. En pratique l'évaluation du niveau de lecture se base classiquement sur deux critères : la précision (pouvant révéler un faible niveau de décodage) et la vitesse de lecture (Goswami, 2003). C'est en s'appuyant sur ces deux indicateurs que va être décelé un déficit en lecture. Cependant la lecture est un processus complexe qui n'est peut-être pas forcément aussi simple à caractériser (Ziegler, 2011). Les troubles engendrés par un tel déficit sont fréquemment qualifiés de troubles dyslexiques. Afin de mieux appréhender les difficultés que la caractérisation de la dyslexie représente, nous allons maintenant décrire cette pathologie d'un point de vue comportemental.

2) Les différents types de dyslexies développementales

Quand on parle de la dyslexie, il serait souvent plus correct de parler des dyslexies. En effet, de tels déficits ne découlent pas tous d'un trouble cognitif unique, et à ce titre, ne se manifestent pas par les mêmes divergences autant d'un point de vue comportemental (INSERM (dir.), 2007) que physiologique (Ramus, 2012). On retrouve ainsi la description des différents types de dyslexies, résultant des dysfonctionnements des différentes voies de lecture décrites plus haut, dans la plupart des articles scientifiques traitant des généralités entourant la dyslexie (Castel, 2008; Delamare, 2012; Démonet, Taylor, & Chaix, 2004; INSERM (dir.), 2007; Jucla, 2009). Ainsi sont reconnues la dyslexie phonologique, la dyslexie de surface et la dyslexie mixte. Ces sous-types sont classiquement identifiés par l'observation des déficits lors de la lecture de mots irréguliers et de non-mots (Castles & Coltheart, 1993). A titre informatif on précisera que les mots réguliers sont ceux respectant strictement les règles de conversion graphème-phonème alors que les mots irréguliers ne les suivent pas. Les non-mots (à ne pas confondre avec les pseudo-mots) ne respectent aucune règles phonologiques, sont assimilés à des suites de lettres imprononçables et ne portent pas de sens.

a) La dyslexie phonologique (ou profonde)

En lecture la voie la plus souvent touchée est la voie d'assemblage, il en résulte un type de dyslexie appelé dyslexie phonologique. Au niveau comportemental cela se traduit par une difficulté à décoder les mots en utilisant le système de conversion entre graphèmes et phonèmes. Les dyslexiques phonologiques ont donc beaucoup de mal à lire les pseudo-mots. Ils ont également des difficultés avec les mots irréguliers, dont la correspondance graphème/phonème ne respecte pas les standards habituels. En revanche, ils peuvent tout à fait reconnaître des mots de façon globale et ont ainsi peu de problèmes avec les mots réguliers et les mots courants. Ce déficit se manifeste de façon similaire dans des langues aux systèmes orthographiques différents (Goswami et al., 2010). En pratique l'évaluation de la conscience phonologique est le prédicteur le plus précis du futur niveau de lecture et d'orthographe (Goswami, 2003).

b) La dyslexie dyséidétique (ou de surface)

Un peu moins répandue que la dyslexie phonologique, la dyslexie de surface désigne les personnes ayant des difficultés à utiliser la reconnaissance globale (Bosse & Valdois, 2009). Ce trouble pourrait être causé par un déficit de traitement simultané (Lassus-Sangosse, N'guyen-Morel, & Valdois, 2008). Cela entraîne plusieurs problèmes différents de ceux évoqués précédemment, notamment un défaut de prononciation des mots irréguliers. Effectivement, les personnes touchées par cette forme de dyslexie auront tendance à « régulariser » les mots, c'est-à-dire à les prononcer en suivant la conversion graphème/phonème standard. Lors d'une tâche de lecture, des mouvements oculaires anormaux, en particulier les fixations, semblent relever d'un tel déficit (Prado, Dubois, & Valdois, 2007).

c) La dyslexie mixte

Enfin, le troisième type de dyslexie reconnue est appelé dyslexie mixte. Dans le cas de la dyslexie mixte, les deux voies de lecture sont touchées. Les symptômes et difficultés qu'on retrouve dans la dyslexie phonologique et dans la dyslexie de surface sont donc réunis. Il n'est d'ailleurs pas clairement établi si les difficultés phonologiques rencontrées sont une cause ou une conséquence du déficit observé en lecture (Vidyasagar & Pammer, 2010). Au-delà de la conscience phonologique, les dyslexiques rencontrent au moins des difficultés avec la dénomination rapide (des images, des couleurs, des chiffres, des lettres)(Castel, Pech-Georgel, George, & Ziegler, 2008) et la mémoire verbale à court terme, ne pouvant ni l'une ni l'autre être reliées à la lecture (Ramus, 2003).

Quel que soit le type de dyslexie, il existe plusieurs niveaux de déficit et les symptômes peuvent donc être plus ou moins sévères (Snowling, 2001). Ils peuvent aller d'une simple gêne sans grande conséquence, qui n'est parfois pas repérée par la personne atteinte ou son entourage, jusqu'à la quasi-impossibilité de communiquer par écrit. Mais, dans la majorité des cas, on constate une lecture lente et saccadée nécessitant un effort (Goswami et al., 2010). De plus, ces erreurs parasitent la compréhension de textes, phénomène amplifié

lorsque la longueur et la complexité de ces derniers augmentent, et finalement, l'extraction du sens ne se fait que partiellement. Il n'en reste pas moins qu'il y a une hétérogénéité remarquable dans la dyslexie, étant donné que la détérioration des capacités est associée à une multitude de déficits et de combinaison de déficits différents (Ziegler, 2011). On peut même préciser que cette classification communément acceptée ne permet pas de tenir compte de toutes les différences individuelles qu'on observe chez les enfants dyslexiques (Castles & Coltheart, 1993). Ceci se ressent d'ailleurs aux travers des critères cliniques utilisés pour caractériser cette pathologie.

3) Définition formelle et caractérisation de la dyslexie développementale

La définition de la dyslexie communément admise dans la littérature scientifique concerne le déficit comportemental. En effet, un enfant est considéré dyslexique si :

- Il n'a pas de déficit intellectuel global (QI supérieur à 70).
- Il n'a pas de trouble sensoriel (visuel ou auditif) facilement repérable.
- Ses difficultés en lecture ne peuvent pas être expliquées par son contexte social ou psychoaffectif.
- Sa scolarisation est adéquate.
- Il obtient, malgré les points précédents, une note d'au moins 2 écarts-types en dessous du niveau attendu à son âge, à une épreuve standardisée de lecture.

Il s'agit d'une définition clinique que l'on retrouve dans les deux classifications médicales de référence que sont la classification internationale ICD-10, sous l'étiquette *F81.0 Trouble spécifique de la lecture* (World Health Organization, 2011) et la classification américaine DSM-IV (American Psychological Association, 2000). En France, aucune définition officielle de la dyslexie n'est donnée, et il faut se contenter d'un renvoi à la classification internationale (Misès, 2012). Cette pathologie y apparaît comme un trouble spécifique de l'apprentissage de la lecture, lié à une difficulté particulière à identifier les lettres, les syllabes ou les mots. D'autre part, il est important de signaler que la dyslexie semble bénéficier d'une composante génétique avérée (Ramus, 2010). Cependant, on remarquera que ces définitions sont très, voire trop, ouvertes. Effectivement elles se basent sur un diagnostic différentiel, par critères d'exclusion, et ont pour effet de rassembler des troubles hétérogènes derrière l'étiquette « dyslexie ». Cette forte hétérogénéité est souvent illustrée par les chercheurs lorsqu'ils procèdent à des études de cas, cependant elle est fréquemment ignorée par les études de groupes. Le diagnostic est du reste établi au moyen d'un bilan pluridisciplinaire mené par un professionnel médical et divers acteurs appartenant à des professions paramédicales. Ce bilan permet d'éliminer les autres causes et de caractériser le type de dyslexie afin d'y apporter les réponses les mieux adaptées en termes d'aménagement scolaire, prise en charge et remédiation.

Pour le moment, il n'existe aucun examen paraclinique permettant d'aider à établir précisément le diagnostic, même s'il semble que certaines méthodes d'imagerie médicale fonctionnelle pourraient apporter des éléments de réponse (Habib & Robichon, 1996). Le problème de précision du diagnostic rend difficile la prise en charge des personnes touchées mais a aussi une incidence sur les protocoles expérimentaux d'étude de la dyslexie par les chercheurs (Goswami, 2003). Par exemple, certains vont considérer qu'une personne est

atteinte de dyslexie si cette dernière présente un retard de lecture de 1.5 écarts-types en dessous du seuil des normo-lecteurs tandis que d'autres vont considérer un minimum de 2 écarts-types (INSERM (dir.), 2007; Jacquier-Roux et al., 2010). Ces critères basés sur l'écart-type lors de tests linguistiques ne sont pas nécessairement discriminants et peuvent mener à des faux-positifs ou faux-négatifs (Asikainen, 2005). D'autre part cette différence vient du fait que les épreuves d'évaluation du niveau de lecture peuvent être différentes selon l'étude et donner des résultats divergents (Jacquier-Roux et al., 2010). La pertinence de ces résultats peut alors être discutable, selon le sexe, l'âge, la langue de l'individu ou même le choix des groupes témoins ainsi que l'environnement d'expérimentation (Alm, 2004; Goswami, 2003). L'âge semble en effet jouer un rôle majeur pour discuter de telles pathologies, car la chronologie du développement est indispensable à la compréhension des troubles développementaux (Karmiloff-Smith, 1998). La cardinalité des panels utilisés pour les étalonnages de ces divers tests peut être elle aussi critiquable d'un point de vue statistique. D'autre part, certains critères sur lesquels s'appuie la définition de la pathologie sont ambigus. Par exemple, le rôle du QI reste flou, et certaines études montrent que les capacités en lecture ne peuvent y être reliées (Tanaka et al., 2011). Ainsi, on peut trouver des lecteurs atteints de dyslexie avec un faible QI ou un QI élevé et ce critère ne serait pas pertinent afin d'établir le diagnostic. Le rôle à attribuer au milieu socioculturel pourrait, quant à lui, être mieux caractérisé. On peut même ajouter que ce manque de précision du diagnostic différentiel a un effet néfaste sur l'évaluation de la population atteinte. Le pourcentage de personnes touchées par la dyslexie développementale, en langue française, serait de 5 à 8% selon le rapport d'expertise collectif mené par l'INSERM (2007). Il s'agit du chiffre le plus fiable à ce jour, compte tenu du nombre d'études et de chercheurs impliqués dans ce rapport, mais la proportion de la population atteinte pourrait être bien plus élevée. Au niveau international, on voit apparaître de grandes disparités au niveau du pourcentage d'enfants, d'une même classe d'âge et sujets à la dyslexie, selon les pays. Deux phénomènes apportent une première explication à cette différence entre les langues :

- La complexité de conversion graphème-phonème n'est pas la même selon la langue, on parle ainsi de profondeur orthographique des langues (Seymour, Aro, & Erskine, 2003). Comme le soulignent Ziegler, Perry, Ma-Wyatt, Ladner et Schulte-Körne (2003), l'anglais est une des langues européennes où cette conversion est la plus difficile. A contrario, les enfants dyslexiques pratiquant des langues transparentes (comme le finnois et l'italien) lisent le plus souvent sans erreur et leur trouble se manifeste essentiellement par une lenteur de lecture. D'une façon générale, les caractéristiques spécifiques des systèmes phonologiques peuvent varier d'une langue à l'autre, bien que la segmentation syllabique du discours soit liée à la rythmique et aux traits prosodiques (Goswami et al., 2010). D'ailleurs, le traitement rythmique apparaît comme la première stratégie de segmentation du discours mise en place par les enfants, et le développement de la conscience phonologique suit un enchaînement très similaire dans toutes les langues. Ceci n'est pas étonnant quand on sait que cette conscience rythmique dépend d'une information sensorielle commune à toutes langues, contrairement à d'autres caractéristiques phonologiques telles que le pitch – l'intonation – ou la longueur d'un phonème (Goswami et al., 2010). On peut aussi noter, qu'autour de l'âge de 10 ans, les différences du développement de la lecture relatives à l'orthographe ont pour la plupart disparu. Seules celles liées au sexe et à la culture subsistent (Goswami, 2003).

- Le système de représentation écrit a une influence sur les processus impliqués dans l'activité de lecture. En effet, si l'on s'intéresse par exemple à la langue chinoise, la lecture ne se fait plus par une conversion de graphèmes à phonèmes mais par une conversion directe des caractères, d'idéogrammes (qui représentent des mots), aux sens (Siok, Perfetti, Jin, & Tan, 2004). Ainsi, les personnes chez qui ce processus de conversion serait lésé seront considérées comme dyslexiques en langue chinoise. Paradoxalement, on peut avoir donc un même trouble du langage du point de vue comportemental avec deux origines physiologiques différentes.

Devant ces constats d'approximations et de disparités, il paraît essentiel de déterminer une classification plus précise, homogène et commune aux cliniciens, professionnels de la santé (orthophonistes, psychologues, etc.) et aux chercheurs, afin de reconsidérer les critères de diagnostic (Asikainen, 2005; Habib & Joly-Pottuz, 2008). On remarquera que des travaux allant dans ce sens ont d'ailleurs déjà été entrepris (Jacquier-Roux et al., 2010), mais il est pour l'instant impossible de diagnostiquer rapidement, et sans ambiguïté, la dyslexie par l'analyse comportementale. En effet, les mécanismes comportementaux impliqués dans la lecture, et a fortiori dans la dyslexie, sont nombreux (Pammer, 2014). Il peut s'agir de traitements sensoriels, de traitements phonologiques et orthographiques et de codages sémantiques. Le diagnostic consiste alors à investiguer du côté de ces différents traitements cognitifs. D'autre part, la difficulté dans l'établissement du diagnostic vient en grande partie du fait que les déficits engendrés par la dyslexie semblent prendre leurs origines à différents niveaux physiologiques. Afin de tenter d'y voir plus clair, il convient de s'informer sur ce que la caractérisation biologique de cette pathologie peut nous apprendre, et nous fournir comme éléments pouvant conduire au diagnostic.

Au déficit comportemental caractérisant la dyslexie s'ajoute un déficit physiologique. Ces anomalies, qu'elles soient microscopiques ou macroscopiques, plaident en faveur de mécanismes neurobiologiques à l'origine de cette pathologie (voir par exemple (Rimrodt, Peterson, Denckla, Kaufmann, & Cutting, 2010)). En effet, il a été montré que la dyslexie résulte d'un dysfonctionnement cognitif sous-tendu par un dysfonctionnement cérébral (Démonet et al., 2004; Habib, 2000). La possibilité de repérer ce déficit permet de compléter l'étude comportementale en ajoutant de nouveaux indicateurs pour permettre le diagnostic de la dyslexie. On peut notamment détecter les dysfonctionnements cérébraux grâce à l'IRM et la MEG (INSERM (dir.), 2007). Mais avant cela, il convient d'en savoir plus concernant les régions du cortex impliquées dans le traitement du langage. Les premiers chercheurs à s'être penchés sur la question sont Broca (Broca, 1865), Wernicke et, dans le cas particulier de la lecture, Dejerine (Dejerine, 1891). Leurs recherches respectives, s'appuyant sur les cas d'individus cérébraux lésés, ont conduit à l'identification d'aires du cortex participant au langage. Ces régions, situées autour de la scissure de Sylvius, apparaissent sur la **Figure 2**. Mais il existe d'autres structures corticales et sous-corticales, organisées en réseau intra et inter-hémisphérique, qui contribuent au langage (Habib et al., 2000). Norman Geschwind (Galaburda, 1985) fut un des premiers chercheurs à s'intéresser à la dominance cérébrale et la spécialisation des hémisphères (Geschwind, 2010) : il se pencha, entre autres, sur le cas de la dyslexie. En effet, l'asymétrie corticale semble avoir joué un rôle majeur dans l'acquisition et le développement du langage (Habib et al., 2000).

A partir des observations par IRM (Leonard et al., 1993) et d'autopsies, visant à l'étude microscopique des aires du langage, une nouvelle théorie a vu le jour sur l'origine de la

dyslexie. Celle-ci pourrait être entraînée par une dysplasie corticale, c'est-à-dire que la migration des neurones utiles pour les tâches de lecture aurait été incomplète au cours de la corticogenèse de l'enfant, provoquant des ectopies neuronales (Galaburda, Sherman, Rosen, Aboitiz, & Geschwind, 1985; Galaburda, 1993; INSERM (dir.), 2007; Richlan, Kronbichler, & Wimmer, 2012). Il a d'ailleurs été mis en évidence, par des études en Potentiels Evoqués, d'autres anomalies au niveau sous-cortical, sur les noyaux géniculés latéraux dans leur portion magnocellulaire qui est sensible aux informations visuelles rapides et de bas contraste (Livingstone, Rosen, Drislane, & Galaburda, 1991). Ces cellules apparaissent atrophiques, contrairement aux parvocellules véhiculant des informations à transmission lente mais liées à la vision fine, qui restent intactes (Galaburda, 1993). Or, chez les dyslexiques, on constate des perturbations dans la perception de certains stimuli visuels et auditifs, notamment ceux se succédant rapidement et à bas contraste. Ces constatations tendent à confirmer l'hypothèse selon laquelle les dyslexiques ont des difficultés à traiter les informations qui se succèdent rapidement par la voie magnocellulaire (Stein, 2001).

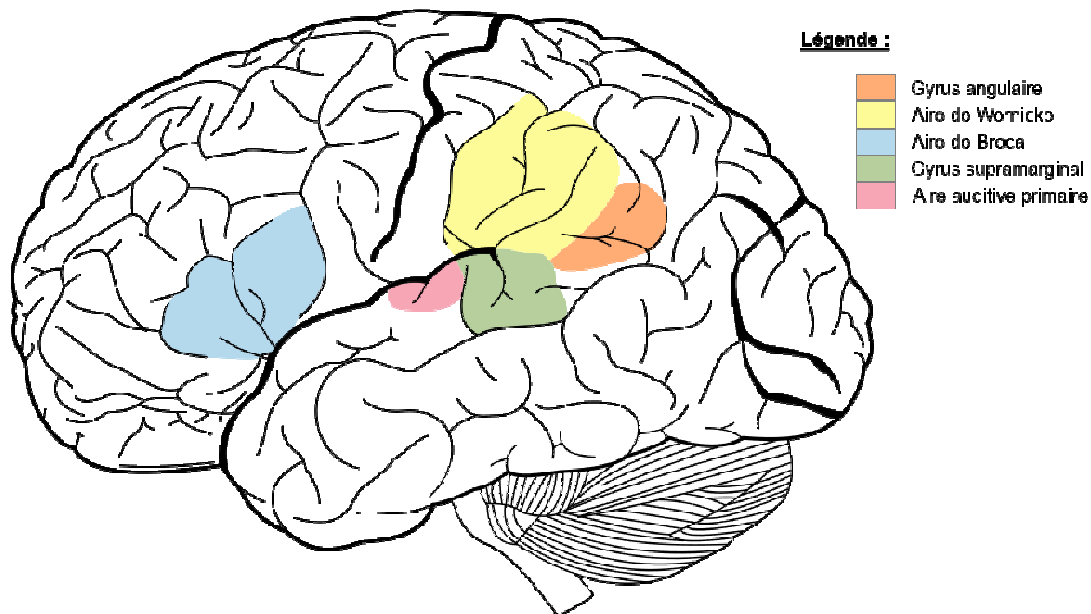


Figure 2 : Représentation des aires du langage (à partir d'une source (Wikimedia, 2015))

D'autres études ont aussi décelé des anomalies macroscopiques liées à la dyslexie. En effet, les sujets atteints de dyslexie exhibent une réduction de l'asymétrie normale entre les deux hémisphères (Galaburda, 1993; Larsen, Høien, Lundberg, & Odegaard, 1990). Cette asymétrie devrait normalement être en faveur du côté gauche du planum temporal (Geschwind & Levitsky, 1968). Comme nous l'avons énoncé précédemment, des aires cérébrales liées au langage, telles que l'aire de Wernicke (**Figure 2**), siègent dans cette zone du cortex. Chez les dyslexiques, le planum droit anormalement volumineux peut laisser penser à un défaut d'élimination de neurones en excès durant la corticogenèse. D'autre part, la communication inter-hémisphérique semble d'ailleurs aussi être affectée. En effet, le corps calleux, la masse de substance blanche assurant le transfert de l'information entre les deux hémisphères, semble modifié dans sa structure par la dyslexie (Robichon & Habib, 1998). Ces anomalies tendent à faire penser que les connexions, assurées par des faisceaux de fibres nerveuses, sont moins bien organisées chez les sujets atteints de dyslexie.

De ces différentes recherches, il ressort que la dyslexie s'exprime à un niveau structurel et à un niveau fonctionnel, c'est-à-dire, respectivement, neuro-anatomiquement et neuro-physiologiquement. Ces modifications de la structure et du fonctionnement de l'encéphale devraient pouvoir être, comme décrit par de nombreuses études, corrélées avec les troubles observés au travers d'analyse par imagerie médicale. En particulier, dans le cadre d'un examen physiologique, l'analyse fonctionnelle par électroencéphalographie (EEG) ou magnétoencéphalographie (MEG) permet de mesurer ce que l'on appelle les potentiels évoqués. Ces derniers sont supposés refléter les variations locales de l'activité corticale. Nous reviendrons ultérieurement, et plus en détail, sur ces PE et expliquerons l'intérêt qu'ils présentent dans l'étude de la dyslexie (voir la section **L'électroencéphalographie**). Pour finir, on remarquera cependant qu'à l'heure actuelle personne n'est capable de dire si les modifications encéphaliques observées sont une cause ou une conséquence de la dyslexie (Habib & Joly-Pottuz, 2008).

4) La comorbidité

La comorbidité désigne la présence simultanée de plusieurs diagnostics. Ceci n'implique pas nécessairement la présence de multiples troubles (ou maladies dans un cadre plus général), mais l'impossibilité d'émettre un seul diagnostic pour un patient donné. Dans le cas d'une personne diagnostiquée dyslexique, la comorbidité est extrêmement fréquente (Habib & Joly-Pottuz, 2008; Kaplan, Wilson, Dewey, & Crawford, 1998; Nicolson & Fawcett, 2007; Witton, 2010). Cela s'explique en partie par le type de diagnostic (différentiel) et par l'imprécision de la définition qui, comme nous l'avons vu, rassemble des troubles hétérogènes. Il devient alors pertinent de s'interroger sur la réelle spécificité (qu'elle soit qualifiée du langage ou des apprentissages) de ce type de troubles (INSERM (dir.), 2007; Jucla, 2009; Kaplan et al., 1998; McArthur, Hogben, Edwards, Heath, & Mengler, 2000; Nicolson & Fawcett, 2007). Les cas de pluri-déficits sont donc très fréquents. Parmi les troubles associés fréquemment, on retrouve la dysorthographe, la dyspraxie, la dyscalculie, la dysphasie, la dysgraphie et le Trouble du Déficit de l'Attention avec Hyperactivité. On remarque d'ailleurs que les enfants dyslexiques sont couramment atteints de TDA/H (Raberger & Wimmer, 2003). Selon Nicolson & Fawcett (2007), le pourcentage de comorbidité entre TDA et dyslexie serait de 11 à 40% selon les études et 50% des enfants dyslexiques auraient montré des troubles moteurs dans leur jeune âge. Pour Habib et Joly-Pottuz (2008), la dysorthographe est étroitement liée à la dyslexie car l'acquisition de l'orthographe est liée à celle de la lecture. Dans certaines études, les deux phénomènes sont considérés comme un seul trouble, appelé dyslexie-dysorthographe (Delamare, 2012; INSERM (dir.), 2007). Habib et Joly-Pottuz (2008) ajoutent que les cas de « dysorthographe isolée » sont probablement plus liés à l'insuffisance d'outils de mesure adéquats qu'à une particularité cognitive.

La comorbidité dans le cas des troubles du langage (ou des apprentissages) relèverait plus d'une règle que d'une exception, ce qui fait que la constitution des groupes de dyslexiques, comme le font remarquer Ramus, Pidgeon et Frith en 2003, conditionne fortement le résultat des études cliniques menées. Cet état de fait pose un réel problème lorsque l'on veut mettre en évidence des relations entre les différents déficits caractéristiques d'une pathologie, comme par exemple dans l'étude de Raberger et Wimmer de 2003. Il arrive

même que la terminologie de « troubles spécifiques du langage » soit abandonnée au profit de celle de « troubles complexes du langage » (Rousseau-Giral, Strohl, Bizot, Ravary, & Gossot, 2002).

5) Les possibilités de remédiation

La dyslexie est un trouble persistant que l'on parvient, parfois, à compenser avec le temps. Cependant, beaucoup d'adultes ayant souffert de dyslexie développementale exhibent toujours de faibles capacités en lecture (Pammer, 2014). Il existe d'ailleurs de nombreuses méthodes d'entraînement plus ou moins liées aux théories tentant d'expliquer la dyslexie développementale (INSERM (dir.), 2007). Jucla (2009) a d'ailleurs dressé une liste d'entraînements décrits dans la littérature. Il est intéressant de constater que, bien que certains d'entre eux soient basés directement sur des activités de lecture, ils ciblent le plus souvent des compétences sous-jacentes à la lecture. Les principaux types de programmes qu'elle a ainsi identifiés sont :

- L'entraînement des capacités phonologiques et de lecture en milieu scolaire
- L'entraînement audiovisuel dans différents contextes
- L'entraînement de la perception auditive
- L'entraînement des capacités phonologiques et de la perception auditive
- L'entraînement de la perception visuelle

Le plus souvent, on fait appel à des entraînements phonologiques mais qui portent aussi sur la morphologie (comme par exemple dans (Arnbak & Elbro, 2000)), la syntaxe ou le niveau discursif. Les méthodes basées sur les aspects phonologiques ont montré une amélioration des capacités phonologiques après un entraînement intensif (Wise, Ring, & Olson, 2000). Jucla (2009) conclut d'ailleurs sur le fait que les entraînements semblant améliorer le plus le niveau de lecture sont ceux qui incluent une part phonologique et un traitement visuo-verbal (comme par exemple dans (Magnan & Ecalle, 2006)). Habib et Joly-Pottuz (2008) confirment aussi qu'il apparaît que les entraînements multimodaux sont plus efficaces que les entraînements exclusivement phonologiques. Cependant, la mise au point efficace de tels d'entraînements à visée de remédiation ne semble pas être une chose évidente (voir par exemple (Bogliotti & Serniclaes, 2011)).

La plasticité cérébrale consécutive aux phénomènes de compensation, ou un entraînement intensif, peut être évaluée à l'aide de l'imagerie fonctionnelle (Nenert, Leveque, Lenormand, Evrard, & De Schonen, 2009; Temple et al., 2003). Par exemple, on peut observer une augmentation de l'activation dans les régions temporelles de l'hémisphère droit chez des individus dont la dyslexie est compensée (Ingvar et al., 2002). Chez l'adulte il est possible, à l'aide de l'imagerie médicale, de différencier des individus n'ayant jamais exhibé de déficit en lecture d'individus avec des antécédents de dyslexie, et de différencier ceux qui l'ont compensée de ceux qui ne l'ont pas compensée (Giraud, Trébuchon-DaFonseca, Démonet, Habib, & Liégeois-Chauvel, 2008). Il reste aussi possible de faire suivre à ces personnes un entraînement améliorant leurs performances et se traduisant physiologiquement par un accroissement de l'activité cérébrale de l'hémisphère gauche ainsi que par une activation compensatoire dans l'hémisphère droit (Eden et al., 2004).

B) Aspects pratiques

La dyslexie et plus généralement les troubles neuropsychologiques sont omniprésents dans toutes les sociétés et pèsent considérablement sur les individus et leurs familles (Collins, Hewson, Munger, & Wade, 2010; Funk & Ivbijaro, 2008; Macdonald, 2010). Ainsi les enfants présentant des troubles de la lecture ont un risque beaucoup plus fort que le reste de la population de présenter des comportements antisociaux (Habib & Joly-Pottuz, 2008). Réciproquement, il est observé un taux élevé de troubles de la lecture chez les délinquants (Macdonald, 2007). Ces constats plaident en faveur d'un lien entre condition sociale et difficultés scolaires (INSERM (dir.), 2007). D'autre part, on peut remarquer qu'il y a un énorme écart entre la quantité de population atteinte de ces troubles et la proportion de cette population bénéficiant d'un traitement. Ceci peut s'expliquer, comme nous venons de le constater, par le fait que le diagnostic de la dyslexie, ou plutôt des dyslexies, n'a rien de trivial. La complexité à définir cette pathologie va s'exprimer au niveau de la prise en charge des individus en souffrant. Cette dernière est d'ailleurs fréquemment qualifiée de « parcours du combattant » (Rousseau-Giral et al., 2002). Nous nous proposons donc d'observer, dans la présente section, en quoi consiste cette prise en charge dans les faits.

1) Au niveau de l'éducation

Un certain nombre de mesures sont actuellement mises en place sur le territoire français afin de répondre aux besoins éducatifs particuliers des enfants atteints de handicaps tels que la dyslexie. La loi française de 2005, relative à l'égalité des droits et des chances des personnes handicapées, met d'ailleurs en avant le droit à la scolarisation des élèves en situation de handicap (Assemblée Nationale, 2005). Cette loi précise que ce n'est pas à l'élève de s'adapter mais à l'école de s'adapter à lui et à son handicap, de l'inclure dans le cursus standard (Dejonckère, 2013). Cependant, le parcours de formation des élèves en difficultés scolaires graves et persistantes apparaît comme complexe (**Figure 3**). D'autre part, l'inclusion de l'élève ne peut être garantie. Afin de déterminer quels enfants doivent bénéficier de ce parcours, il existe des mesures visant à dépister les enfants jugés potentiellement dyslexiques, et à mettre en place des aménagements pour ceux chez qui le trouble est avéré. Les aménagements ont pour objectif d'éviter l'orientation de l'élève vers un institut spécialisé (Gombert, Feuilladiu, Gilles, & Roussey, 2008). Les nouvelles technologies participent efficacement à la mise en place de tels aménagements. Aux États-Unis, un fond fédéral est d'ailleurs consacré à fournir des technologies d'assistance aux élèves qui en ont besoin (United States Congress, 2004).

a) La détection et le dépistage

La prise en charge est initiée lorsque l'équipe d'enseignants, ou la famille de l'enfant, s'interroge sur les difficultés graves et persistantes d'un élève. La détection consiste à déceler chez les élèves, au moment de l'acquisition du langage oral ou écrit, les signes qui préfigurent de difficultés, par exemple à parler, à communiquer, à comprendre les mots ou encore à reproduire ou segmenter les phonèmes (Rousseau-Giral et al., 2002).

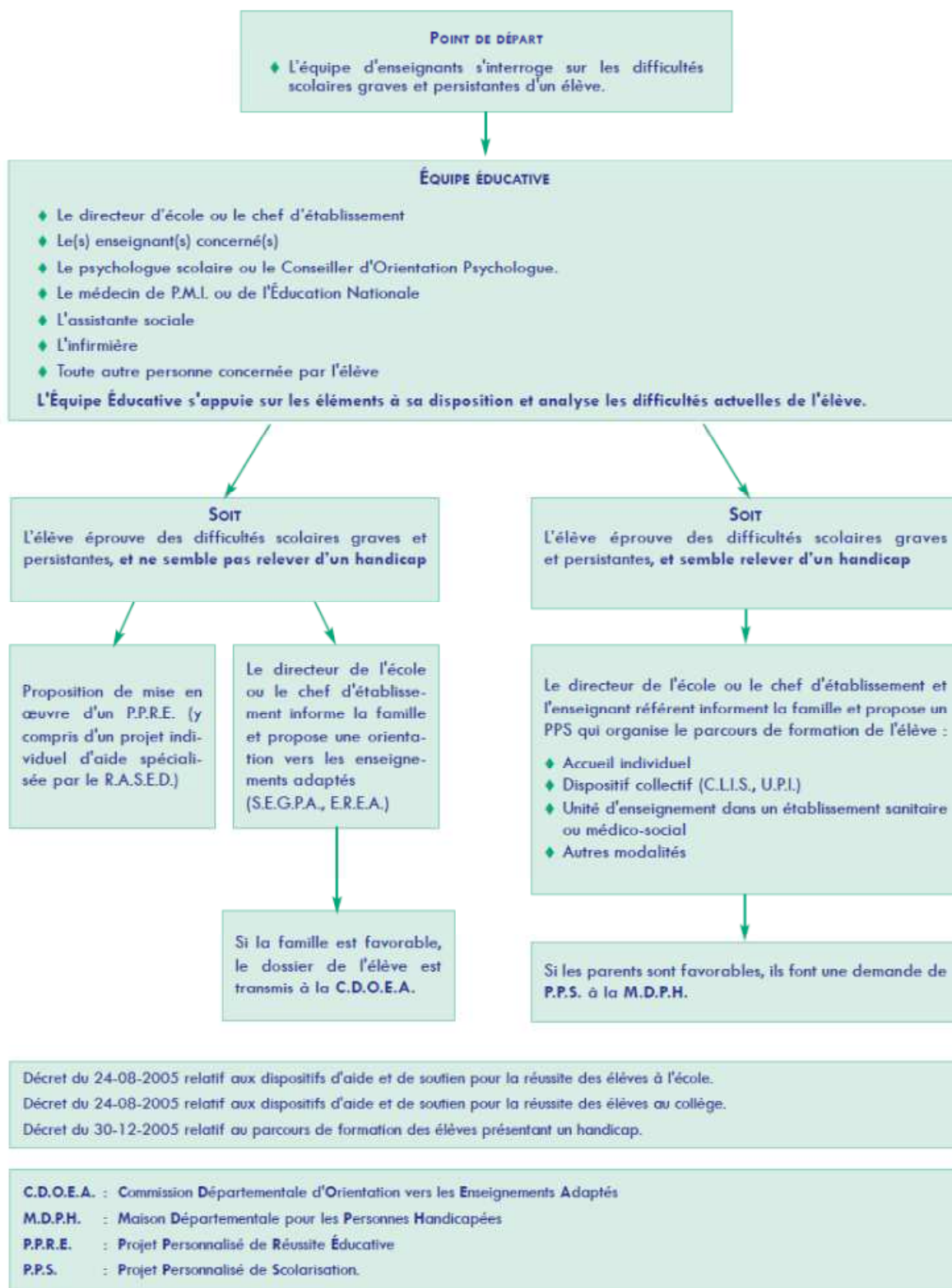


Figure 3 : Représentation du parcours de formation des élèves en difficultés scolaires graves et persistantes, avec ou sans handicap (source (M.D.P.H. de Guadeloupe, 2007))

Malheureusement cette détection survient souvent à un âge avancé. Le meilleur moyen de repérer de façon précoce des troubles reste la vigilance des enseignants. Ces derniers ne sont cependant pas nécessairement correctement informés quant aux difficultés

rencontrées par les dyslexiques (Dejonckère, 2013). A ce titre, il leur est de nos jours dispensé des formations et remis des guides (comme par exemple (Académie de Dijon, 2009)) destinés à les sensibiliser aux caractéristiques des troubles des apprentissages. La dyslexie semblant profiter de facteurs congénitaux (Ramus, 2010), il peut aussi être pertinent de cibler particulièrement la détection sur les enfants qui ont des antécédents familiaux (Habib & Joly-Pottuz, 2008).

Après la détection vient le dépistage. Il s'agit d'effectuer un bilan préliminaire afin d'éliminer les autres hypothèses qui pourraient expliquer les difficultés observées (voir la sous-section **Définition formelle et caractérisation de la dyslexie développementale**). Le dépistage peut être effectué à l'aide de questionnaires (INSERM (dir.), 2007). Si ce dépistage s'avère positif, il convient alors d'orienter l'enfant vers un spécialiste, ou une équipe de spécialistes, dans le but de procéder à un bilan plus approfondi. On remarquera cependant que le dépistage est encore peu organisé, parfois négligé, et que ces modalités varient en fonction du choix des acteurs locaux (Rousseau-Giral et al., 2002). Pour être efficace, le dépistage doit rapidement être coordonné par le médecin scolaire, suite à toute détection.

b) Les aménagements et la compensation en milieu scolaire

L'aménagement passe par la mise en œuvre d'une pédagogie et d'un soutien adaptés. Cette nécessité de personnaliser l'enseignement doit se traduire par l'utilisation de supports et d'explications appropriés au handicap de l'élève (Dejonckère, 2013). L'important est d'adapter l'enseignement aux besoins spécifiques des élèves souffrant de troubles des apprentissages (pour des exemples d'aides et d'adaptations, consulter (Académie de Dijon, 2009; Faure-Brac, Gombert, & Roussey, 2012; Gombert et al., 2008)). Les élèves en grande difficulté peuvent de plus bénéficier d'une auxiliaire de vie scolaire. Cependant, les aménagements existants doivent faire l'objet d'adaptations nouvelles afin de consolider leur efficacité, d'uniformiser les bonnes pratiques et de lutter contre les discriminations (Becchetti-Bizot et al., 2008; Rebeyrol et al., 2010). De nombreuses avancées ont déjà été réalisées consécutivement à des rapports collectifs réalisés ces dernières années (INSERM (dir.), 2007; Ringard, 2000; Rousseau-Giral et al., 2002), mais il reste encore beaucoup de chemin à parcourir dans le domaine de l'égalité des chances au sein de notre système éducatif.

La technologie moderne peut être utilisée à titre compensatoire dans le cadre d'un aménagement scolaire. Elle rend, par exemple, possible la mise en place d'un profilage individuel relatant de la grande diversité de dyslexiques. Des initiatives (Mejía Corredor & Fabregat Gesa, 2012; Sampath, Sivaswamy, & Indurkha, 2010b) en milieu scolaire et universitaire ont déjà montré l'intérêt d'une telle gestion de profils. La multimodalité des interactions permet, en plus d'un profilage individuel, de proposer une interaction personnalisée avec l'outil informatique (Mejía Corredor & Fabregat Gesa, 2012). D'autre part, un grand nombre d'études déjà publiées utilisent, sans forcément l'explicitier (comme par exemple dans (Facoetti et al., 2010)), des approches multimodales. Ce genre d'approches a pour but de permettre l'intégration des informations par redondance/recoupement de données, c'est-à-dire de susciter un effet de synesthésie (Turk, 2014). De plus, il est largement reconnu que le traitement automatique de la parole est une

technologie d'assistance qui offre des avantages potentiels pour les personnes avec un large éventail de handicaps. En particulier, pour les personnes ayant un handicap physique et celles souffrant de dyslexie, la technologie de reconnaissance vocale peut grandement améliorer la qualité de vie (Conn & McTear, 2000). On pourra aussi noter que, dans le cas de déficiences auditives à un niveau sensoriel, il existe des solutions pour lesquelles les analyses multimodales peuvent être efficacement mises à contribution (Tessendorf et al., 2010). Les logiciels d'assistance, pour élèves ayant des difficultés d'apprentissage liées au langage, sont catégorisés de la façon suivante (Hecker & Engstrom, 2005) :

- Le traitement de texte
- La reconnaissance vocale
- La synthèse vocale
- La cartographie mentale

En particulier l'efficacité de la reconnaissance et de la synthèse vocale, aussi bien en cours qu'en situation d'examen, est avérée pour assister les dyslexiques (MacArthur & Cavalier, 2004). Chez eux, on constate que des modifications acoustiques peuvent améliorer l'intelligibilité du discours (Sampath, Sivaswamy, & Indurkha, 2010a). En effet, l'utilisation du « discours clair » a montré son efficacité auprès de tels enfants (Bradlow, Kraus, & Hayes, 2003). Au niveau du traitement de texte, une possibilité de compensation consiste en la mise en place de mécanismes visant à faciliter la lecture pour les enfants dyslexiques. Par exemple les espaces entre les lettres (Zorzi et al., 2012) ou encore leurs couleurs (Altschuler, Altschuler, Altschuler, Samber, & Ramachandran, 2007) peuvent être adaptés. Des polices spécifiques sensées faciliter la lecture, telle que OpenDyslexic (Gonzalez, 2014), existent aussi. Pour finir, on signalera que de tels moyens de compensation donnent, à certaines personnes ayant des difficultés d'apprentissage, les moyens de surmonter les obstacles sociaux auxquels ils sont confrontés dans la vie en général (Macdonald, 2010). Il existe un très grand nombre de logiciels dédiés intégrant tout ou partie de ces moyens de compensation, mais nous n'en ferons pas d'énumération exhaustive ici.

2) Au niveau des soins médicaux et paramédicaux

La mise en place de solutions pour les enfants dyslexiques passe aussi par leur prise en charge par des professionnels de santé des milieux médicaux et paramédicaux. Les principaux professionnels auxquels sont confrontés les enfants sont les orthophonistes, mais leur seule intervention n'est pas suffisante. Voilà concrètement comment est réalisée la prise en charge et quels sont les outils qui peuvent être employés à cette occasion.

a) Le diagnostic et la prise en charge

Consécutivement à un dépistage permettant de suspecter un trouble du langage, l'enfant doit faire l'objet d'examens diagnostiques afin de confirmer ou d'infirmer l'hypothèse émise. Il appartient au diagnostic d'identifier précisément les fonctions défaillantes, ainsi que les troubles éventuellement associés ou générés. Le diagnostic approfondi relève d'ailleurs d'une équipe pluridisciplinaire (Rousseau-Giral et al., 2002). De plus, on notera que depuis 2001 des centres référents, implantés majoritairement dans les CHU, ont été créés pour

assurer la qualité de ce diagnostic. Ces centres n'ont d'autre part pas vocation à réaliser un quelconque suivi des patients qu'ils diagnostiquent, mais peuvent être sollicités pour valider le résultat de prises en charge à l'aide de bilans (pour une liste de ces centres consulter les annexes de (INSERM (dir.), 2007)).

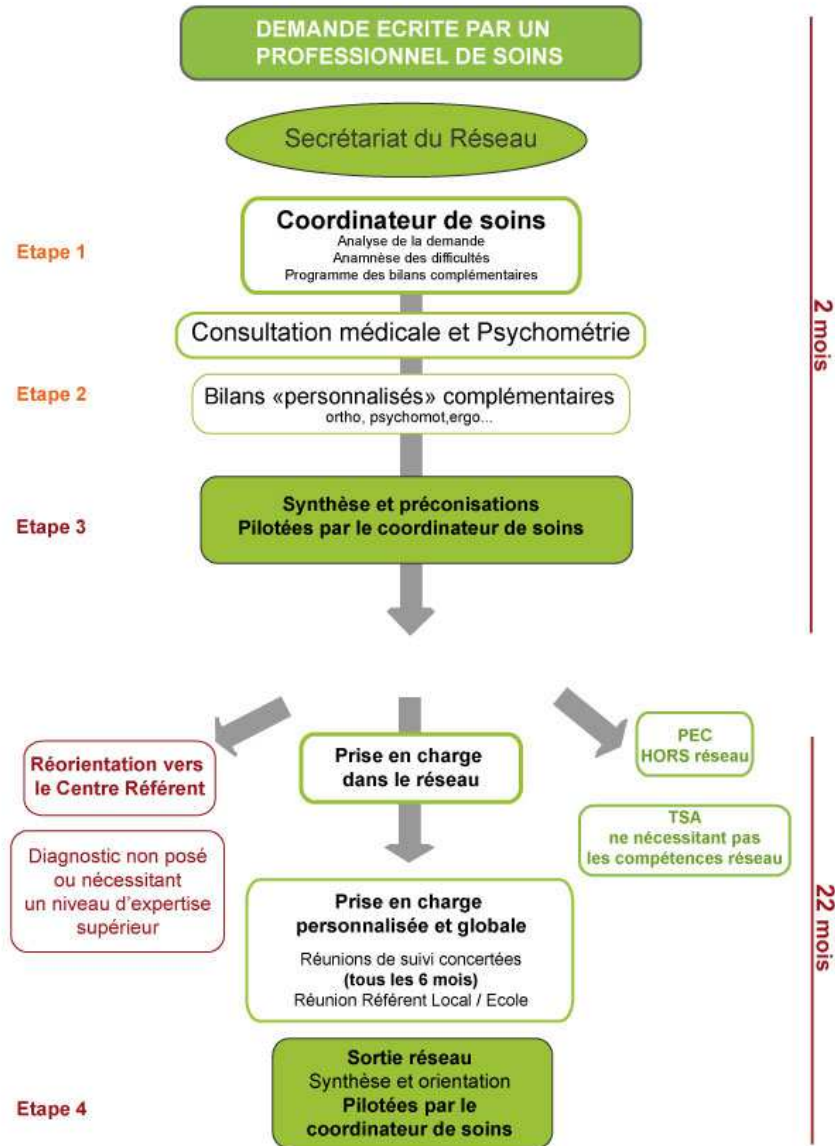


Figure 4 : Représentation du parcours de soins au sein du réseau de santé Dys/10 (source (E=mc^{dys}, 2013))

En 2000, dans son rapport pour l'Education Nationale, Ringard signalait l'insuffisance, aussi bien d'un point de vue qualitatif que quantitatif, de structures d'établissement ou de services spécialisés dans la prise en charge d'enfants dyslexiques. En effet, il avait constaté des listes d'attentes importantes, des orientations inadéquates et des prises en charge insuffisamment adaptées. Ceci pourrait peut-être s'expliquer par le fait que la prise en charge repose en grande partie sur des méthodes qui n'ont pas fait l'objet d'évaluations scientifiques (Habib & Joly-Pottuz, 2008). Un autre rapport de l'Education Nationale et des Affaires Sociales (Rousseau-Giral et al., 2002) préconisait l'organisation en réseau

multidisciplinaire des professionnels de santé impliqués dans la prise en charge des enfants dyslexiques (pour un exemple de parcours de soin au sein d'un tel réseau voir **Figure 4**). Les membres d'une équipe de soin, ou d'un réseau de santé, appartiennent à des disciplines différentes et travaillent de concert afin de fournir des soins de manière coordonnée. Ils partagent ainsi la charge et la responsabilité des soins apportés au patient (Collins et al., 2010). Il existe d'ailleurs maintenant de tels réseaux, à vocation de diagnostic et de prise en charge thérapeutique des enfants et adolescents présentant des troubles spécifiques des apprentissages et/ou de la communication, un peu partout sur le territoire. Cependant ils relèvent d'initiatives locales et les processus organisationnels (voir le chapitre suivant pour plus de précisions concernant les processus métiers) qu'ils mettent en œuvre ne sont régis par aucune standardisation à l'échelle nationale.

Lorsque nous avons été confrontés à la complexité de fonctionnement du système éducatif et du système de santé, il nous a paru judicieux de nous intéresser au ressenti des parents d'enfants dyslexiques (voir l'annexe **Enquête auprès des parents d'enfants dyslexiques**). Nous voulions ainsi en savoir plus sur la compréhension qu'ils avaient du système de santé et mieux appréhender leur rapport à la prise en charge de leurs enfants. En particulier, notre enquête montre qu'il y a une confusion de rôle entre médecin et orthophoniste (voir les réponses à la question 4) comme le soulignait déjà un rapport de l'Education Nationale (Ringard, 2000). D'autre part, une enquête avait montré que les médecins eux-mêmes, qui sont les principaux interlocuteurs des orthophonistes, ne savent pas très bien quels sont leurs champs d'intervention (Bresac & Chigros, 2012). Dans la grande majorité des cas, les enfants des personnes ayant répondu à notre questionnaire ont d'ailleurs droit à des séances d'orthophonistes (voir les réponses à la question 5). Cependant, ils ont l'impression que la progression de leur enfant est lente, limitée, voire inexistante (voir les réponses à la question 6).

b) Les outils de diagnostic et de remédiation

Il existe de nombreux outils et méthodes utilisés pour le diagnostic et/ou la remédiation. Dans le cas du diagnostic, il s'agit de tests ou de batteries de tests. Nous n'allons pas lister ici l'intégralité des outils existants, mais en donner quelques exemples significativement utilisés en milieu clinique en France :

- Capacités du langage écrit : L'alouette (Lefavrais, 2005), BELEC (Mousty & Leybaert, 1999), EVALEC (Sprenger-Charolles, Colé, Béchenec, & Kipffer-Piquard, 2005), B.A.L.E. (Jacquier-Roux et al., 2010)
- Capacités orales : BEPL (Chevrie-Müller, Simon, Le Normand, & Fournier, 1988)
- Capacités visuo-constructives : figures de Rey
- Intelligence (verbale et non-verbale) : WISC-R, WISC-III ou WISC-IV

En dépit de l'existence de toutes ces méthodes, Habib et Joly-Pottuz (2008) font remarquer qu'il y a un manque d'outils couvrant tous les domaines du trouble du langage, écrit et oral, et validés pour toutes les tranches d'âge. Il existe également, depuis de nombreuses années (Morchio, Ott, Pesenti, & Tavella, 1990; Morchio & Ott, 1989), des solutions logicielles permettant d'évaluer les processus cognitifs des individus potentiellement atteints de dyslexie. Parmi ces outils on trouve des systèmes experts (Blonk, Van Den Bercken, & De Bruyn, 1996). De tels logiciels servent d'outils pour établir le diagnostic et présentent le gros

avantage de pouvoir automatiser la collecte et/ou l'analyse croisée des données issues de différentes modalités (voir par exemple (Costa et al., 2013; Le Jan et al., 2007; Ow, 2012; Trolès et al., 2007)). Certaines de ces batteries informatisées sont destinées à un public d'adultes, comme par exemple des étudiants (Mejía, Clara, & Fabregat, 2011; Mejía, Diaz, Jiménez, & Fabregat, 2012). D'autres, se voulant fondamentalement objectives, peuvent s'intéresser aux mouvements oculaires (voir par exemple (Al-Wabil & Al-Sheaha, 2010)). D'une manière générale, nous avons remarqué que les outils informatisés permettant d'assister l'ensemble du processus de diagnostic, et ayant fait l'objet d'une publication scientifique, sont encore à l'état de recherche et s'attachent principalement à une analyse comportementale. La question reste alors posée : avoir recours systématiquement à l'imagerie fonctionnelle permettrait-il un diagnostic précoce de l'enfant dyslexique ? Ceci pourrait, par exemple, permettre de débiter les traitements avant le début de l'apprentissage de la lecture.

La remédiation dispose également d'une multitude d'outils très divers. Les approches orthophoniques classiques s'efforcent de développer et d'enrichir les composants langagiers reposant sur la phonologie, le lexique, la morphosyntaxe et le discours au travers d'exercices ciblés et structurés (Bresac & Chigros, 2012). Il existe aussi des approches de remédiation moins conventionnelles, comme celles basées sur le rythme et la musique qui peuvent agir sur le déficit phonologique. En effet, la perception précise du Rise time peut sous-tendre la détection de structures périodiques quand on perçoit à la fois la musique et la langue (Goswami et al., 2010). Par conséquent, les activités musicales nécessitant une coordination entre les rythmes linguistiques et des rythmes musicaux (par exemple : le chant ou la récitation de poésies au rythme d'un tambour) peut avoir des avantages insoupçonnés dans le cadre du développement phonologique (Corriveau & Goswami, 2009). D'autres méthodes, fondamentalement différentes à celles précédemment évoquées, font par exemple travailler l'équilibre postural ou ont recours à des lunettes aux verres teintés (INSERM (dir.), 2007). On pourra remarquer qu'un grand nombre de ces outils adopte une approche ludique afin de favoriser l'engagement de l'enfant dans son traitement (Durkin, 2010; Olson, 2010). D'autre part, ces outils peuvent se présenter sous la forme de logiciels et permettre aux professionnels, qui encadrent la remédiation, de suivre pas à pas les progressions de l'enfant grâce à l'enregistrement précis des erreurs et des temps de réponse. Pour les enfants, ils présentent l'avantage d'être interactifs, et de pouvoir fournir des feedbacks immédiatement (Castel, 2008). Grâce à cette possibilité de personnaliser l'expérience, et a fortiori l'apprentissage qu'elle engendre, les interventions réalisées à l'aide d'ordinateurs semble être plus efficaces que celles effectuées classiquement (Mioduser, Tur-Kaspa, & Leitner, 2000). Voici quelques exemples de logiciels de remédiation s'apparentant à des jeux vidéo :

- Détermination phonologique : Play On (Danon-Boileau & Barbier, 2006)
- Reconnaissance de mots : Dominos des Mots (Castel et al., 2008)
- Enseignement multi sensoriel : Dyslexia Activity System (Ohene-Djan & Begum, 2008)

On notera que, quels que soient les soins prescrits, ces derniers doivent toujours s'associer à la prise en charge pédagogique de l'enfant (INSERM (dir.), 2007). Ainsi les aménagements scolaires ont pour objectif d'offrir une pédagogie adaptée à ses besoins et ses possibilités dans ses domaines déficitaires (voir la sous-sous-section **Les aménagements et la compensation en milieu scolaire**).

Avec notre enquête (voir l'annexe **Enquête auprès des parents d'enfants dyslexiques**), nous nous sommes aussi intéressés à la connaissance, par les parents, des outils informatiques entourant la prise en charge des enfants dyslexiques. Nous avons ainsi pu constater qu'ils étaient presque tous informés de l'existence de logiciels de compensation (voir les réponses à la question 9) et qu'ils pensent majoritairement que ce type d'aide est efficace (voir les réponses à la question 12). D'autre part, ils sont prêts à les conseiller à d'autres familles (voir les réponses à la question 13). Cependant, seulement la moitié de leurs enfants y ont réellement recours (voir les réponses à la question 11) et parmi ceux-là, plus d'un tiers n'ont reçu aucune formation à l'utilisation de ces outils (voir les réponses à la question 11 bis). Il semble donc ressortir de notre enquête, sous réserve de représentativité de notre échantillon, que l'allocation de ces technologies d'assistance ainsi que la formation à leur prise en main posent problème.

C) Les différents problèmes posés

Comme nous avons pu le voir tout au long de cette partie, la dyslexie est une pathologie complexe dont la compréhension n'est pas encore entière. En particulier, le parcours de l'enfant dyslexique, du repérage à la fin de sa prise en charge, est un véritable dédale impliquant un nombre considérable d'intervenants. Certains d'entre eux ne sont d'ailleurs pas suffisamment informés de l'état des connaissances entourant ce trouble et de la démarche à suivre. L'examen que nous avons pu en faire, aussi bien au niveau théorique que pratique, nous a permis d'identifier un certain nombre d'obstacles à son étude et à la prise en charge des personnes qui en souffrent. Le **Tableau 1** résume ceux, parmi ces problèmes, auxquels nous pensons pouvoir apporter des solutions par le biais de l'informatique. Assurément, l'informatique appliquée au domaine médical/paramédical peut présenter d'énormes avantages et permettre de significatives avancées pour les professionnels de santé ainsi que pour les patients (Hersh, 2004).

Référence du problème	Descriptif du problème
PROB1	Les pratiques requièrent de l'harmonisation, aussi bien au niveau des critères de diagnostic que des processus de prise en charge. Le niveau de standardisation affecte l'efficacité et la qualité des soins prodigués.
PROB2	La recherche autour de la dyslexie est active et a besoin d'outils, efficaces et simples d'usage, permettant la mutualisation et les échanges. La diversité des recherches, et des outils utilisés, peut limiter la reproductibilité et ralentir l'avancée des progrès.
PROB3	La distance entre praticiens et chercheurs est parfois abyssale, ainsi un grand nombre de méthodes utilisées en clinique sont obsolètes et/ou ne reposent sur aucun fondement scientifique. Il est nécessaire de pouvoir valoriser efficacement la recherche.
PROB4	Le diagnostic nécessite l'acquisition et l'analyse de beaucoup de données, afin d'estimer les caractéristiques du patient, ce qui prend beaucoup de temps et implique beaucoup de praticiens.

	Le partage entre cliniciens de ces informations est critique.
PROB5	Les caractéristiques physiologiques de la pathologie ne sont pas assez utilisées en contexte clinique. Le cloisonnement entre les analyses comportementales et physiques reste important, en partie en raison du coût d'acquisition exorbitant du matériel.
PROB6	La mise en place d'une remédiation adaptée, dont on peut mesurer les effets et l'efficacité, n'est pas chose facile. Une telle nécessité de personnalisation des soins passe par une prise en charge individu-centrée.

Tableau 1 : Récapitulatif des problèmes identifiés

Pour être plus spécifique, nous envisageons de développer un système apportant des solutions à ces différentes difficultés simultanément. A cette fin, nous allons maintenant présenter les différentes améliorations techniques qu'un tel système devrait pouvoir offrir.

II) Les axes d'amélioration technique de la prise en charge

Dans la présente partie, nous allons traiter du développement d'un système informatique pouvant contribuer à contourner certaines des difficultés liées à la prise en charge de la dyslexie. En effet, comme nous l'avons exposé précédemment, cette pathologie pose problème au niveau de sa compréhension théorique, de son diagnostic et de sa remédiation. Il conviendrait donc d'identifier quels outils informatiques pourraient mitiger ces difficultés, c'est-à-dire déterminer quels composants d'un système permettraient la mise en place d'un cadre propice à des avancées théoriques et pratiques autour la dyslexie. D'autre part, on notera que la coordination des soins et les tâches administratives requièrent beaucoup de temps et d'effort alors qu'elles ne sont pas véritablement créatrices de valeur pour les patients (Naylor, Kudlow, Li, & Yuen, 2011). En particulier, ceci nous amène à considérer les aspects de leur prise en charge qui sont relatifs aux processus de soins collaboratifs et intégrés (Blatt, Crouse, & Wilson, 2012), c'est-à-dire relatifs respectivement aux interactions et aux incorporations de spécialités médicales/paramédicales avec les soins courants (Collins et al., 2010). On peut d'ailleurs préciser que les processus de soins intégrés peuvent être définis comme des principes organisationnels englobant le suivi, le partage et la fluidité des soins dispensés (Protti, 2009). Ces processus sont des « processus métiers », c'est-à-dire des séquences structurées d'activités spécifiques, effectuées par plusieurs personnes, machines et systèmes, dans le but d'accomplir un objectif d'entreprise (Davis & Brabänder, 2007; Ko, 2009). On fera remarquer qu'ils pourront aussi être appelés processus de travail. Afin de cerner les niveaux sur lesquels un tel système peut agir, commençons par décrire de manière basique son contexte d'incorporation au sein des pratiques.

A) Le principe général du système

Le but de notre système est de supporter le travail coopératif en fournissant des outils pour construire, assister et contrôler des processus médicaux et/ou paramédicaux. En effet, afin d'assurer l'organisation fonctionnelle – la coordination entre les différents acteurs – il est nécessaire de se concentrer sur les processus métiers, c'est-à-dire sur l'ensemble des activités créatrices de valeur du point de vue du patient (Johannesson & Perjons, 2000). En particulier, un tel système informatique doit permettre de faire de la télémédecine (Wootton, 2001) et de suivre les dossiers médicaux des patients (Porumb, Porumb, Orza, & Budura, 2010). Dans notre cas spécifique, nous concentrons notre attention sur les patients souffrant de troubles neuropsychologiques, mais ces approches techniques ont pour mérite d'être applicables à un large spectre d'autres domaines cliniques (Forslund, Phillips, Kilman, & Cook, 1996). Il est par ailleurs notable que les patients, tels que ceux atteints de dyslexie, rencontrent presque systématiquement des handicaps de communication. De plus, il est reconnu que la communication au sein de domaines pluridisciplinaires, comme dans celui de la médecine, peut être compliquée et devenir critique (Lanzola, Gatti, Falasconi, & Stefanelli, 1999; Powell & Hohenhaus, 2006). L'idée pour surmonter simultanément ces difficultés, rencontrées à la fois par les patients et les professionnels de santé, est de partager les

efforts et les composants impliqués dans une communication assistée par ordinateur pour à la fois le diagnostic et la remédiation des patients, mais aussi pour le travail collaboratif et le partage de connaissances au sein du personnel médical (Sainfort, Jacko, Edwards, & Booske, 2007). En effet, la communication entre les différents utilisateurs du système – les praticiens, les patients et les chercheurs – est un point clef de la bonne marche des processus relatifs à la prise en charge des patients et pourrait être efficacement assurée par un **Système d'Information** (Mechanic, 2008; National Transitions of Care Coalition, 2010). Proposer à ces différents acteurs un système informatique, intégrant fonctionnellement et techniquement les différents aspects liés à la communication dans le cadre de leurs activités, mais aussi dans celui de la coordination de ces dernières, apparaît crucial. On notera qu'on parle d'intégration verticale lorsqu'un système informatique permet d'assister les activités dans un domaine fonctionnel spécifique (**Figure 5**). Dans le cas d'un système assistant les activités transverses à plusieurs domaines fonctionnels, on parle d'intégration horizontale (Morley, Bia-Figueiredo, & Gillette, 2011). En particulier, l'intégration horizontale va fréquemment porter sur la description de processus organisationnels, c'est-à-dire sur la planification et la gestion des ressources. Avec notre système, nous recherchons à supporter ces deux types d'intégration. Ainsi chaque corps de métier et/ou département, constitué de professionnels partageant une même expertise et impliqué dans la prise en charge de la dyslexie, sera vu comme un domaine fonctionnel dont nous tenterons d'automatiser tout ou partie des interactions intrinsèques et extrinsèques.

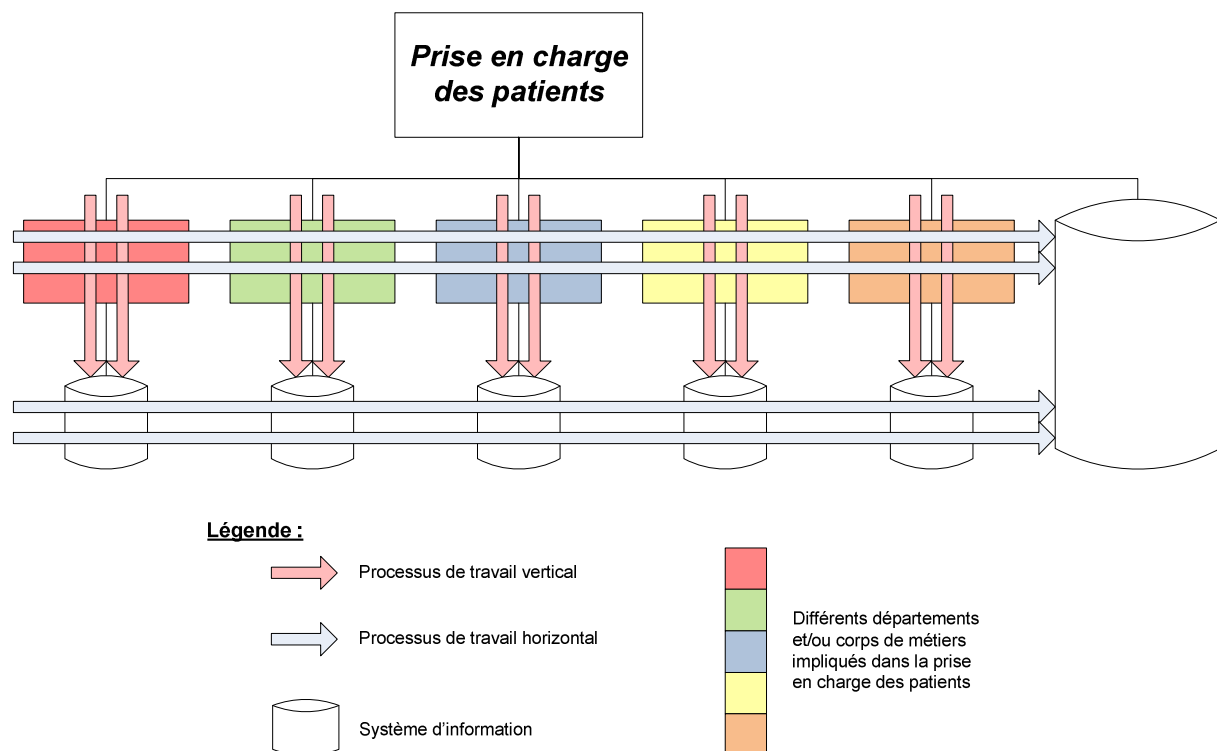


Figure 5 : Représentation du principe d'intégration verticale et horizontale dans notre contexte d'application

L'intégration d'une organisation se dégage autour d'un continuum de collaborations (B. Heath, Romero, & Reynolds, 2013) et, d'une manière générale, elle est motivée par une volonté d'augmenter son efficacité et de gagner des parts de marché tout en maîtrisant ses coûts (Walston, Kimberly, & Burns, 1996). En particulier, l'intégration horizontale vise à

procéder à des économies d'échelle alors que l'intégration verticale permet principalement d'optimiser le coût des transactions entre les activités constituant les processus opérationnels (Hernandez, 2000). Il existe d'ailleurs plusieurs niveaux de processus métiers pouvant faire l'objet d'une intégration (Ko, 2009). Ces processus peuvent être impliqués soit dans la production directe de bien et/ou service, appelée front-office, soit dans le développement et la maintenance d'outils et/ou de services indispensables à cette production, appelée back-office (**Figure 6**). Une des idées derrière un système intégré est d'offrir un accès facilité à l'information via différents systèmes implantés dans différents contextes (Ellingsen & Røed, 2010). Par conséquent, les nouvelles technologies de l'information peuvent permettre de constituer des infrastructures, c'est-à-dire des ressources partagées entre groupes d'utilisateurs du système, aptes à l'intégration des pratiques au sein de tout type d'organisations, y compris au sein de celles de santé (Hanseth & Lundberg, 2001; Hanseth & Lyytinen, 2004).

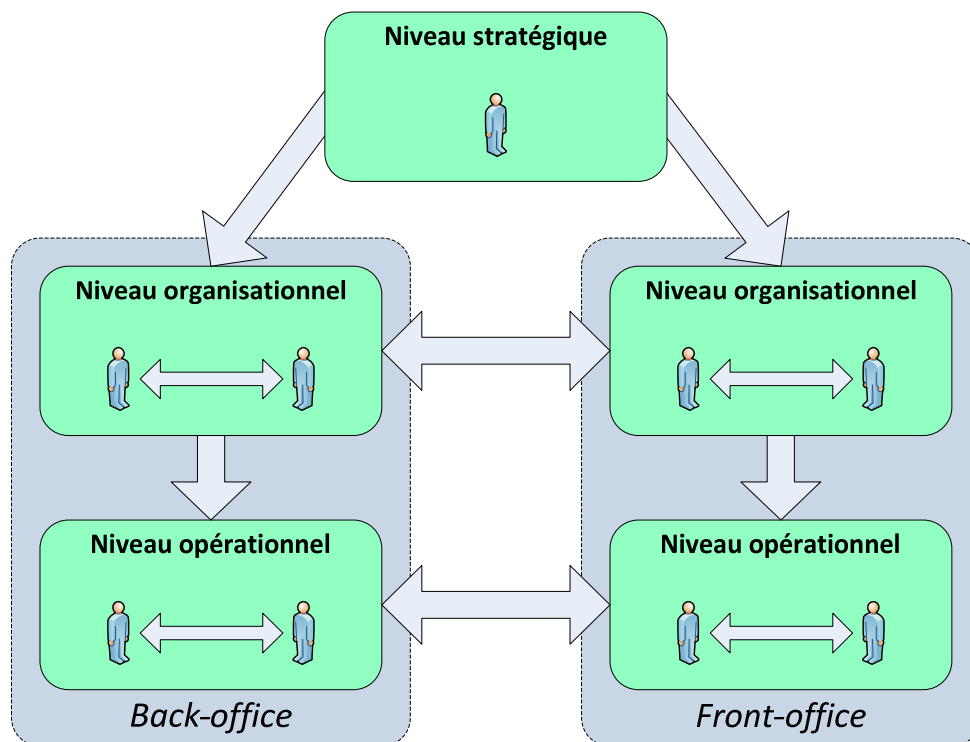


Figure 6 : Représentation des différents niveaux des processus métiers (à partir de (Ko, 2009))

Dans le contexte clinique, la collaboration entre praticiens est importante car elle améliore les connaissances et les compétences de chacun (Doherty, 1995). Afin d'appréhender une organisation médicale dans sa globalité, on identifie fréquemment jusqu'à six niveaux de collaborations (Collins et al., 2010; Doherty, 1995; Heath et al., 2013) :

- Collaboration minimale : les praticiens ne sont pas dans le même établissement, ne partagent pas le même SI et ne communiquent que sporadiquement.
- Collaboration basique à distance : les praticiens ne sont pas dans le même établissement, ne partagent pas le même SI et ils communiquent régulièrement d'une manière plus structurée mais pas systématique.

- Collaboration basique co-localisée : les praticiens sont dans le même établissement, mais ne partagent pas le même SI. Ils communiquent régulièrement d'une manière structurée, mais pas systématique, qui est facilitée par leur proximité.
- Collaboration étroite à travers un système partiellement intégré : les praticiens sont dans le même établissement et partagent certaines parties du SI. Ils se comportent comme une équipe où les prises de décisions s'effectuent collégialement.
- Collaboration étroite à travers un système intégré : les praticiens sont dans le même établissement et se partagent le même SI pour lequel certains éléments peuvent ne pas être intégralement implémentés. Ils se comportent comme une équipe où chaque membre a conscience de son rôle et de celui des autres membres.
- Collaboration entière au sein de pratiques fusionnées : les praticiens sont dans le même établissement et se partagent un même SI entièrement implémenté. Les patients et le personnel de santé perçoivent la prise en charge comme un tout.

Le choix d'un niveau de collaboration, et donc d'intégration, revient aux décideurs du milieu médical en fonction de leurs besoins et des moyens qu'ils peuvent mettre en œuvre (Boon, Verhoef, O'Hara, & Findlay, 2004). En ce qui nous concerne, nous ne devons que réaliser, au sens large, les outils techniques permettant la réalisation d'une intégration du niveau ainsi retenu. A ce titre, si nous voulons que notre système soit le plus polyvalent possible, il faut que nous puissions livrer un système complètement intégré. Eventuellement, il sera ainsi toujours possible d'en limiter la portée afin de réaliser des intégrations de niveau inférieur.

Comme nous l'avons déjà annoncé, le système que nous voulons développer s'adresse à plusieurs classes d'utilisateurs : les praticiens, les patients et les chercheurs. Les interactions entre ces classes d'utilisateurs sont illustrées sur la **Figure 7**. Cette dernière montre effectivement que les interactions dans un contexte médical sont complexes (**Figure 7 (a)**), mais que les informations échangées au travers de ces interactions peuvent être supportées par un SI (Wangler & Backlund, 2005)(**Figure 7 (b)**) résultant en à la substitution des interactions Homme-Homme par des interactions Homme-Ordinateur standardisées et distribuées (**Figure 7 (c)**). On peut d'ailleurs remarquer que chaque instance de ces classes, c'est-à-dire chaque utilisateur distinct de chaque groupe d'utilisateurs, est capable d'interagir avec d'autres instances de sa propre classe (l'incluant elle-même). Un exemple pourrait être une équipe de praticiens ou de chercheurs travaillant collaborativement, ou un patient surveillant ses propres constantes médicales. Evidemment, un système informatique peut aider à assurer plus facilement les interactions/connexions entre ces acteurs, car il permet d'assurer le stockage et la présentation des informations, des connaissances. Ces services assurés numériquement peuvent par ailleurs être adaptés pour chaque catégorie d'utilisateurs. Afin de permettre au système de faciliter la communication entre eux, il leur sera ainsi uniquement présenté des types de données approuvés au préalable et exposés d'une manière prédéfinie et adaptée. C'est par ce biais que les différents types d'interactions impliqués dans la communication peuvent être simplifiés par l'utilisation du système informatique. Le système se pose alors comme un véritable médiateur à la communication (Coiera, 2001) permettant de passer d'interactions Homme-Homme à des interactions Homme-Machine-Homme (Breen & Matusitz, 2010). On notera en plus que ce dernier type d'interactions, qui peut être fortement automatisé et standardisé (Hanseth & Lundberg, 2001), est moins soumis aux contraintes spatiales ou temporelles et est donc propice à soutenir tous types de collaborations humaines.

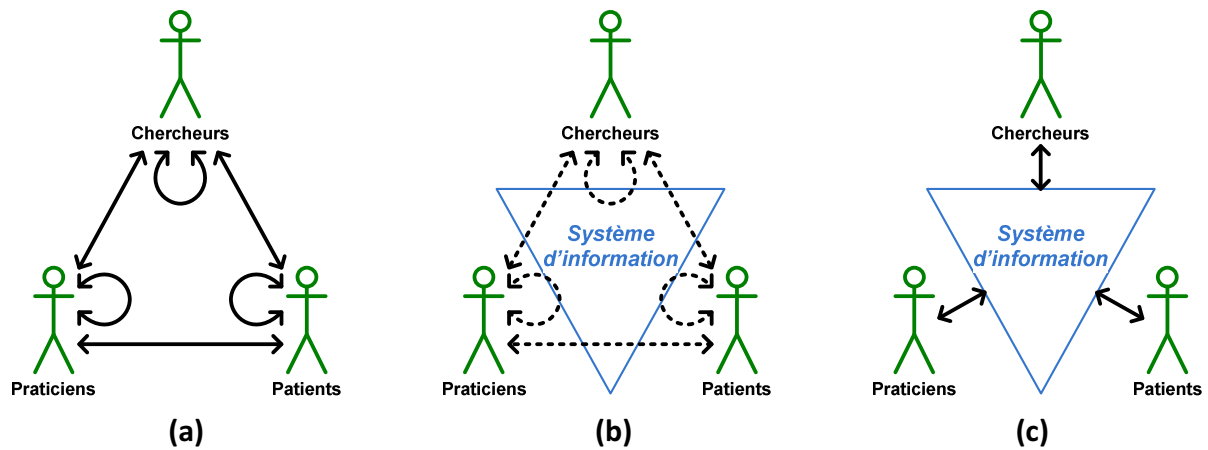


Figure 7 : Représentation de l'intégration à la communication d'un SI dans le domaine de la santé

Les avantages de la mise en place d'un tel système sont multiples. On notera en premier lieu qu'il va permettre de virtualiser les données liées aux patients. La nature virtuelle de ces dossiers offre la possibilité de les stocker de manière distribuée ainsi que de les présenter de manières différentes selon le type d'utilisation (Forsslund et al., 1996). Nous reviendrons, un plus loin dans ce chapitre, sur les problèmes de confidentialité des données que fait émerger ce type de gestion. De telles informations vont évidemment permettre de meilleures prises de décision de la part des praticiens (Johnson, Tu, Musen, & Purves, 2001). On désignera fréquemment ces dossiers médicaux par les termes « dossiers virtuels », « dossiers électroniques », « dossiers numériques », « dossiers informatiques » ou encore « **Dossier Médical Personnel** » (en France un service public éponyme est d'ailleurs mis à disposition des professionnels ainsi que du public (Ministère des affaires sociales et de la santé, 2014)). Avec des DMP bien conçus, et utilisés efficacement, il est possible d'améliorer le niveau de résultats obtenus avec les patients et l'efficacité clinique (Weibel et al., 2013). En effet, il est reconnu, depuis un certain nombre d'années, que l'introduction de l'outil informatique auprès des praticiens augmente l'efficacité et la précision des soins qu'ils dispensent (Makoul, Curry, & Tang, 2001; Sullivan & Mitchell, 1995). De la même manière, si le patient a lui-même accès à ses informations médicales personnelles, il peut informer efficacement les cliniciens sur son propre état de santé (Weiss, 2009). D'une manière générale, l'objectif d'un DMP est d'offrir un support à la prise de décision médicale, d'accroître la coordination entre les différents prestataires de santé et de promouvoir l'utilisation de directives/recommandations médicales afin d'améliorer la qualité globale des soins (Delpierre et al., 2004). Ceci est rendu possible par l'automatisation de nombreuses procédures liées au traitement des informations relatives aux données du patient. Effectivement les systèmes de gestion électroniques des dossiers médicaux présentent de nombreuses fonctionnalités, accessibles potentiellement à distance, telles que (Bates et al., 2001; Ilie, Courtney, & Van Slyke, 2007; Lakbala & Dindarloo, 2014; Makoul et al., 2001) :

- Enregistrer l'historique des patients.
- Rédiger des ordonnances pour des médicaments ou des examens.
- Faire des comptes-rendus des rendez-vous avec des patients.
- Afficher des résultats d'examens médicaux.
- Utiliser des outils d'aide à la décision.
- Bénéficier d'alarmes et de rappels.

- Interagir avec ses collègues (textuellement, vocalement ou visuellement).
- Imprimer des recommandations et/ou informations destinées aux patients.

Un autre avantage lié à l'utilisation de dossiers médicaux électroniques, est de permettre un accès immédiat aux informations relatives aux patients (Martins-Baltar, Laurillau, & Calvary, 2013). Un tel accès à l'information est plus rapide, plus fiable et moins onéreux que les méthodes classiques. En effet, dans les organisations n'ayant pas de SI intégré, les professionnels de santé passent souvent plus de temps à chercher l'information qu'à s'occuper du patient (Protti, 2009). De nos jours, les avancées et la démocratisation des technologies de l'information, déjà impliquées dans la mise en places des DMP, rendent aussi la mise en place de solutions de télémédecine possible (Yellowlees, 2005) et intéressante en termes de coût (Wootton, 2001). D'autre part, remplacer les supports sur papier aide à libérer l'espace physiquement occupé par ces derniers, qui pourra ainsi être mise à profit pour les soins apportés aux patients (Naylor et al., 2011). La standardisation de ces informations ainsi que leur interopérabilité sont aussi des points clefs de cette réduction des coûts, qui ne peut cependant se faire que si les différents systèmes existants peuvent communiquer entre eux (Arney et al., 2014; Protti, 2009; Weiss, 2009). De tels dossiers sont d'ailleurs, par nature, distribués entre plusieurs sites, il convient donc de mettre en place des mécanismes d'agrégation de données afin d'en proposer une vue complète à partir de n'importe quel endroit (Forslund et al., 1996; Johnson, Tu, Musen, et al., 2001).

Enfin, le système doit pouvoir s'interfacer à du matériel médical. En effet, comme nous l'avons précédemment évoqué (voir la sous-section **Définition formelle et caractérisation de la dyslexie développementale**), la dyslexie se caractérise par certaines manifestations physiologiques que l'imagerie médicale moderne est en mesure d'observer. A ce titre, notre système doit pouvoir se connecter au matériel d'imagerie afin d'en intégrer directement les données aux DMP. De tels systèmes, alliant l'intégration de matériels à un SI médicale, sont appelés **Plateforme d'Applications Médicales** (Hatcliff et al., 2012). D'une manière générale, les appareils médicaux sont intégrés verticalement par leurs constructeurs respectifs et les adapter à différents contextes, au travers de mises à jour de leurs capacités de fonctionnement, est difficile et coûteux. On pourra d'ailleurs remarquer qu'un système utilisant différents appareils médicaux est, par essence, distribué (Arney et al., 2014) et donc adapté à la télémédecine (Hanák et al., 2011). L'utilisation des PAM vise à s'affranchir de ces obstacles à l'innovation, c'est pourquoi nous souhaitons que notre système en reprenne les différents aspects. En particulier une PAM a pour but de (Larson, Hatcliff, Procter, & Chalin, 2012) :

- Permettre, au travers d'une infrastructure partagée et dédiée à la communication, l'intégration d'appareils hétérogènes à des SI et de visualisation.
- Supporter des applications d'intérêt médical grâce à cette infrastructure.

Des telles plates-formes ont donc clairement vocation à permettre une intégration des pratiques médicales en exploitant des solutions informatiques aussi bien logicielles que matérielles. On peut d'ailleurs ajouter que ces techniques d'intégration, que les PAM mettent à profit, s'appuient sur plusieurs tendances (Arney et al., 2014; Hatcliff et al., 2012; Larson et al., 2012) :

- L'interopérabilité du matériel, c'est-à-dire le respect de standards en ce qui concerne les fonctions de connexions entre matériel et logiciel. De tels appareils peuvent par exemple disposer de connectivités USB, Bluetooth ou RJ45.

- L'approche en plate-forme, c'est-à-dire avoir systématiquement recours à des API standardisées créant une abstraction du matériel en permettant une exploitation contrôlée et appropriée. On dira communément que cette plate-forme s'appuie sur des middlewares.
- La perspective d'un méta-système, c'est-à-dire l'utilisation de frameworks (Johnson, 1997) permettant la réalisation de fonctions à l'aide de multiples services, fournis par divers systèmes, indépendamment de notions temporelles ou spatiales (Yoo, Kim, Kim, Park, & Chang, 2008). D'une manière générale, on peut d'ailleurs définir un framework comme une application partiellement achevée et réutilisable qui peut être spécialisée pour produire d'autres applications personnalisées (Fayad & Schmidt, 1997).

Nous nous appuyerons, nous aussi, sur ces différentes tendances pour concevoir les différents éléments constitutifs de notre système. Notre PAM va ainsi être un framework apte à supporter l'implémentation d'applications diverses poursuivant des buts variés. Un des points forts d'une PAM est d'être évolutive car elle permet de réaliser des applications qui étendent les fonctionnalités du système global (comme par exemple dans (Kirn, Heine, Herrler, & Krempels, 2003)). En particulier, il peut se créer un lien de dépendance entre applications de telle sorte que certaines applications fournissent un service indispensable au fonctionnement d'autres.

La conception d'une infrastructure d'information entièrement intégrée implique de prêter attention aux pratiques, acteurs et objectifs du système à intégrer (Ellingsen & Røed, 2010). A ce titre notre présentation, préalablement menée sur la problématique de la dyslexie, nous conduit à identifier un certain nombre d'axes d'amélioration potentiels en ce qui concerne la prise en charge des patients souffrant de cette pathologie. Pour être précis, ces axes se présentent comme des réponses apportées aux problèmes que nous avons préalablement soulevés (voir la section **Les différents problèmes posés**). Ces axes sont autant d'objectifs, de besoins, que la mise en place de notre PAM doit viser à satisfaire. Nous nous proposons maintenant d'énumérer, tout en prenant la mesure de leur mise en place dans la prise en charge de patients, les différents points que nous avons ainsi retenus (des constations similaires apparaissent dans (Sainfort et al., 2007)).

B) L'automatisation des tâches de bas niveau

L'intégration verticale que nous comptons proposer passe par l'automatisation de tâches de bas niveau. Par là on entend vouloir, de manière systématique, assister les professionnels de santé principalement dans les parties de leurs activités ne produisant pas nécessairement de valeur curative et pouvant être sources d'erreurs. Ces activités relèvent essentiellement de l'acquisition, du calcul et de l'enregistrement de données dans le SI ou encore le paramétrage d'appareils médicaux. Leur automatisation revient à standardiser certains aspects de la prise en charge des patients et devrait permettre une augmentation de la qualité de soins prodigués (Collins et al., 2010; Dart, Xu, Chatellier, & Degoulet, 2001; Larson et al., 2012). Effectivement un système informatisé, gérant automatiquement des listes de contrôle, peut permettre de pallier les limitations des professionnels de santé, qui ne sont ni plus ni moins que des êtres humains, dans le but d'améliorer la sécurité des patients (Bates

& Gawande, 2003; Bates et al., 2001) et la qualité des soins, tout en augmentant l'efficacité et en réduisant les coûts (Naylor et al., 2011; Powell & Hohenhaus, 2006; Wang et al., 2003). On remarquera d'ailleurs que l'amélioration de la sûreté des soins passe par exemple par le développement de la communication, l'aide aux calculs, le monitoring en temps réel ou l'aide à la décision (Bates & Gawande, 2003).

1) Garantir la reproductibilité et la qualité des pratiques

Un facteur impactant fortement la qualité et la reproductibilité des pratiques est la communication entre les praticiens et les patients. En effet, il est reconnu que la communication dans le domaine multidisciplinaire comme la médecine – aussi bien dans la relation médecin-patient que dans le travail d'équipe – peut être compliquée, doit être contrôlée et peut être un facteur critique conditionnant le succès d'une tâche (Duffy et al., 2004; Powell & Hohenhaus, 2006). Dès lors il paraît pertinent de vouloir assister cette communication par ordinateur afin qu'elle réponde à des standards de qualité préétablis. Par exemple, au cours d'une consultation, l'utilisation de l'outil informatique peut affecter cette communication aux niveaux verbal, visuel et postural. Il a d'ailleurs été identifié spécifiquement quatre domaines pouvant affecter la communication entre cliniciens et patients au cours d'une consultation assistée par ordinateur (Frankel et al., 2005) :

- L'organisation de la consultation
- Les comportements verbaux et non-verbaux
- L'utilisation et la maîtrise de l'ordinateur
- L'agencement spatial de la salle d'examen

Dans tous ces domaines, on remarque que l'utilisation de l'ordinateur, qui ne se fait pas d'une manière aussi simple et naturelle que l'utilisation d'une documentation au format papier, risque d'altérer la connexion entre le praticien et le patient et d'allonger la durée de consultation (Sullivan & Mitchell, 1995). Effectivement, d'une part la technicité requise pour utiliser l'outil peut constituer un frein à la fluidité normal d'un déroulement de la consultation, si le praticien est amené à rentrer manuellement trop de données dans le système (Lobach & Hammond, 1997). D'autre part l'encombrement du matériel qui constitue la partie tangible de l'outil peut aussi introduire des contraintes concernant les manipulations et placements possibles de l'outil. A contrario, l'ordinateur a des avantages indéniables, à commencer par sa capacité à supporter l'acte médical au travers de chacune de ses étapes, et peut aussi fluidifier la consultation en récupérant des données directement à partir d'un SI et renforcer la connexion entre le clinicien et le patient en étant un nouveau support/vecteur de communication entre eux (Frankel et al., 2005). Il convient donc de mettre en place un système informatique, permettant une amélioration de la qualité des soins et veillant simultanément à ne pas mettre le patient en marge de sa propre prise en charge. Pour cela notre système doit permettre au praticien de ne se concentrer que sur son cœur de métier, sans avoir à se soucier des tâches techniques qu'impose fréquemment l'usage de solutions informatiques. Nous voulons ainsi pouvoir nous affranchir de la variabilité que peut avoir l'accomplissement de ces tâches sur l'acte médical, c'est-à-dire répondre au problème **PROB1** en œuvrant dans le sens de la reproductibilité des pratiques par la standardisation à un niveau technique.

Dans l'optique de rendre les pratiques reproductibles, il faut aussi modéliser les processus métiers sur lesquels reposent ces pratiques sous la forme de workflows (Gooch & Roudsari, 2011; Morley et al., 2011; Workflow Management Coalition, 2014). Ainsi, une fois ces processus formalisés, il devient plus aisé de les faire appliquer sur différents sites par différents professionnels disposant de compétences différentes (Leonardi, Panzarasa, Quaglini, Stefanelli, & van der Aalst, 2007). De plus, cette modélisation des processus métiers va permettre d'en assister informatiquement l'exécution (Ko, 2009; Leymann & Roller, 1997) de manière distribuée (Dayal, Hsu, & Ladin, 2001). Par rapport à une description papier, cette informatisation a pour avantage de clarifier les processus et d'en permettre la visualisation à différents niveaux de détails sans pour autant perdre la vue d'ensemble (Ciccarese, Caffi, Quaglini, & Stefanelli, 2005). Il devient aussi possible de travailler spécifiquement sur les interactions entre les utilisateurs et le système lors du déroulement du processus afin d'en améliorer l'efficacité (Naylor et al., 2011). Framinan et al. (2005) identifient d'ailleurs trois niveaux d'abstraction auxquels ce travail de modélisation va se dérouler :

- Le niveau du méta-modèle, où sont spécifiés les outils nécessaires à la description des processus métiers. Un tel méta-modèle de processus métiers est un véritable framework pour ces processus.
- Le niveau de définition des processus métiers, où sont fonctionnellement décrits tous les processus métiers. Chaque processus sert un but spécifique, ce qui permet de les regrouper par types de processus servant un même but.
- Le niveau du modèle de référence, où sont conservées les modélisations retenues comme standards pour les différents types de processus métiers. Ces références ouvrent la voie à l'harmonisation des pratiques, c'est-à-dire à un contrôle plus efficace des processus métiers (Powell & Hohenhaus, 2006).

On remarquera que ce découpage est conforme aux tendances sur lesquelles s'appuient les PAM. D'autre part, en intégrant explicitement des processus métiers, notre système pourra ainsi se poser comme garant de la reproductibilité des pratiques et de leur conformité aux directives/recommandations en vigueur (de Clercq, Blom, Korsten, & Hasman, 2004; Johnson, Tu, & Jones, 2001). Ceci permet aussi d'aider les cliniciens à travailler efficacement en équipe (Protti, 2009). On pourra particulièrement préciser que cette reproductibilité des pratiques passe également par l'automatisation du paramétrage des appareils médicaux (Hatcliff et al., 2012), ce qui est partie intégrante de l'intégration des processus de bas niveaux recherchée. Ce genre d'optimisation ergonomique peut aussi porter sur les parties logicielles appartenant à l'interface utilisateur (voir par exemple (Martins-Baltar et al., 2013)). Il est à ce titre envisageable d'agrémenter notre système de composants offrant des possibilités d'interactions naturelles dans le but d'en simplifier l'utilisation. L'emploi d'ordinateurs peut ainsi permettre d'améliorer la qualité et la reproductibilité des pratiques pour peu qu'on prenne la peine de prendre en compte les contraintes que ses usages spécifiques en milieu clinique imposent.

2) Recueillir et agréger systématiquement les données

Afin de contracter la durée de la prise en charge, c'est-à-dire la rendre plus rapide/efficace, il est primordial que les différents cliniciens partagent des processus de soins et qu'ils aient

conscience des étapes constituant ces processus. En effet, il est nécessaire que les pratiques reposent sur des processus standardisés offrant une visibilité aux praticiens quant aux données attendues par les autres professionnels avec lesquels ils ont à interagir (Collins et al., 2010). De telles considérations sont imputables à la résolution des problèmes **PROB1** et **PROB4**.

Au travers de processus métiers, les échanges entre cliniciens se font à l'aide de transferts et de partages de données. En effet, une intégration efficace nécessite que le personnel de santé partage des informations relatives au patient à certains moments du processus de soins (Protti, 2009). De la sorte l'action de chaque praticien va s'appuyer sur les données que ses collègues lui transmettent ainsi que sur celles qu'il introduit lui-même dans le système coopératif. A ce titre, l'informatique, dont la nature même est de traiter automatiquement de l'information, peut être d'une aide précieuse pour gérer efficacement de gros volumes de données hétérogènes. En particulier, ces traitements automatisés permettent le recueil et l'agrégation de données générées par les patients (comme par exemple dans (Han, Song, Lee, Jang, & Shin, 2006)). Avec l'explosion des capacités de stockage et de la puissance de calcul des ordinateurs modernes, ainsi que de la bande passante des réseaux numériques locaux et mondiaux, il est maintenant possible de procéder systématiquement à ces traitements. Lorsqu'un praticien doit introduire des données dans le système, ces dernières peuvent être collectées par l'intermédiaire de questionnaires, de formulaires, ou d'appareils de mesure. Ces appareils peuvent être plus ou moins intégrés et, dans le meilleur des cas fonctionnellement parlant, allouer la possibilité d'injecter automatiquement leurs mesures dans le système informatique. On remarquera cependant que ce genre de matériel est rarement bon marché.

De ces constats, il apparaît primordial que notre système soit capable de gérer automatiquement et systématiquement de grandes variétés et quantités de données. Ceci pourra d'ailleurs être réalisé à deux niveaux :

- Logiciellement, au travers d'agrégation et de stockage.
- Matériellement, à l'aide d'un matériel bon marché, facile à utiliser et permettant l'acquisition de données multimodales.

Un tel matériel permettant l'acquisition de données multimodales, couplé à un logiciel en permettant l'agrégation et le stockage, ouvre la perspective de collecter automatiquement et systématiquement de grandes quantités de données se rapportant aux patients. Ces données, même si elles ne sont pas toujours indispensables dans l'immédiat, pourraient s'avérer précieuses ultérieurement. De plus, on notera que le fait de rassembler ainsi de grandes quantités de données est d'un grand intérêt pour mesurer l'efficacité des soins et/ou effectuer des recherches (Cragin & Shankar, 2006). On peut imaginer que des études pourront être effectuées sur les DMP de multiples patients afin, par exemple, d'identifier des facteurs de risques de certaines maladies ou de mener des analyses épidémiologiques (Forslund et al., 1996). Pour de raisons d'interopérabilité, la gestion de multitudes des données devrait se faire par l'intermédiaire de méthodes et de structures standardisées. Cela apporte notablement un début de solution au problème **PROB2**. De plus, on note qu'il y a un écart entre la volonté des pouvoirs publics et les difficultés pratiques qu'implique la manipulation de grandes quantités de données hétéroclites (Weiss, 2009). Ainsi, afin de résoudre techniquement ces difficultés, il paraît pertinent de vouloir proposer un framework qui permettra aussi bien de recueillir ces données, les stocker/organiser, les analyser et les

présenter. Un tel framework doit être souple et permettre la réutilisabilité des éléments s’y inscrivant (Jalender, Govardhan, & Premchand, 2012). Pour ce faire il doit définir des composants génériques au travers d’interfaces stables (Fayad & Schmidt, 1997). On signalera que le développement de ces composants génériques doit d’ailleurs passer par l’étude des « design patterns » (Gamma, Helm, Johnson, & Vlissides, 1994). Cependant l’implémentation d’un framework dans son intégralité reste une tâche ardue. En dessiner les contours à travers notre réflexion, et tester différents concepts qu’il pourrait exploiter, sera plus à notre portée.

3) Proposer de l’aide à la décision

Le recueil et l’agrégation de données sont des étapes importantes dans le cheminement qui affranchira le praticien de tâches techniques de bas niveau relevant de l’utilisation d’un quelconque système informatique. L’étape suivante de ce cheminement est l’analyse « intelligente » de ces données, localisées au sein du SI, visant à procéder automatiquement à des calculs ou à des recoupements de ces dernières. Un tel type d’assistance est couramment appelé « aide à la décision » et constitue une fonctionnalité corollaire importante des SI médicaux (Wright et al., 2009). Cette aide apportée aux praticiens peut d’ailleurs être décomposée en deux types (Greenes, 2014; Huser, Rasmussen, Oberg, & Starren, 2011) :

- L’aide active à la décision, basée sur la capture d’évènements du système afin de réagir de manière proactive (comme par exemple les rappels ou les alarmes).
- L’aide passive à la décision, réagissant aux requêtes spécifiques de l’utilisateur et initiant les accès aux données et/ou calculs nécessaires (comme par exemple l’accès à des résultats d’analyse ou à toute information complémentaire).

De telles fonctionnalités peuvent, par exemple, aider les praticiens à mieux suivre certaines directives/recommandations cliniques (Fox et al., 1997; Lobach & Hammond, 1997; Moser, Heller, Gries, Engelbrecht, & DIADOQ, 2001; Peleg & González-Ferrer, 2014), ce qui induit de meilleurs performances de la part de ces professionnels indépendamment de leur niveau d’expérience (Garg et al., 2005; Hunt, Haynes, Hanna, & Smith, 1998). L’aide à la décision tend de cette façon à répondre aux difficultés posées par le problème **PROB1**.

A cette aide à la décision dans un contexte opérationnel peut s’ajouter une aide à la décision à un niveau plus administratif, comme par exemple pour la planification, l’ordonnancement, des soins (Glasspool, Fox, Oettinger, & Smith-spark, 2006) ou à des fins d’analyse économique (Shtilyanova, Feschet, & Pommier, 2011). Dans tous les cas, l’aide à la décision s’appuie sur des connaissances relatives aux domaines pour lesquels ce support est conçu, telles que celles spécifiques au patient, celles générales d’ordre médical (présentation des pathologies, médication, etc.) et celles propres aux procédures médicales (Fox et al., 1997; Wilk, Michalowski, O’Sullivan, Farion, & Matwin, 2008). Dans le cas de notre système, ces connaissances vont prendre la forme de workflows (Quaglini et al., 2000), dont l’efficacité pour l’aide à la décision est avérée (Blaszczynski et al., 2005) ainsi que de modèles multivariés pour, respectivement, les aides actives et les aides passives. On précisera d’ailleurs que les modèles multivariés d’aide à la décision vont s’appuyer sur des indicateurs obtenus à partir des données brutes et/ou agrégées. Dans le contexte de notre travail, un indicateur est un outil d’aide à la décision permettant d’évaluer de manière relativement

objective de l'état d'une ou de plusieurs capacités d'un patient. Une PAM doit être capable, à partir d'informations provenant d'un matériel médical, d'un DMP ou d'autres bases de connaissance, de fournir aux cliniciens les moyens pour apprécier de la qualité de différents indicateurs (Hatcliff et al., 2012). Un tel système permet aussi de comparer l'état actuel de santé du patient avec son historique de santé (Porumb et al., 2010).

On remarquera que les outils déjà sur le marché peuvent présenter une grande hétérogénéité de fonctionnalités (Wright et al., 2009), ainsi il paraît intéressant de vouloir proposer aux praticiens un outil capable d'évoluer simplement en accord avec leurs besoins spécifiques en termes d'aide à la décision. On notera cette fois encore qu'un framework affiche bien ce genre de propriétés. Finalement on remarquera, qu'étant donné l'aspect critique du domaine dans lequel ces outils sont utilisés, il faut particulièrement prêter attention à ce que leur usage soit sûr du point de vue des patients (Fox, 2002).

C) Le support au travail collaboratif

La collaboration virtuelle ainsi que l'ubiquité de l'informatique (Weiser, 1993) jouent ensemble un rôle de plus en plus important dans le système de santé moderne (Kilman & Forslund, 1997; Martins-Baltar et al., 2013; Sainfort et al., 2007). Ceci s'avère particulièrement impactant sur les résultats de l'acte médical, ainsi que sur le coût des services de santé qui lui sont associés (Porumb et al., 2010). Nous avons vu que la prise en charge de patients souffrant de dyslexie peut être particulièrement complexe et impliquer un grand nombre de praticiens différents (voir la section **Aspects pratiques**). C'est pourquoi nous nous proposons maintenant de passer en revue les différents avantages, et parfois contraintes, que la mise en place d'un tel système peut avoir dans le cadre de collaboration entre ces différents professionnels de santé.

On fera remarquer que la mise en place de tels environnements de travail collaboratif, accessibles en ligne, pose la question de la confidentialité des données (Anderson, 1996; Kilman & Forslund, 1997; Lakbala & Dindarloo, 2014; Vasserman & Hatcliff, 2014). En effet le système doit, par exemple, s'assurer que les différents utilisateurs n'accèdent qu'à des données qu'ils ont le droit de consulter et doit procéder à la journalisation de tous les accès à ces informations. D'autre part, les données, empruntant des réseaux plus ou moins publics, doivent faire l'objet de protocoles de transferts spécifiques, utilisant le chiffrement de données, afin d'en protéger la lecture non-autorisée (Arney et al., 2014). Il peut aussi être intéressant d'utiliser un système d'authentification par signatures numériques. De telles exigences en termes de sécurité sont fréquemment établies par les pouvoirs publics, les organismes de soins et les patients eux-mêmes (comme dans le cas de l'étude présentée dans (Wangler, Åhlfeldt, & Perjons, 2003)). Cet aspect, à la fois éthique et légal, est directement lié géographiquement aux endroits où est utilisé le système et peut constituer un frein à son utilisation (Breen & Matusitz, 2010). Cependant, il est primordial d'assurer aux patients que tout sera fait pour garantir la confidentialité de leurs données personnelles, sinon ces derniers perdront confiance dans le système et feront tout pour limiter la quantité de données les concernant (National Transitions of Care Coalition, 2010), ce qui rendra difficile la possibilité de traitements précis et adaptés (Anderson, 1996).

D'autre part, pour pouvoir assurer efficacement la collaboration entre les différents utilisateurs de notre système, il est fondamental d'intégrer les processus métiers qu'ils partagent (Lenz, Peleg, & Reichert, 2012). Ceci signifie, comme nous l'avons déjà évoqué, que la PAM doit pouvoir gérer des workflows afin de répondre au problème **PROB4**.

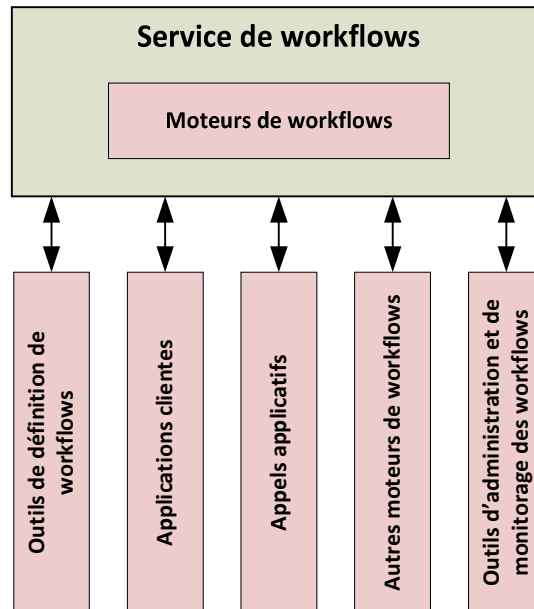


Figure 8 : Schéma des éléments et interfaces du modèle de référence des workflows (d'après (Hollingsworth, 1995))

Comme présenté sur la **Figure 8**, un système de gestion de workflows doit disposer d'un service de workflows (comme, par exemple, Bossa (Gonçalves, 2013)). Ce dernier permet, au travers d'interfaces, d'accéder à des moteurs de workflows (voir par exemple (Greenes, Boxwala, Sloan, Ohno-Machado, & Deibel, 1999)). Ces interfaces offrent, de la sorte, une interopérabilité avec différents types d'applications (Hollingsworth, 1995, 2004) :

- Des outils de définition de workflows, c'est-à-dire des applications permettant de décrire formellement les processus métiers dans un format exploitable par les moteurs de workflows. Dans notre cas, il faudra prévoir un éditeur de processus métiers en permettant une représentation graphique qui pourra être exportée dans un format d'échange standardisé.
- Des applications clientes, à savoir des programmes permettant de piloter l'instanciation et l'exécution des workflows par les moteurs. Nous devons ainsi concevoir un composant permettant d'agir directement sur les moteurs afin de créer, démarrer, suspendre ou encore abandonner des workflows.
- Des appels applicatifs, permettant des interactions entre les moteurs de workflows et des applications externes. Pour pouvoir intégrer les processus métiers, en partie ou totalité, il est pour nous primordial que les moteurs puissent interagir avec différents SI, outils d'analyses et/ou de mesures.
- D'autres moteurs de workflows. Si notre système venait à être distribué, ce type d'interfaces nous permettrait de connecter entre eux plusieurs moteurs physiquement disséminés.
- Des outils d'administration et de monitoring des workflows, c'est-à-dire des applications destinées à la gestion même du service de workflows. De telles

applications sont indispensables pour garantir l'état du service ainsi que pouvoir contrôler l'utilisation qui en est faite et qui accède au service.

La prise en compte de ces nouveaux éléments nous permet d'identifier d'autres axes d'amélioration permettant de supporter le travail collaboratif de manière proactive (Nurcan, 1996). Plus précisément, certains éléments architecturaux, indispensables au bon fonctionnement de notre PAM, commencent à se dégager.

1) Permettre le télétravail et la télémedecine

Comme nous l'avons souligné (voir le problème **PROBA**) la mise en adéquation de multiples compétences et données est soumise à des contraintes spatiales et temporelles fortes. La télémedecine semble pouvoir permettre de s'affranchir de ces barrières (Collins et al., 2010). Le travail coopératif assisté par ordinateur dans la télémedecine est basé sur des modèles, des processus et des techniques qui permettent aux professionnels de la santé d'accéder et de transmettre de manière efficace l'information à toutes les parties associées à une quelconque prise en charge de santé (Porumb et al., 2010). Le recours de plus en plus fréquent à une telle pratique est inhérent à la géographie de la population (Breen & Matusitz, 2010). D'un côté la population tend à être répartie d'une manière assez inéquitable sur les territoires, et d'un autre côté la mondialisation a pour effet de rendre cette population très mobile (Kilman & Forslund, 1997). Ceci a pour conséquence de poser couramment des difficultés (Jovanov, Starčević, Samardžić, Marsh, & Obrenović, 1999) :

- D'accès aux dossiers des patients
- De disponibilité de praticiens qualifiés ou de spécialistes
- D'accès à un équipement médical sophistiqué

C'est dans l'optique de surmonter ces difficultés que l'on fait de plus en plus appel à des solutions de télémedecine. En effet, la qualité des soins dépend de la continuité des soins, que le recours de manière longitudinale à l'utilisation du dossier électronique du patient peut apporter (Glick & Moore, 2001; Kilman & Forslund, 1997). La mise en place de telle solution passe d'ailleurs par la mise en place d'infrastructures d'information, l'utilisation de matériels et le développement d'application pouvant s'exécuter en temps réel (Larson et al., 2012; Mandellos, Koutelakis, Panagiotakopoulos, Koukias, & Lymberopoulos, 2009). Il est ainsi créé un environnement virtuel permettant la collaboration entre de multiples cliniciens et les patients d'une façon indépendante de leurs emplacements physiques (Kilman & Forslund, 1997). Un tel environnement devant offrir un maximum de flexibilité pour l'acquisition et l'exploration des données, il paraît pertinent de vouloir proposer un framework (Johnson, 1997) facilitant la mise en œuvre ainsi que la capacité à évoluer du système. En effet, la tendance au découpage en composants logiciels, chacun responsable d'une tâche ou d'un service spécifique, permet d'avoir la liberté de les combiner de diverses manières afin de réaliser de nouvelles tâches ou services (Wangler & Paheerathan, 2000). Dans cette optique de réutilisabilité fonctionnelle et réagencement dynamique de composants logiciels, recourir à un framework semble complètement adapté. Ainsi, en nous appuyant sur les infrastructures des réseaux mondiaux actuels, nous allons pouvoir décrire des processus médicaux impliquant des acteurs géographiquement éloignés. On peut même préciser que, dans le cadre de l'établissement d'un diagnostic, si les données concernant le patient, contenues dans le SI ne sont pas suffisantes, alors une surveillance distante,

pouvant être en temps réel, peut être réalisée (Porumb et al., 2010). Une telle surveillance peut consister à relever des informations en provenance du patient (Wootton, 2001) ou inversement à lui envoyer des informations comme, par un exemple, un rappel concernant sa médication (Collins et al., 2010). Il devient alors plus facile d'assurer une prise en charge du patient conforme aux standards de soins (Magrabi, Lovell, Huynh, & Cell, 2001). Dans le cadre d'un trouble neuropsychologique, la télémédecine peut aussi permettre au patient de suivre un programme de remédiation à son domicile. Chez des enfants, il a d'ailleurs été montré, par exemple, que ce type d'entraînements peut être plus efficace à domicile qu'à l'école (Magnan & Ecalle, 2006). D'une manière générale, la télémédecine présente des avantages pour les cliniciens, les chercheurs et les patients en facilitant l'accès aux soins, augmentant leur qualité et réduisant simultanément les délais et les coûts (Mandellos et al., 2009; Matusitz & Breen, 2007). Ce genre de solution apporte des pistes pour résoudre les problèmes **PROB3** et **PROB6**.

Les possibilités offertes par ce type de système sont primordiales dans le cadre de la prise en charge des patients souffrant de dyslexie. En effet, leurs parcours de soin impliquent un grand nombre de professionnels différents, souvent situés dans des lieux différents, et pour lesquels les durées d'attente avant d'obtenir une entrevue peuvent être assez longues. L'introduction des fonctions de télémédecine peut donc ainsi contribuer à une prise en charge plus rapide et plus efficace. Pour finir, il paraît important de préciser que notre système, à travers ses fonctionnalités en télémédecine, ne se restreindra pas à faire de l'e-santé. Effectivement l'e-santé est une branche spécifique de la télémédecine qui a principalement pour but de communiquer et d'informer les praticiens et les patients, ou d'une manière générale le grand public, et d'étendre, hors du cadre de la seule application clinique, l'usage de la télémédecine (Matusitz & Breen, 2007). Notre système, lui, se veut définitivement apte à supporter l'exercice de la médecine à distance et permettre, le cas échéant, une interaction en temps réel entre praticiens et patients (Wootton, 2001). Pour finir, on signalera que 48% des personnes interrogées suite à notre enquête (voir l'annexe **Enquête auprès des parents d'enfants dyslexiques**) trouvent judicieuse la possibilité pour un praticien de pouvoir intervenir à distance avec un ordinateur (voir les réponses à la question 17).

D'autre part, le concept de travail coopératif assisté par ordinateur joue un rôle important dans l'enseignement et le processus de formation dans lequel les professionnels de la santé sont continuellement impliqués (Porumb et al., 2010). En effet, la télémédecine, à travers les structures dédiées à l'e-santé, peut permettre aux praticiens d'approfondir leurs connaissances (Matusitz & Breen, 2007). L'utilisation des nouvelles technologies peut ainsi conduire à placer le patient dans sa globalité au centre de cette nouvelle façon de soigner ainsi que d'apprendre (Glick & Moore, 2001). Le travail coopératif assisté par ordinateur s'impose ainsi comme une première étape vers une solution au problème **PROB6**.

2) Faciliter le développement et le déploiement de tests informatisés

Un constat se dégage nettement lorsque l'on commence à s'intéresser à la dyslexie : personne ne sait exactement ce que c'est, et on cherche plutôt à déterminer ce que ce n'est pas. Les connaissances à des fins thérapeutiques ont donc encore besoin d'être étoffées et la recherche évolue constamment dans ce domaine. Par conséquent, notre système doit être

adaptable et apte à intégrer les avancées de la recherche. A ce titre, il doit supporter les ajouts et modifications de son contenu, permettant ainsi aux chercheurs d'implémenter, tester et valider leurs protocoles conformément aux exigences posées par le problème **PROB2**. Dans un deuxième temps, les modèles logiciels de ces protocoles expérimentaux pourront être simplement transférés ou rendus accessibles aux praticiens pour un usage en milieu clinique répondant ainsi au problème **PROB3**. Ceci est encore un élément prônant le fait que notre PAM doit pouvoir gérer les processus métiers, dans la mesure où ces protocoles expérimentaux peuvent être assimilés à tout ou partie de tels processus. De la sorte, les données utilisées par ces différents processus peuvent être partagées (pour un exemple d'un tel usage consulter (Prokosch et al., 2011)). A ce titre les chercheurs doivent avoir accès à un éditeur (comme par exemple dans (Mejía, Clara, et al., 2011)), intégrant un outil de définition de workflows (similaire à celui de (Greenes et al., 1999)), assez abouti afin d'être utilisé pour décrire tous les composants de leurs expériences (interfaces graphiques, accès à des données, calculs, etc.). Pour tenir ces objectifs notre système doit ainsi être :

- Capable de gérer dynamiquement les métadonnées permettant la description des protocoles, afin de permettre l'implémentation et l'exportation
- Constitué de briques génériques, constituant les éléments permettant la réalisation de traitements pendant les protocoles

Le choix des métadonnées est particulièrement important pour que les données soient partageables, réutilisables et puissent faire l'objet de collaborations. Ceci doit s'appliquer aussi bien aux données relatives aux patients que celles servant à décrire les processus métiers impliqués dans des collaborations pluridisciplinaires (Cragin & Shankar, 2006). Notre PAM a donc besoin de reposer sur des standards de données et d'échanges de données afin d'atteindre un niveau d'interopérabilité essentiel à sa large adoption. En particulier dans les domaines de recherche sur la dyslexie, nous espérons mettre de la sorte à disposition des chercheurs un framework leur permettant de mettre au point des indicateurs physiologiques solides et utilisables en clinique afin de répondre au problème **PROB5**.

3) Mutualiser nativement des données multimodales

La maîtrise des processus de travail doit permettre une création de valeur aussi efficacement que possible, c'est-à-dire qu'elle doit conduire à supprimer les activités inutiles ou redondantes (Johannesson & Perjons, 2000). Dans le cas de la prise en charge des dyslexiques, une telle maîtrise passe par le fait d'éviter les doublons de bilans, d'examen ou d'une manière générale les collectes de données de nature fondamentalement similaire. Ce genre d'améliorations passe nécessairement par une mutualisation/partage des données concernant chaque patient (Martins-Baltar et al., 2013). D'autre part, il est reconnu que les troubles développementaux ne sont que très rarement spécifiques à une modalité (Karmiloff-Smith, 1998). Il est donc ainsi primordial de pouvoir analyser les différentes modalités potentiellement révélatrices de déficits (comme, par exemple, ceci est fait dans le projet READY (Trolès et al., 2008)). Un système qui se voudrait particulièrement efficace pour la prise en charge de ces patients pourrait donc procéder au recueil systématique de toutes les données provenant des différentes modalités potentiellement affectées par la pathologie. Effectivement, les procédures de diagnostic nécessitant fréquemment

différentes vues des mêmes données, l'acquisition et la présentation de données multimodales, et a fortiori des traitements effectués en parallèle, sont d'une grande valeur (par exemple, consulter (Jovanov et al., 1999; Mejía, Díaz, Jiménez, & Fabregat, 2011)). D'autre part, les mêmes données multimodales peuvent servir à plusieurs praticiens, de domaines différents, dont les interventions respectives sont nécessaires à la prise en charge du patient.

Pour finir, comme nous l'avons déjà évoqué, on pourra ajouter que recueillir de la sorte de grandes quantités de données peut être d'un intérêt non négligeable pour la recherche (Cragin & Shankar, 2006). En particulier, l'analyse massive de données multimodales peut conduire à la découverte de nouveaux indicateurs ou prédicteurs des pathologies, et faire écho au problème **PROB2**. Naturellement, ces données multimodales peuvent provenir de l'observation de caractéristiques comportementales aussi bien que physiologiques, et ainsi contribuer à établir des liens entre ces deux domaines médicaux/paramédicaux (voir le problème **PROB5**). Dans la sphère de la dyslexie, de telles découvertes seraient évidemment de premier ordre.

4) Proposer un outil de prise en charge et de suivi pluridisciplinaire

Grâce à la mutualisation des données, chaque praticien peut être informé de l'état du parcours de soin de son patient ainsi qu'avoir accès aux données relatives à ce dernier et produites lors de son parcours. Comme précédemment cité, ceci peut être techniquement assuré par, respectivement, un gestionnaire de workflows ainsi que par un SI centralisant les données des patients. L'association de ces deux types d'outils, appliquée au contexte médical, ouvre la voie à la mise en place de systèmes informatiques automatisant l'accès à la bonne information au bon moment afin d'induire de meilleures prises de décision (Dadam, Reichert, & Kuhn, 2000). Il est également remarquable que de tels outils numériques peuvent même permettre des collaborations à l'échelle internationale (Kilman & Forslund, 1997). Un système prenant en considération cela est apte à répondre au problème **PROB4**.

D'autre part, on notera que le concept d'un réseau professionnel peut être intégré dans un environnement virtuel collaboratif de santé, dès qu'on assimile l'ensemble des professionnels de santé le constituant à une équipe virtuelle. Les membres de cette équipe collaborent pour atteindre un ensemble d'objectifs, pour partager des ressources, des connaissances, des compétences, des équipements, des coûts, des risques et des bénéfices (Porumb et al., 2010). Une organisation ainsi intégrée informatiquement doit pouvoir procurer aux patients un sentiment de fluidité pendant leurs parcours de soins (Protti, 2009). Il faut donc particulièrement prêter attention à l'utilisation d'un dossier médical électronique qui, nécessitant une technicité particulière, peut compliquer la tâche du praticien et détourner son attention du patient (Makoul et al., 2001). Il est donc important que la manipulation de notre système soit le plus naturel possible afin de réellement améliorer la manipulation des informations relatives au patient, et non l'inverse. Il paraît également pertinent d'effectuer le recueil d'informations/données techniques auprès du patient d'une façon elle aussi naturelle, c'est-à-dire d'une manière la plus proche possible de celle qu'emploieraient un ou plusieurs autres êtres humains. La portée d'une telle remarque est double, étant donné qu'un système l'intégrant complètement devrait permettre :

- Le remplacement du « tout papier » au profit d'un SI informatisé.
- L'assistance technique au professionnel dans le cadre de sa pratique à travers un outil informatique analysant des données multimodales.

D'autre part, l'analyse multimodale de données complexes est un défi pour un être humain, qui ne peut réaliser que peu de tâches intellectuelles en parallèle (Pashler, 1993). L'ordinateur se propose ainsi comme étant une alternative à cette limitation humaine, dans la mesure où son utilisation en tant qu'outil permet intrinsèquement une parallélisation de tâches. Il peut donc particulièrement s'avérer utile dans l'établissement d'un diagnostic ainsi que dans le suivi d'un programme de remédiation.

5) Impliquer le patient dans une remédiation centrée autour de lui

On sait que de nombreux traitements médicaux ne sont efficaces que si le patient est prêt à en être partie prenante (Doherty, 1995). Il est donc primordial qu'il soit impliqué dans sa prise en charge et que, pour plus d'efficacité, il ait accès à des données du SI le concernant (National Transitions of Care Coalition, 2010; Weibel et al., 2013). D'autre part il est important que le patient soit suffisamment informé sur sa pathologie (Collins et al., 2010). Afin d'améliorer la communication entre le praticien et le patient, et d'impliquer ce dernier, notre système doit pouvoir fournir à ces différents acteurs une interface adaptée à la consultation et la mise à jour de données. La possibilité offerte aux patients d'agrémenter par eux-mêmes leurs propres dossiers peut d'ailleurs améliorer leur culture médicale et leur habilité à gérer leur propre santé (Kilman & Forslund, 1997; Peleg & González-Ferrer, 2014). D'une manière plus globale, on ajoutera qu'il est indispensable que le patient ait un droit de regard et de décision en ce qui concerne son traitement, et a fortiori qu'il ait accès aux informations s'y rapportant (Mechanic, 2008). Ainsi, un SI accessible via Internet peut être un bon moyen pour engager significativement le patient dans sa propre prise en charge. D'autre part, dans un contexte académique, ce genre d'approche permet par exemple d'inciter l'étudiant atteint de trouble des apprentissages à la réflexion sur son apprentissage et à l'autorégulation (Mejía, Bull, Vatrappu, Florian, & Fabregat, 2012). Dans le cas de la prise en charge de dyslexiques, la motivation des enfants semble d'ailleurs être un élément d'importance pour expliquer les variations d'amélioration de leurs performances au travers d'un entraînement (Magnan & Ecalte, 2006). Ces considérations, relatives aux moyens d'engager le patient dans son traitement, doivent aussi être étendues à sa famille. En effet, il est prépondérant que la famille du patient l'assiste dans ses démarches de santé (Doherty, 1995). Un système intégrant ces éléments tend ainsi à apporter des réponses à **PROB6**. On pourra d'ailleurs ajouter que les patients semblent ne pas faire de rejet de la télémédecine et être plutôt contents du matériel qu'on peut être amené à leur confier à cette occasion (Wootton, 2001). Des études ont d'ailleurs montré que cette technique, en particulier au travers d'interactions en temps réel, offre généralement satisfaction aux patients (Allen & Hayes, 1994). La satisfaction des patients est un facteur clef permettant d'envisager l'utilisation de la télémédecine dans la vie de tous les jours (Mandellos et al., 2009). L'utilisation de DMP semble aussi être accueillie positivement par les praticiens et les patients (Delpierre et al., 2004).

Cependant, mettre l'informatique entre les patients et les praticiens présente un risque de désengagement du patient si ce dernier pense que le clinicien n'a pas son attention portée

sur lui (Frankel et al., 2005; Makoul et al., 2001; Weibel et al., 2013). Il convient donc de concevoir la solution de telle sorte qu'elle s'imisce entre ces acteurs d'une façon constructive, qui soutienne leur interaction plutôt qu'elle ne la parasite. Les interactions entre ces acteurs étant de nature multimodale (Obrenović & Starčević, 2004), notre solution devra pouvoir supporter la multimodalité des interactions d'une manière relativement naturelle (Oviatt & Cohen, 2000).

L'utilisation de données des patients précises et comprises de ces derniers est le premier pas pour pouvoir efficacement personnaliser les soins qui leur sont apportés (Isern, 2008). En effet, s'adapter précisément aux besoins et/ou déficit des patients est primordial afin d'être efficace. A ce titre, pour assurer une remédiation individu-centrée, il va falloir conserver pour chaque utilisateur les valeurs d'un ensemble de paramètres, c'est-à-dire gérer des profils (comme par exemple dans (Mejía Corredor & Fabregat Gesa, 2012)). Notre système doit ainsi pouvoir profiler ses utilisateurs sur la base des données fournies par le SI. On pourra aussi signaler que, dans notre enquête (voir l'annexe **Enquête auprès des parents d'enfants dyslexiques**), les familles des patients semblent très fortement favorables à l'utilisation de logiciels de remédiation (voir les réponses à la question 14).

III) La division en sous-problèmes

Réaliser un système informatique permettant d'assister la prise en charge de patients à tous les niveaux de celle-ci n'est pas une chose simple. Effectivement, une telle prise en charge est d'une grande complexité, ce qui implique qu'un système tentant de l'intégrer dans son intégralité sera lui aussi complexe. Une approche holistique du problème est donc à exclure au profit d'une approche cartésienne et systémique. En effet, la complexité d'un système peut être gérée à l'aide de (Booch et al., 2007) :

- La décomposition, consistant à diviser un vaste problème en sous-problèmes.
- L'abstraction, consistant à ne modéliser que les détails et propriétés pertinentes.
- La hiérarchisation, consistant à définir et gérer les relations entre les sous-systèmes.

En décomposant le problème global ainsi en sous-problèmes cloisonnés – dont les entrées, sorties et interactions sont connues – nous avons l'intention d'en réduire la complexité et en faciliter la modélisation (Jennings, 2001; Wangler & Backlund, 2005). Par conséquent, nous nous efforcerons, aussi bien dans le choix d'une méthode de développement que dans celui de la structure de notre système, de clairement définir les données et traitements associés à chaque étape du développement et/ou composant de la PAM. Le choix de la méthode de développement fera d'ailleurs l'objet d'une partie dans la suite de ce manuscrit (voir le chapitre **Le choix d'un modèle de développement**). La modélisation des différents composants et traitements sera elle aussi détaillée ultérieurement (voir la section **L'architecture du système**). Cependant nous pouvons dès à présent tracer une ébauche du dispositif et des différents éléments le composant. Cette esquisse est principalement dirigée par la nécessité d'intégrer horizontalement et verticalement les processus impliqués dans l'organisation de la prise en charge des patients (Wangler & Paheerathan, 2000). Dans les faits notre système pourra se composer d'une partie logicielle, réunissant toutes les fonctionnalités d'un SI, et d'une partie matérielle, rassemblant des équipements d'analyse intégrés par la partie logicielle. Ainsi, une fois en voie d'industrialisation, notre dispositif devrait se composer :

- D'une plate-forme thérapeutique – le noyau du système – constituée :
 - D'un outil de travail collaboratif, qui est la partie du dispositif intégrant horizontalement les processus de travail intra et inter-organisationnels, aussi respectivement appelés processus métiers privés et publics (Ko, 2009), c'est-à-dire servant principalement de SI. En particulier ce composant sert à piloter l'exécution de tous les processus de travail intégrés.
 - D'un support d'intégration intra-organisationnelle verticale et multimodale, qui est formé des briques logicielles assistant les praticiens dans l'acquisition et le traitement (potentiellement synchrone) des données patients. Ce composant a pour but de fournir une abstraction du matériel utilisé, un middleware, pour la réalisation de processus métiers.
 - D'un utilitaire de gestion des processus, permettant de contrôler l'implémentation des processus intégrés aussi bien horizontalement que verticalement. Il s'agit d'un composant offrant une interface graphique facilitant la définition/modélisation des processus métiers pour les non-informaticiens (Aigner & Miksch, 2006; Huser et al., 2011; Klinker, Bhola, Dallemagne, Marques, & McDermott, 1991; Pattison-Gordon et al., 1996).

- D'une interface multimodale, étant l'interface tangible/physique permettant des interactions les plus naturelles possibles entre le système et le patient. Cette interface pourra aussi bien être utilisée pour les examens diagnostiques qu'à des fins de compensation. On remarquera également qu'elle comprend, entre autres, les Entrées/Sorties classiques d'un ordinateur personnel que sont le clavier, la souris et un moniteur par exemple, et qu'elle se conforme aux standards d'interopérabilité de manière à être exploitable par le support d'intégration verticale.
- D'un contenu spécifique au traitement de la dyslexie, c'est-à-dire la description des processus spécifiques à la prise en charge de la dyslexie qui, par instanciation, permettront d'assister le travail de chaque corps de métier impliqué (intégration verticale) ainsi que la coordination/coopération entre eux (intégration horizontale). Ce contenu est naturellement éditable par l'utilitaire de gestion des processus.

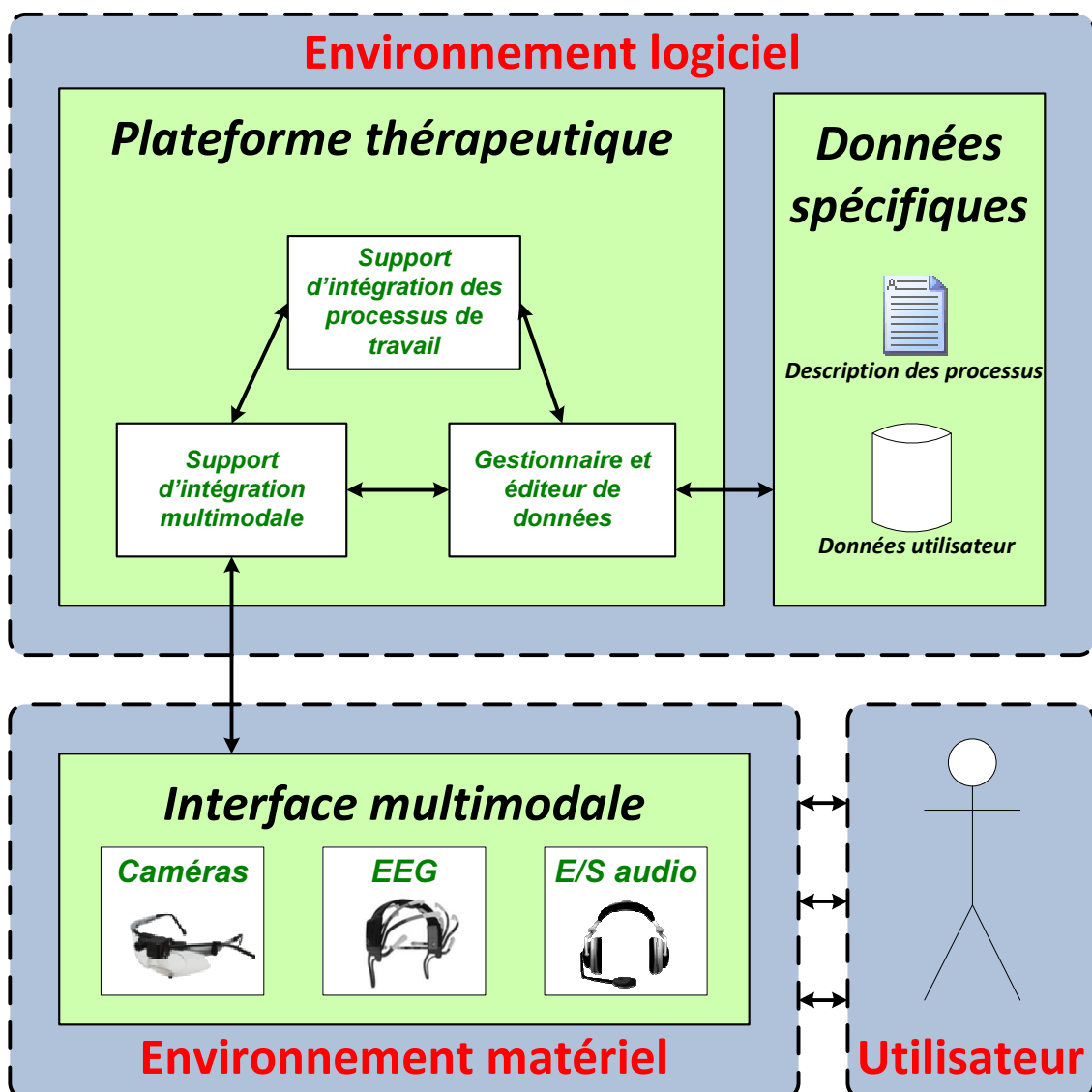


Figure 9 : Schéma d'ébauche de l'architecture informatique du dispositif

Cette ébauche va nous permettre de nous concentrer sur la relation qu'un utilisateur aura avec un système disposant de toutes ces fonctionnalités. Il nous sera aussi plus simple d'identifier des cas d'utilisation cohérents avec les résolutions de problèmes telles que nous

venons des les envisager. La **Figure 9** propose une vue d'ensemble de ce que nous venons d'énoncer.

De plus, notre système étant destiné à impliquer de multiples utilisateurs qui se trouveront en toute vraisemblance à de multiples endroits distincts et qui solliciteront différents services offerts par la solution, il paraît logique d'affirmer qu'il utilisera un réseau pour fonctionner. Ainsi notre dispositif devra être conçu pour pouvoir fonctionner en mode client-serveur étendu, c'est-à-dire pour supporter la communication entre ordinateurs et programmes à travers ce réseau (Liang, Lai, Chen, Wei, & Chen, 1994). Finalement, notre système intégrera effectivement toutes les tendances de l'intégration, telles que précédemment énumérées (Hatcliff et al., 2012; Larson et al., 2012).

Le processus de développement d'un système implique un processus de résolution de problèmes au niveau macroscopique et une méthodologie scientifique au niveau microscopique (Alhir, 2002). Au cours de cette thèse, nous allons donc nous efforcer de traiter des éléments relevant de chacune de ces deux échelles. En effet, d'un côté nous mettrons en place les fondations d'un processus de gestion du projet, et de l'autre nous détaillerons des points techniques demandant une recherche pointue. Nous allons donc maintenant, dans l'optique d'éprouver la capacité de notre système à être industrialisé, nous fixer des objectifs attendant à chacun ces deux domaines.

IV) Les objectifs fixés

Afin de conduire notre projet, nous nous fixons plusieurs desseins à atteindre. En effet, nous allons réaliser des descriptifs et des preuves de concept des différents éléments constitutifs du système, de sorte que chaque modèle ou prototype ainsi développé satisfera un but déterminé. De cette façon, nous comptons démontrer la faisabilité du dispositif tout en nous confrontant à plusieurs des difficultés s'opposant à cette réalisation. En particulier, la réalisation de prototypes, qui est largement pratiquée (Neill & Laplante, 2003), permettra de vérifier l'utilisabilité de certains aspects du système avec ses futurs utilisateurs (Beaudouin-Lafon & Mackay, 2007; Ferré, Juristo, Windl, & Constantine, 2001). Certains de nos objectifs nous amèneront à mettre à l'épreuve le processus de modélisation de l'architecture du futur système, alors que d'autres testeront directement les méthodes scientifiques de traitement de données à un plus bas niveau. Cette diversité, cette hétérogénéité au sein des productions que nous visons, aura pour avantage de nous faire manipuler des parties du projet de différentes granularités. Une telle démarche ne peut être que plus formatrice et nous amener à une meilleure compréhension de la problématique dans sa globalité : c'est ce que l'on pourrait appeler le méta-objectif de notre travail de thèse.

La partie centrale du système, le cœur même du dispositif, est la plate-forme thérapeutique. Il s'agit de la partie du système requérant le plus de travail car elle rassemble l'intégralité des fonctionnalités primordiales du dispositif. En particulier, la réalisation des composants du SI peut être très chronophage et nécessiter l'intervention à plein de temps de plusieurs personnes pendant des mois, voire des années. Le développement de cette plate-forme dans son intégralité n'est donc clairement pas réaliste dans le temps imparti d'un projet de thèse. Nous comptons donc nous limiter à la proposition d'une modélisation générique de cette portion du système. Le modèle ainsi proposé devra représenter les interactions entre le système et ses utilisateurs ainsi que les types de données manipulés à cette occasion. La réalisation de cette modélisation sera d'ailleurs le premier objectif que nous nous fixons.

Parmi les données destinées à être manipulées par la plate-forme thérapeutique, on trouvera, non-exhaustivement, des données des patients issues des modalités de suivi oculaire, d'électroencéphalographie et d'analyse de la parole. L'intégration de cette dernière modalité, relative à l'étude de la voix du patient, peut s'appuyer sur grand nombre de produits (gratuits et/ou payants) accessibles à tout un chacun à un prix relativement modique pour quiconque dispose d'un microphone de qualité correcte. Nous n'essaierons donc pas de redévelopper par nos propres moyens des composants d'analyse vocale. A contrario, les deux autres modalités présentent une toute autre limitation : le prix du matériel nécessaire à l'acquisition des données est généralement exorbitant, voire prohibitif. Nous souhaitons donc trouver des matériels bon marché nous permettant de procéder à la capture et l'analyse de ces données, et si besoin nous développerons nous-mêmes les composants logiciels en permettant l'intégration à notre programme. Cette démarche aboutira par la validation d'objectifs spécifiques à chacune de ces deux modalités.

Pour finir, nous devons nous assurer que notre système permette, dans le domaine des troubles neuropsychologiques, de conduire convenablement un examen médical. A ce titre, nous comptons implémenter une épreuve extraite d'un protocole d'évaluation des capacités en lecture. Ceci va nous permettre d'évaluer la propension de notre système à fonctionner

correctement dans des conditions proches de celles dans lesquelles il est voué à être déployé. Cette dernière évaluation sera le dernier objectif que nous nous fixerons.

Afin de bien garder à l'esprit les objectifs que nous nous sommes fixés, et pouvoir facilement y faire référence ultérieurement, nous nous proposons de les récapituler dans le **Tableau 2** :

Référence de l'objectif	Descriptif de l'objectif
MODEL1	Modélisation du système informatique en mettant en avant la généralité des interactions et des données manipulées.
PROTO1	Implémentation d'un composant logiciel de suivi oculaire utilisant du matériel bon marché.
MODEL2	Rédaction d'un cahier des charges d'un matériel bon marché pour le suivi oculaire.
PROTO2	Implémentation d'un composant logiciel d'acquisition de données EEG à partir d'un matériel bon marché.
PROTO3	Implémentation en environnement immersif d'un test d'évaluation de la lecture reconnu et extrait de la littérature.

Tableau 2 : Récapitulatif des objectifs fixés

Afin de nous atteler à l'accomplissement de notre premier objectif, nous devons choisir un processus de développement logiciel. C'est-à-dire déterminer quel ensemble d'activités il nous faut mener pour transformer les exigences de nos futurs utilisateurs en un système informatique (Khamis & Abdelmonem, 2002; Obrenović & Starčević, 2006). A cette fin, nous allons passer en revue les différents modèles couramment utilisés et en adopter un, correspondant à nos besoins, que nous mettrons en œuvre par la suite.

L'ingénierie logicielle

Avant de vouloir développer une solution informatique, qu'elle soit matérielle ou logicielle, nous allons traiter de la méthodologie mise en œuvre afin d'y parvenir. En effet, tout développement doit être précédé d'une phase de réflexion, de conception, et la manière d'appréhender cette étape préliminaire à la réalisation se doit d'être raisonnée si l'on souhaite maximiser nos chances d'amener le projet à son terme. La mise en œuvre de telles méthodes est partie intégrante de l'ingénierie logicielle (Ramsin & Paige, 2008). A ce titre, nous allons nous intéresser dans cette partie aux différents types de modélisations impliqués dans l'ingénierie des systèmes (Kossiakoff, Sweet, Seymour, & Biemer, 2011; Wangler & Backlund, 2005), et plus particulièrement à l'ingénierie logicielle. Chaque notion ainsi abordée sera ensuite projetée dans le contexte de notre projet, c'est-à-dire que nous illustrerons notre description théorique de l'ingénierie logicielle par l'étude pratique du développement de notre projet. Nous espérons à cette occasion pouvoir faire, lors des phases en amont de l'implémentation logicielle, des choix méthodologiques les plus pertinents possibles. Cette étude va d'ailleurs nous permettre de diriger le développement des preuves de concept de la solution informatique que nous envisageons, ce qui dans le cas d'un système inexistant est indispensable (Royce, 1970). Un tel prototypage permet de prendre en considération le fait que les utilisateurs, leurs besoins et leur environnement évoluent durant le processus même de développement (McCracken & Jackson, 1982). En effet, l'évolution et la maintenance d'une solution informatique sont primordiales (Balzer, Cheatham, & Green, 1983; Mathur & Malik, 2010) et les négliger pourrait mener à l'échec du développement. Les méthodes de modélisation occupent une place palpable dans le développement logiciel, donc nous nous efforcerons dans le présent chapitre de donner un certain nombre de références à des ouvrages techniques destinés principalement à un public de techniciens et/ou d'ingénieurs. Effectivement, l'aspect très concret de notre projet, lié au contexte de transfert technologique de la thèse, nous pousse à nous intéresser à une littérature à la fois spécifique et pratique pour traiter de la problématique du génie logiciel. La rationalisation du développement logiciel est une discipline scientifique qui a débuté il y a plusieurs décennies : nous citerons des publications qui peuvent paraître un peu anciennes de prime abord mais qui n'en restent pas moins pertinentes.

La résolution d'un problème passe par sa compréhension et sa conceptualisation en le représentant et l'interprétant. De la sorte, il est possible de résoudre ce problème en manipulant la représentation afin d'en déduire ou d'en définir la solution (Alhir, 2002). La façon de procéder à cette représentation est communément appelée modélisation. La modélisation consiste en la proposition d'un modèle représentant un phénomène et possédant des propriétés analogues à ce dernier. Avec l'expérimentation, il s'agit de l'une des principales composantes de la démarche scientifique (Treuil, Drogoul, & Zucker, 2008). D'ailleurs, comme le dit avec justesse Guérin (2012) lorsqu'il traite de la conduite de projets informatiques, la modélisation est un processus qui permet de traiter au mieux un sujet en favorisant la réutilisation et la généricité. En informatique, la modélisation est impliquée dans toutes les étapes du développement d'un projet, c'est-à-dire dans (Noël, 2012) :

- La formulation de la problématique (voir la partie **La problématique**).
- La conception algorithmique et logicielle, apportant une solution au problème (voir la partie **Les développements**).

- La planification de la réalisation du produit, c'est-à-dire le processus de développement de la solution retenue (qui fait l'objet de la présente partie).

Une méthodologie de modélisation encourageant à la généricité doit ainsi fournir des outils permettant d'accomplir avec succès ces différentes étapes et de résulter en la production de livrables facilement évolutifs et potentiellement réutilisables dans d'autres contextes (Fortune & Valerdi, 2013). L'impératif de généricité est établi afin de favoriser la réutilisabilité et l'évolutivité des développements pour les rentabiliser le plus possible (Jalender et al., 2012), cependant il faut bien avoir conscience que ce n'est pas une chose facile à réaliser (Garlan, Allen, & Ockerbloom, 2009). On peut, d'autre part, préciser qu'un projet est une vision instantanée du processus visant à produire de tels livrables (matériels ou immatériels). Conduire un tel projet revient à maîtriser ce processus en s'appuyant sur un ensemble de techniques précises et éprouvées (Guérin, 2012; Morley et al., 2011). Ainsi, opter pour une méthodologie de suivi de projet revient à choisir sur quelles techniques nous allons nous appuyer. Ces différentes techniques sont régies par des modèles, des abstractions de la réalité, qui agissent selon les trois dimensions structurant la gestion de projets :

- L'axe temporel de développement
- L'axe d'analyse du système
- L'axe de pilotage des processus

Les modèles de développement déterminent explicitement l'organisation temporelle du projet, son cycle de vie. En effet, une méthode de développement fait référence à un framework que l'on utilise pour structurer, planifier et contrôler le processus de développement. D'une manière générale, l'utilisation d'un framework permet d'améliorer la productivité de ceux qui l'utilisent, ainsi que la qualité, les performances, la fiabilité et l'interopérabilité de ce qu'il permet de produire (Fayad & Schmidt, 1997; Johnson, 1997). Les fonctions primordiales de la modélisation des processus de développement logiciel sont de déterminer les étapes impliquées dans le développement et l'évolution du logiciel, d'établir l'ordre dans lequel elle sont sollicitées et d'établir quels sont les critères de transition pour passer de l'une à l'autre (Boehm, 1988). Chacune des méthodes existantes est destinée préférentiellement à certains types de projets particuliers répondant à diverses considérations techniques et organisationnelles, liées au projet en lui-même ou encore à l'équipe de développement. Ces modèles sont des représentations abstraites du processus de développement qui méritent que nous en évaluions les avantages et les inconvénients afin de choisir la mieux adaptée à la situation (Centers for Medicare & Medicaid Services, 2008; Munassar & Govardhan, 2010). A ce titre il est pertinent d'examiner, parmi les modèles les plus répandus, ceux dont nous pouvons disposer :

- Le modèle « code-and-fix » qui consiste à simplement coder un programme et successivement le corriger jusqu'à ce qu'il satisfasse des objectifs fixés.
- Les modèles en étages, appelés aussi modèles linéaires ou nominaux, et en cascade dans lesquels un certain nombre d'étapes successives doit être réalisé dans un ordre prédéfini (comme par exemple celui décrit par Benington en 1987 ou celui proposé par Royce en 1970).
- Le modèle en V, qui consiste en un modèle en cascade amélioré où des tests de contrôle anticipé sont ajoutés au processus (pour une version améliorée de ce modèle, consulter (Mathur & Malik, 2010)).

- Les modèles itératifs comme le modèle évolutif (qui encourage à la réalisation itérative de prototypes/logiciels dont le concept est décrit par McCracken et Jackson en 1982), le modèle de transformation (basé sur la génération automatique de code à partir de spécifications (Balzer et al., 1983)), le modèle en spirale (Boehm, 1988) ou encore les modèles orientés-objets tel que l'Unified Process (Jacobson, Booch, & Rumbaugh, 1999).

La complexité des modèles proposés ci-dessus augmente au fur et mesure que l'on descend dans cette liste. Selon Boehm (1988), un processus de développement logiciel raisonné se fait de façon itérative afin de prendre en compte, dans la démarche de développement, le facteur de risque. Le facteur de risque est simplement la possibilité que le système puisse ne pas fonctionner du tout ou ne pas fonctionner comme ses utilisateurs le voudraient consécutivement au développement (Hijazi, Khmour, & Alarabeyyat, 2012). La gestion des risques est une partie intégrante du processus d'intégration (Madni & Sievers, 2014). D'autre part, plus les modèles précédemment cités sont en fin de notre énumération, plus ils prennent en compte le risque lié au développement, ainsi ils conduisent à des projets de plus en plus compliqués à mettre en œuvre, longs et coûteux à réaliser. Cette complexification se fait heureusement au profit de la diminution des chances d'échec du projet en question. Dans le cas du développement d'une application médicale, la prise en considération des risques, en termes de sûreté pour les utilisateurs, liés à son usage est aussi à prendre précautionneusement en compte (Fox, 2002). Dans le premier chapitre de cette partie, nous comparerons brièvement les avantages et inconvénients des différents modèles de développement que nous venons de présenter, et nous en retiendrons un pour notre propre développement (voir le chapitre **Le choix d'un modèle de développement**).

Les modèles d'analyse, qui dans les faits sont naturellement imbriqués dans ceux liés au développement, permettent de définir les contours du futur système puis d'en décrire les interactions avec ses utilisateurs. Parmi les célèbres modèles d'analyse, nous pouvons par exemple citer ceux décrits dans :

- La méthode IDEF0, qui sert à la modélisation et au développement de SI (IEEE Computer Society, 1998).
- L'analyse Merise, qui est particulièrement adaptée aux bases de données relationnelles (Nanci & Espinasse, 2001; Tardieu, Rochfeld, & Colletti, 2000).
- Le langage UML, qui est polyvalent et peut très bien être utilisé de façon complémentaire à d'autres modèles (Object Management Group, 2014e).

Les modèles d'analyse proposés par ces méthodes et langages permettent de décrire la structure et le comportement de systèmes en tout ou partie. Il est attendu de leur usage qu'ils renseignent l'équipe de développement, et les futurs utilisateurs, sur le comportement attendu du système. On signalera qu'il est également possible de combiner plusieurs de ces modèles afin d'arriver au niveau d'expression souhaité (Dorador & Young, 2000). Le choix de tels modèles, adaptés à nos besoins spécifiques, fera aussi l'objet d'un chapitre dans cette même partie (voir le chapitre **Le choix des modèles d'analyse**).

Les derniers modèles dont nous allons parler, dans le contexte de la conduite de projets informatiques, sont les modèles de pilotage. Contrairement aux deux types précédemment énumérés, qui constituent littéralement le processus de réalisation du projet, ces derniers ont pour but de contrôler l'évolution de ce processus. Ils visent, en particulier, à s'assurer

que le développement respecte les contraintes en termes de qualité et de coûts établies au préalable. Effectivement, le développement d'un tel projet est régi par un certain nombre de contraintes. Ces contraintes vont permettre de juguler les différents moyens déployés dans le cadre du développement logiciel. Nous nous proposons ici de lister les contraintes les plus rencontrées (Guérin, 2012) :

- Le cahier des charges, servant à la fois au recueil fonctionnel et aux spécifications techniques, qui sont autant de besoins que le produit doit satisfaire.
- La maîtrise des coûts, c'est-à-dire le respect du budget alloué au projet.
- La maîtrise de la qualité, correspondant à s'assurer de la satisfaction générale des utilisateurs à l'égard du produit.
- La maîtrise des délais, qui se traduit par une nécessité de planification des différentes activités composant le processus de développement de la solution.
- La matrice d'ajustement, cherchant à permettre la prise en compte et la pondération de différents sous-objectifs poursuivis à travers le développement.

A ce titre, on précisera que, dans le cas de la matrice d'ajustement d'un projet de recherche, certains éléments sont fréquemment laissés libres pour ne pas restreindre le périmètre fonctionnel exploré par le projet (Guérin, 2012). Nous ajouterons aussi que la mise en œuvre de modèles de pilotage ne peut avoir lieu que si le processus de développement est exécuté dans son intégralité et engage tous les acteurs potentiellement nécessaires à sa réalisation. Dans le cas du développement de notre PAM, nous ne réalisons qu'une modélisation du système, l'implémentation de la partie « intégration horizontale » ne sera pas entamée. Nous n'avons donc que trop peu d'éléments pour supporter une telle modélisation, et nous nous affranchirons donc d'étudier plus en détail ces modèles de pilotage.

D'une manière générale, on constatera que les différents types de modèles, que nous avons évoqués, relèvent de l'utilisation de frameworks. De tels frameworks sont des méta-modèles, c'est-à-dire des modèles de langages de modélisation permettant l'expression de modèles dynamiques (Treuil et al., 2008). Dans le cas présent, les modèles ainsi définis sont ceux de développement, d'analyse ou de pilotage. Ils permettent respectivement d'explicitier, afin de répondre aux besoins d'un projet particulier, les processus de développement, les représentations fonctionnelles et techniques du système, et finalement les moyens de contrôle de la qualité et des coûts du projet. Il est ainsi possible d'identifier un méta-modèle de modélisation générique régissant cette relation entre méta-modèles, modèles et instances de modèles (**Figure 10**). Naturellement, ce méta-modèle est un framework, c'est-à-dire une sorte de « boîte à outils » permettant la définition de modèles. Ces modèles sont des représentations théoriques de systèmes qu'il est possible de rendre plus concrets et fonctionnels au travers d'instances. On ajoutera également qu'un méta-modèle peut permettre de générer différents modèles, et que chaque modèle peut conduire à plusieurs instances différentes. D'autre part, un modèle est déjà une instance d'un méta-modèle. Ceci s'applique aux différents types de modélisation que nous allons être amenés à manipuler, c'est-à-dire pour le développement du système, pour son analyse et même dans le cadre de l'utilisation attendue du framework sur lequel s'appuiera la PAM que nous comptons modéliser. Ainsi, nous reviendrons sur ces points de notre modélisation plus tard dans notre exposé.

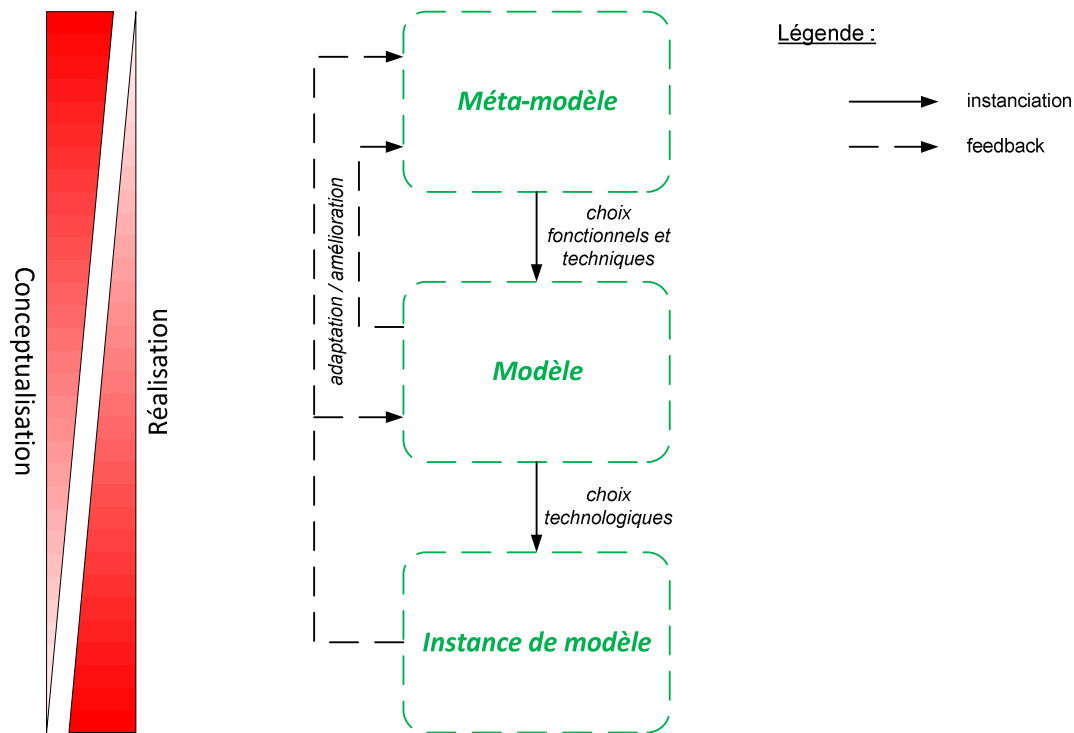


Figure 10 : Représentation de la relation entre méta-modèle, modèle et instance de modèle

Quel que soient les modèles utilisés, la gestion du développement d'un projet informatique est impossible sans suffisamment de documentation (Royce, 1970). Ainsi, nos différentes réflexions, qui seront présentées dans les chapitres suivants, feront l'objet de la production d'une documentation dont la vocation est d'expliquer le fonctionnement du processus de développement logiciel ainsi que celui du logiciel. La réalisation de ces documents constituera la première étape de notre développement (voir l'objectif **MODEL1**). En particulier, les informations y figurant devront permettre de :

- Fixer clairement les différents sous-niveaux d'objectifs à atteindre, et en contrôler l'achèvement.
- Fournir des spécifications sur lesquelles le design du produit va s'appuyer.

Nous soutenons qu'ainsi, à chaque étape du développement du projet, la documentation sera véritablement un support statique de l'état des différents modèles de développement et d'analyse utilisés. De cette manière, étudier ces différents modèles revient à produire la documentation spécifique à ces derniers.

I) Le choix d'un modèle de développement

Il est temps pour nous de choisir un modèle de développement et pour cela nous devons apprécier lequel serait le mieux adapté à notre projet. En effet, avant de vouloir maîtriser les processus de corps de métiers des milieux médicaux, il faut déjà être capable de maîtriser les processus de son propre corps de métier (comme on peut par exemple l'observer avec les travaux conduits dans (Framinan et al., 2005)). Notre but maintenant va être de déterminer une méthode adéquate à l'expression de nos propres processus de développement. Mener à bien un développement consiste à satisfaire aux besoins de ses utilisateurs finaux tout en franchissant les obstacles à ce développement, que nous appellerons les risques (Hijazi et al., 2012). Il faut bien garder à l'esprit qu'un système ne répondant pas correctement aux besoins de ses utilisateurs est un système voué à ne pas être utilisé. Afin de procéder au choix d'une méthode de développement, nous allons maintenant estimer, en nous appuyant sur des descriptions et des comparatifs des différents modèles (Centers for Medicare & Medicaid Services, 2008; Munassar & Govardhan, 2010), les contraintes et/ou avantages majeurs que peut avoir leur utilisation respective dans le contexte de notre projet :

- Le modèle « code-and-fix » est le modèle le plus archaïque, il ne consiste qu'à coder rapidement et corriger les problèmes rencontrés plus tard. L'utilisation d'un tel modèle permet de s'affranchir quasiment de toute réflexion préalable, ce qui n'est clairement pas adapté à un projet complexe comme le nôtre.
- De leur côté les modèles en cascade ont pour avantage d'être simples à appréhender et à mettre en œuvre, ce qui leur a valu d'être largement utilisés (Neill & Laplante, 2003). Cependant, ils sont plutôt rigides et ne produisent du logiciel qu'en fin de processus. Ceci les rend uniquement adaptés si les objectifs du projet, et les solutions retenues pour y arriver, sont claires dès le début. Ainsi, les éventuelles erreurs n'étant réellement visibles qu'à la fin de la procédure de développement (Hijazi et al., 2012), ces modèles s'accommodent assez peu d'une activité de recherche. Pour les mêmes raisons, ces modèles rendent impossible la prise en considération de contraintes d'utilisabilité (Ferré et al., 2001).
- Le modèle de développement en V est relativement similaire aux modèles en cascade mais il procède à davantage de tests en cours de processus. Ceci a pour effet de minimiser les risques et augmenter d'autant les chances de succès du projet (Hijazi et al., 2012). Cependant le développement y est effectué de manière monolithique, ce qui manque de souplesse et est donc peu adapté aux systèmes complexes.
- Les modèles itératifs, quant à eux, bien que nombreux et tout de même différents les uns des autres, présentent certaines caractéristiques communes intéressantes. En effet, ils intègrent continuellement une analyse des risques durant le développement en favorisant, entre autres, l'implication des futurs utilisateurs ainsi que le travail collaboratif au sein de l'équipe. D'ailleurs, la majorité des modèles de développement s'accordent sur le fait qu'il faut impliquer les futurs utilisateurs tout au long de la conception jusqu'à la réalisation des tests finaux, afin de maximiser la satisfaction du client vis-à-vis du logiciel (Gabay & Gabay, 2008; McCracken & Jackson, 1982; Royce, 1970). Dans notre contexte d'application, il est par exemple primordial qu'un médecin porte le changement auprès de ses collègues et servent véritablement de référent en ce qui concerne les pratiques cliniques (Levick & Osheroff, 2014; Naylor et al., 2011). D'autre part, et en particulier s'il implique du

prototypage, une méthode itérative de développement est à préconiser si les utilisateurs finaux n'ont pas une idée précise du genre de logiciel auquel le projet doit aboutir. Ceci maximise les chances d'éviter les problèmes d'inadéquation de l'architecture logicielle (Garlan et al., 2009) ou de l'interface utilisateur (Bouchet, 2006). Ces méthodes de développement semblent ainsi plus flexibles et adaptées à notre activité de recherche, régies par une certaine incertitude, mais aussi plus coûteux et difficiles à maîtriser que les modèles linéaires. Effectivement, il n'est pas possible de planifier le nombre d'itérations nécessaires pour atteindre les objectifs initialement fixés. Cependant, chaque nouvelle itération permet d'affiner incrémentalement la compréhension du problème, de déboucher sur un exécutable et d'améliorer le produit (Jacobson, 2008b; Rational Software, 2001). Ceci est par exemple illustré dans le contexte du développement de jeux vidéo lorsque Schell (2010) énonce la règle de la boucle : « plus vous passez de temps à tester et améliorer votre concept, et plus votre jeu sera bon ». Certaines de ces méthodes sont axées sur la réactivité et la flexibilité du processus, reprenant ainsi la philosophie du modèle code-and-fix, et sont qualifiées de « méthodes Agiles », comme par exemple le framework de processus de développement qu'est Scrum (pour un guide pratique consulter (Aubry, 2013)).

Dans le contexte de notre problématique, nous avons besoin d'un modèle de développement pour modéliser notre PAM (voir l'objectif **MODEL1**). Notre domaine d'application, c'est-à-dire la prise en charge de la dyslexie, va nous permettre d'extraire des instances de modèles d'analyse du fonctionnement attendu de la PAM. De tels modèles pourront, par la suite, nous servir de spécifications, de documentations, permettant l'implémentation du système. C'est pourquoi, parmi les méthodes itératives et incrémentales existantes, nous avons choisi d'utiliser l'**Unified Process** (Jacobson et al., 1999)(aussi appelé Unified Software Development Process) qui a pour avantage d'être relativement récent, de bénéficier de nombreuses implémentations (telle que **Rational Unified Process** d'IBM (Rational Software, 2001)), d'être complet, d'être bien documenté, et surtout de s'appuyer sur le langage de modélisation reconnu, répandu et maintenu qu'est UML (pour un ouvrage technique traitant de l'utilisation de ce langage, consulter par exemple (Gabay & Gabay, 2008)). D'autre part, UP permet de générer beaucoup de documentation (comme, par exemple, son prédécesseur le « 4+1 view model » (Kruchten, 1995)), ce qui lui confère un intérêt certain pour la réalisation de ce manuscrit, contrairement à une méthode Agile qui ciblerait la production d'exécutables comme priorité. Effectivement, les méthodes Agiles sont plus adaptées au développement d'applications reposant sur notre framework (pour une proposition d'utilisation de ces méthodes à cette fin, voir la section **La modélisation des processus métiers**) que pour le développement du framework en lui-même. L'UP est un méta-modèle de développement, aussi nommé framework de développement, qui a été mis au point et affiné par Jacobson, Booch et Rumbaugh au cours des dernières décennies. Ce modèle précise les activités, ainsi que leurs entrées et sorties, mais n'impose pas la façon de les réaliser (Hunt, 2003). Il peut d'ailleurs s'appliquer au développement de logiciels de tous types et toutes tailles. L'UP requiert la mise à profit d'efforts, c'est-à-dire de travail, dans trois domaines (Alhir, 2002) :

- Les interactions, qui sont les réalisations du processus que sont les cas d'utilisation et l'architecture du système.

- Le contexte, c'est-à-dire piloter l'utilisation qui est faite du framework dans le cadre du projet.
- Les collaborations, qui constituent la concrétisation des interactions dans le contexte défini pour le projet.

Nous formulerons ainsi, autant que possible, nos modèles d'analyse relatifs à notre PAM en nous appuyant sur UML. Précisons aussi que ce que nous avons appelé le contexte de l'UP se compose de quatre phases (Jacobson, 2008b; Rational Software, 2001) :

- Le lancement, pendant lequel il faut délimiter clairement le périmètre du projet. Cette phase doit formaliser la vision générale du projet (besoins, fonctionnalités, contraintes, coûts envisagés, risques, ...) et peut déboucher sur du prototypage. On doit établir durant cette phase si la poursuite du projet en vaut la peine.
- L'élaboration, pendant laquelle les décisions sur l'architecture logicielle vont être prises (Jacobson, 2008a). Au cours de cette phase, les cas d'utilisation sont presque tous identifiés et le planning de développement est établi. On notera que cette phase est la plus critique des quatre et qu'il est vivement conseillé d'avoir prototypé l'architecture candidate pour éviter les mauvaises surprises.
- La construction, pendant laquelle toutes les fonctionnalités du produit sont implémentées et testées. Cette phase doit déboucher sur une version fonctionnelle du produit, couramment appelée version beta, qui est accompagnée d'une documentation utilisateur (Priestley & Utt, 2000).
- La transition, pendant laquelle les retours des utilisateurs permettent la correction de bogues, ainsi que l'ajout et la finalisation des fonctionnalités du produit. Cette phase vise à rendre les utilisateurs autonomes dans leur utilisation du produit en se conformant au possible à la vision du produit réalisée préliminairement. Cette phase prend fin lorsque les utilisateurs finaux sont satisfaits ou lorsque l'on n'a plus assez de ressources pour poursuivre le projet.

Ces phases constituent la partie dynamique du modèle de développement et chacune d'elle est composée d'activités. La spécificité de l'UP réside dans le fait que la gestion même du projet est une de ses activités. Remarquons aussi que ces phases peuvent faire l'objet d'une ou plusieurs itérations. Les itérations sont planifiées, exécutées puis évaluées. Elles doivent permettre de réduire les risques en s'appuyant sur les cas d'utilisation les plus critiques, et chacune d'entre elles se conclut par un livrable sous forme d'une maquette ou d'un prototype (Gabay & Gabay, 2008). Fox (2002) précise que, dans le cas des applications médicales, ces risques peuvent relever de la mise en danger des patients et qu'à ce titre il convient de les identifier pour les écarter durant le développement ou, à défaut, pour pouvoir les monitorer pendant le fonctionnement du programme. Ces itérations ne sont clairement pas indépendantes et ne constituent que des étapes au sein du projet, qu'on assimile souvent à de mini-modèles linéaires similaires à ceux précités. La fin d'une phase coïncide avec la fin de sa dernière itération et la production d'un livrable que l'on appelle « milestone ». Ainsi chacune des phases précédemment énoncées se termine par la livraison d'une « milestone » spécifique, respectivement appelée (Jacobson, 2008b) :

- Les objectifs du cycle de vie
- L'architecture du cycle de vie
- La capacité opérationnelle initiale
- La livraison du produit

D'un point de vue statique, ce modèle de développement intègre des concepts structurants afin de savoir « qui » fait « quoi », « quand » et « comment » (Hunt, 2003). Ces éléments constituent véritablement les briques élémentaires permettant la définition des collaborations du processus telles que définies précédemment. Elles sont au nombre de quatre :

- Les rôles, correspondant à « qui ». Précisons qu'il s'agit des affectations des ressources humaines, basées sur la reconnaissance de certaines compétences, afin de participer à une activité et/ou réaliser des artefacts.
- Les activités, correspondant à « comment ». Il s'agit des unités de travail visant un but clairement défini tel que la création ou la mise à jour d'artefacts par exemple.
- Les artefacts, correspondant à « quoi ». Ce sont les productions concrètes du projet.
- Les processus de travail, correspondant à « quand ». Ces éléments, couramment appelés « workflows » dans le cadre de leur automatisation (Ko, 2009), décrivent sémantiquement les séquences d'activités ainsi que les interactions entre les différentes ressources humaines (nous avons déjà abordé ce concept dans le chapitre **Les axes d'amélioration technique de la prise en charge**). Il est important de signaler que la réalisation de ces processus de travail va permettre l'évolution dynamique du modèle de développement visant à l'accomplissement de ses différentes phases, et qu'à ce titre ils font partie du contexte.

On notera que, pour ne pas trop complexifier inutilement notre démarche, nous faisons l'amalgame entre les activités et les processus de travail dans ce manuscrit. En effet, nous considérerons que dans le cadre de la modélisation générique seule du système, notre démarche n'impliquant ni réellement de collaboration ni d'ordonnancement complexe, toute distinction serait superflue. D'autre part, on peut préciser que nous souhaitons effectuer ces activités séquentiellement dans la mesure du possible.

Le but de cette méthode est d'assurer la production de logiciels de haute qualité répondant aux besoins des utilisateurs finaux tout en respectant une planification et un budget établi (Rational Software, 2001). A ce titre, l'UP prône la conception d'une architecture logicielle constituée de composants réutilisables, ce qui est totalement dans l'optique de généricité recherchée, précédemment évoquée, visant à maximiser l'efficacité de nos efforts. Il ne faudra pas non plus perdre de vue que les différentes phases du modèle devront aussi s'appuyer sur l'activité réelle et quotidienne des utilisateurs finaux, et qu'il faudra trouver un juste équilibre entre impératifs architecturaux et fonctionnels au travers de séries d'itérations (Jacobson, 2008b). En résumé, on retiendra que l'UP est un framework de processus de développement itératif, incrémentiel, dirigé par les cas d'utilisation et centré sur son architecture qui s'appuie sur UML (Alhir, 2002).

En ce qui concerne l'implémentation que nous allons employer, nous avons fait le choix d'une implémentation papier, intitulé UP7, ayant pour mérite d'être assez souple, plutôt concise et se voulant avant tout pragmatique (Gabay & Gabay, 2008). Ceci devrait nous permettre d'avoir la liberté de ne pas avoir à suivre le processus dans son intégralité mais aussi de pouvoir l'adapter à nos besoins spécifiques sans être trop contraints par la procédure. Au passage, on peut noter que cette propriété d'extensibilité du framework de développement qu'est l'UP, est d'ailleurs ce que l'on attend de tout framework (Fayad & Schmidt, 1997). Ainsi, nous nous conformerons uniquement à cette implémentation du modèle de développement dans ses phases de lancement et d'élaboration, et nous nous

cantonnerons à réaliser des activités n’impliquant aucune implémentation logicielle ou activité en découlant. En effet, comme nous l’avons précisé dans notre objectif **MODEL1**, nous souhaitons modéliser l’architecture de référence de la PAM, dont nous aurions besoin pour programmer cette dernière si nous le souhaitions (Jacobson, 2008a). Nous introduirons alors le maximum de généricité dans notre modélisation afin de lui garantir une stabilité maximum à l’usage. La démarche que nous adopterons est constituée des quatre phases classiques d’UP au cours desquelles se dérouleront sept activités. Idéalement elle devrait se dérouler de la manière représentée sur la **Figure 11**. Sur ce schéma des différentes étapes du processus de développement, on pourra préciser que :

- L’axe horizontal représente le temps et traduit l’aspect dynamique du processus lors de son déroulement.
- L’axe vertical représente l’aspect statique du processus, c’est-à-dire la façon dont il est défini en termes d’activités, de ressources humaines, d’artefacts et de processus de travail.
- Les zones grisées, quant à elles, représentent l’effort à consentir pour chaque activité durant les différentes phases d’un projet.

ACTIVITES↓	PHASES →	Lancement	Elaboration	Construction	Transition
1 - Modélisation métier					
2 - Exigences fonctionnelles					
3 - Analyse des cas d’utilisation					
4 - Synthèse de l’analyse					
5 - Conception					
6 - Implémentation					
7 - Tests					

Figure 11 : Schéma d’ensemble de la démarche UP7 (d’après (Gabay & Gabay, 2008))

Etant donné que la PAM est vouée à être uniquement modélisée, nous ne réaliserons que les activités 1 à 5, et seulement pour les deux premières phases du processus de développement. A cette occasion, nous produirons tout de même des exécutables, des prototypes pour éprouver certains aspects de notre modélisation du système (voir les objectifs **PROTO1**, **PROTO2** et **PROTO3**). C'est d'ailleurs la force de l'UP que de pouvoir être adapté à des besoins spécifiques en termes de phases et d'activités du processus de développement (Hunt, 2003).

Il est par ailleurs possible de replacer ces choix, concernant une méthode de développement, dans le contexte du méta-modèle de modélisation que nous avons précédemment évoqué (**Figure 10**). En effet, la méthode que nous avons retenue, UP, est un méta-modèle de développement dont UP7 est une instance. Les relations entre ces modèles, de niveaux d'abstraction différents, et notre processus de développement sont représentés sur la **Figure 12** conformément à notre framework de modélisation.

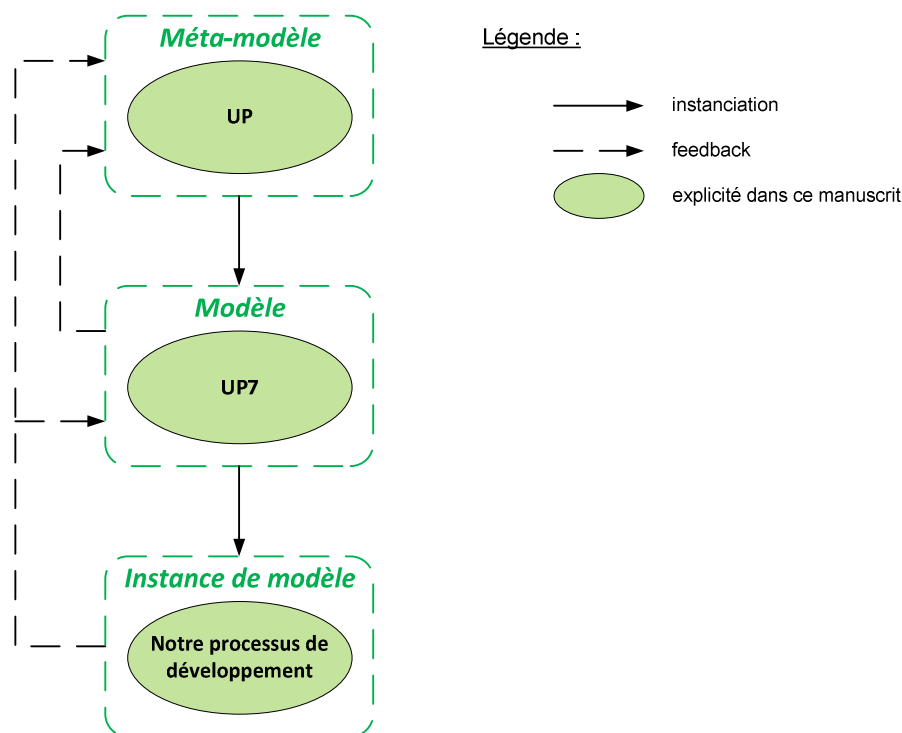


Figure 12 : Représentation de la relation entre UP, UP7 et notre processus de développement

Nous détaillerons les résultats obtenus par l'application de l'UP7 plus loin dans ce document. Nous n'effectuerons pas les activités d'implémentation et de test, mais nous nous restreindrons à en évoquer certains aspects. Nous présentons dans la suite les modèles indispensables à la description du fonctionnement de notre système, c'est-à-dire les modèles d'analyse que nous utiliserons lors de la mise en œuvre de cette démarche.

II) Le choix des modèles d'analyse

Cette analyse correspond à l'étape de modélisation préalable à toute implémentation, à tout codage (Royce, 1970). Elle doit nous permettre de décrire l'architecture logicielle de notre solution, d'en réaliser les spécifications fonctionnelles et techniques (Noël, 2012). L'architecture logicielle d'un système informatique est la ou les structures du système constituées de composants logiciels, des propriétés de ces composants visibles de l'extérieur et des relations qui existent entre ces composants (Bass, Clements, & Kazman, 2003). Comme nous l'avons évoqué lors de la partie dédiée aux modèles de développement, nous allons principalement utiliser le langage orienté objet UML (Object Management Group, 2014e) pour décrire le fonctionnement attendu de notre PAM. Ce langage est évolutif, largement applicable et s'appuie sur des outils et standards de l'industrie (Ramsin & Paige, 2008). A ce titre, en qualité de standard, il intègre beaucoup d'éléments provenant d'autres méthodes de modélisation développées durant les années 90 (pour en savoir plus sur l'historique de ce langage, consulter (Hunt, 2003)). Il peut être vu comme un ensemble de techniques de modélisation pour spécifier, visualiser, construire et documenter les éléments constitutifs des processus d'un système (Alhir, 2002). Ainsi la démarche de développement UP7 (Gabay & Gabay, 2008) que nous adoptons va nous conduire à la réalisation de vues et diagrammes UML accompagnés d'indispensables descriptions (Hunt, 2003). Ces éléments seront autant de modèles d'analyse permettant la représentation de tout ou partie du logiciel selon des points de vue différents. Comme nous l'avons évoqué, UML est composé de différents diagrammes permettant la représentation :

- Des cas d'utilisation
- Des classes et d'objets
- Des activités
- Des séquences
- Des collaborations
- Des états
- Des composants
- Du déploiement

Nous allons grâce à cela être en capacité de représenter l'architecture du système selon deux dimensions, l'une structurelle et l'autre fonctionnelle. La dimension structurelle, appelée aussi statique, décrit quels éléments constituent le système et quelles relations ils entretiennent (Bass et al., 2003). Avec la démarche UP, cette dimension est matérialisée par l'architecture du système. La dimension fonctionnelle, aussi appelée dynamique, dépeint le comportement de ces éléments à travers leur façon de collaborer et d'interagir pour satisfaire aux fonctionnalités du système (Alhir, 2002). Il s'agit pour nous des cas d'utilisation, qui sont autant de parties des fonctionnalités de la PAM en devenir. Mais tous les aspects du futur système ne peuvent pas être décrit à l'aide d'UML, et il conviendra d'étendre et de compléter ce langage à l'aide d'autres types de modèles (Hunt, 2003). Par exemple, nous pourrions utiliser des diagrammes de contexte afin de décrire les pourtours du système. De tels diagrammes ne sont pas réellement formalisés et n'appartiennent donc pas à la norme UML 2 (Kossiakoff et al., 2011). De la même manière, il peut aussi être utile d'utiliser des schémas de principe pour représenter graphiquement une idée, et ainsi en

faciliter la compréhension, sans pour autant respecter un formalisme qui pourrait nuire à sa lisibilité.

Notre PAM se voulant être clairement utilisateur-centré, les exigences fonctionnelles de cette dernière seront prépondérantes pour sa modélisation. En effet, notre système est destiné à différents types d'utilisateurs :

- Les patients, qui sont les individus dont le système doit améliorer la prise en charge.
- Les praticiens, qui sont des professionnels de santé, appartenant aux milieux médical et paramédical, destinés à intégrer leurs pratiques à l'aide du système.
- Les chercheurs, qui doivent pouvoir réaliser simplement leurs expérimentations à l'aide du système.

Comme énoncé ci-dessus chacun de ces types d'utilisateurs a ses propres attentes vis-à-vis du système. Les besoins qui en découlent vont devoir être pris en compte comme contraintes de développement. En particulier, le premier besoin inhérent à l'utilisation d'un système, et commun à tous les types d'utilisateurs que nous venons d'énoncer, est de pouvoir interagir avec lui. Cette nécessité d'interaction avec le système, de pouvoir communiquer avec lui, est une condition sine qua non à son usage. D'ailleurs plus un système permet une interaction proche de la communication entre êtres humains, plus il est facile d'accès, c'est-à-dire plus il est aisé d'interagir avec lui (Bernsen, 1994; Bouchet, 2006; Obrenović & Starčević, 2004). On parle d'ailleurs de langage naturel dans le cas d'une communication entre humains, et on peut ajouter que cette communication est distribuée et multimodale (Weibel et al., 2013; Weiss, Scheffler, Möller, & Reithinger, 2012). Notre PAM voulant intégrer, au travers d'un SI, des interactions entre êtres humains dans le cadre de leurs pratiques, il paraît pertinent de vouloir assurer informatiquement la médiation et l'assistance entre ces différents acteurs en supportant les interactions multimodales. Une analyse des modalités d'interaction entre les utilisateurs et le système est donc primordiale. On précisera que les modalités d'interaction composent le module d'interface du système (Bouchet, 2006). A ce titre, conformément à notre choix de modèle de développement, les modèles d'analyse que nous allons utiliser doivent être en mesure de décrire en détail :

- Les interactions des différents types d'utilisateur avec le système.
- L'architecture de la partie logicielle de la solution.

L'interface utilisateur peut être vue comme la « peau » du système, alors que l'architecture représente le « squelette » et les fonctionnalités « les muscles » (Hunt, 2003).

L'étude des interfaces utilisateur relève du domaine des « Interface Homme-Machine » (pour des ouvrages très complets sur le sujet, consulter (Sears & Jacko, 2007; Shneiderman & Plaisant, 2010)). A ce titre, notre recherche va se concentrer sur leur utilisabilité et les fonctionnalités qu'elles offrent. On pourra aussi préciser que ces interfaces peuvent être de natures très diverses. En particulier, il en existe des interfaces logicielles – virtuelles – ainsi que des interfaces matérielles – tangibles. Il convient donc préliminairement de passer en revue les différents types d'IHM afin de déterminer lesquelles présentent le plus d'intérêt dans le cadre du développement de notre PAM. En effet, les interactions homme-machine sont essentielles au succès d'une telle plate-forme à visée médicale (Sainfort et al., 2007). Naturellement, nos analyses ultérieures s'attacheront à approfondir notre compréhension des types d'interfaces ainsi retenus.

Pour pouvoir aborder la conception de l'architecture d'un système il faut s'intéresser aux design patterns. En effet, chaque pattern a pour but de décrire un problème récurrent et y apporter un schéma de solution afin d'en permettre la résolution systématique (Buschmann, Meunier, Rohnert, Sommerlad, & Stal, 1996). Ces patterns saisissent ainsi la faculté d'une modélisation à être réutilisable, et ils capitalisent l'expérience des solutions apportées par d'autres chercheurs et développeurs à des problèmes fréquents. On peut même ajouter que ces patterns peuvent former un « langage » si l'on considère qu'ils forment un réseau de patterns interconnectés, de la même manière que les problèmes qu'ils traitent sont interconnectés au travers d'une décomposition cartésienne. L'intérêt, en termes d'augmentation du niveau de qualité, à y avoir recours a d'ailleurs été montré (Ng, Cheung, Chan, & Yu, 2006). On pourra également noter que la **Programmation Orientée Objet** a fait l'objet d'un recensement et une classification des différents patterns s'y appliquant (Gamma et al., 1994). D'autre part, on signalera que, logiquement, les modèles de développement tirent aussi parti de patterns entrant dans la conception des méta-modèles qui permettent de les caractériser (Ambler, 1998; Ramsin & Paige, 2008). Dans ce dernier cas, on parlera alors de « process pattern ».

Comme nous l'avons avancé, la modélisation de notre PAM va principalement s'articuler autour de la description de deux constituants du dispositif : les interfaces utilisateur et l'architecture du système. Dans les sections suivantes, nous allons considérer les concepts attenants à chacun de ces domaines et déterminer quels modèles d'analyse nous allons pouvoir utiliser. Pour cela, nous nous appuyerons sur de nombreux patterns dont nous mettrons les relations en évidence.

A) Les interfaces utilisateur

Avec l'évolution de l'outil informatique, les interfaces utilisateur sont devenues un point critique pour de nombreuses applications. Dans le cadre d'un processus de développement logiciel, il devient ainsi indispensable de mettre à profit des connaissances et outils dédiés aux IHM (Obrenović & Starčević, 2006). En particulier, les interfaces utilisateur sont directement impliquées dans l'accomplissement des cas d'utilisation, et, à ce titre, leur étude est essentielle dans le cadre de notre implémentation de l'UP. On remarquera également que des IHM adaptées et adaptables sont un moyen efficace de faire profiter les utilisateurs, présentant divers types de handicaps, des avantages offerts par l'informatique (Shneiderman & Plaisant, 2010). Dans cette section, nous proposons d'aborder brièvement l'étude des IHM selon différents points de vue :

- L'évolution des interfaces utilisateur (pour un panorama plus complet des publications dans ce domaine consulter (Grudin, 2012; Jørgensen, 2008))
- Les différents types d'interfaces utilisateur modernes
- L'utilisabilité et les fonctionnalités

Ces interfaces jouent un rôle essentiel dans la communication entre le système et les utilisateurs, et influent logiquement sur l'expérience que ces derniers vont en avoir (Wangler & Backlund, 2005). De la sorte, nous allons être amenés à étudier cette communication via des langages d'interactions. Nous voulons ainsi pouvoir identifier des modèles d'analyse propices à la modélisation d'IHM dans le cadre de notre développement d'une PAM.

Pour des raisons de clarté, nous allons définir les différentes notions manipulées dans le cadre des langages d'interactions (**Tableau 3**).

		Domaines des langages d'interactions	Domaines de l'étude des IHM	
conditionne ↗ ↖		Domaine sémantique	Modalités	Niveau conceptuel
		Domaine pragmatique	Modes	
conditionne ↗ ↖		Domaine syntaxique	Média/Canaux	Niveau physique

Tableau 3 : Relations entre les concepts liés aux langages d'interactions

Les média, constitués de canaux, désignent les moyens de diffusion des informations. Les modes s'appliquent à préciser la manière selon laquelle se fait l'interaction. Finalement, les modalités se placent au plus haut niveau d'abstraction des langages d'interaction et cherchent à expliquer le sens porté par les informations échangées pendant l'interaction.

1) L'évolution des interfaces utilisateur

Les premiers ordinateurs sont apparus à la fin de la seconde guerre mondiale et ils étaient manipulés à l'aide de commutateurs et de câbles. Ils n'étaient donc pas particulièrement faciles à utiliser et requéraient l'implication d'un grand nombre d'opérateurs humains. Avec l'invention du transistor pour remplacer le tube à vide, les ordinateurs devinrent plus accessibles, autant financièrement qu'au niveau de leur utilisabilité, et il était dès lors possible d'interagir avec eux à l'aide de cartes perforées et/ou de bandes magnétiques (Grudin, 2012). Cette génération de machines a d'ailleurs commencé à disposer de fonctionnalités d'affichage. Ils ne permettaient cependant pas encore une interaction en temps réel et étaient voués à des activités de calculs.

Dans les années 60 sont apparus les premiers systèmes aptes à partager leur temps d'exécution entre plusieurs utilisateurs. Ces derniers exploitaient ce type de machines centralisées au travers de terminaux. De telles machines disposent alors de systèmes de mise en réseau permettant de les interconnecter. C'est aussi à cette période qu'Ivan Sutherland développa le système Sketchpad (**Figure 13**) démontrant ainsi entre autres l'intérêt d'une représentation iconographique et de l'utilisation du crayon optique (Grudin, 2012; Jørgensen, 2008). La décennie suivante vit se généraliser les interactions en les lignes de commandes.

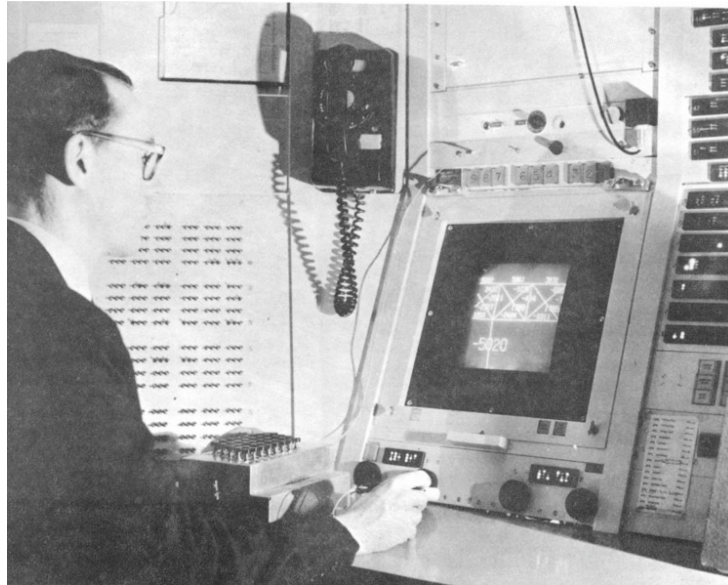


Figure 13 : Ivan Sutherland utilisant Sketchpad (source (Sutherland, 2003))

Popularisées à la fin des années 70, les interfaces graphiques, qui sont aussi appelées « **Graphical User Interface** », se sont imposées comme la norme depuis les années 80 (Turk & Kölsch, 2003). L'utilisation de ces interfaces implique que l'utilisateur puisse interagir avec le système via un clavier et une souris, inventée en 1964 par Doug Engelbart (Beaudouin-Lafon, 2001), afin de manipuler des représentations pictographiques affichées sur un moniteur (Myers, 1998). D'autre part, les systèmes centralisés commencent à être largement supplantés par des stations personnelles dont le prix permet maintenant l'accès au grand public. Ces nouvelles machines servent de plus en plus à manipuler de l'information et de moins en moins à effectuer des calculs.

La fin des années 90 a vu se démocratiser le travail coopératif assisté par ordinateur au travers de l'utilisation massive d'Internet (Grudin, 2012). Il est ainsi apparu de nombreux systèmes ubiquitaires (Weiser, 1993) permettant l'accès et l'utilisation d'un même système grâce à différents matériels et applications (Kaschek, Matthews, Schewe, & Wallace, 2006; Seffah & Forbrig, 2002). L'apparition de tels matériels relativement hétérogènes (nombre et taille d'écrans, types de périphériques d'entrées disponibles, ...) a remis en question l'hégémonie des GUI (Myers, Hudson, & Pausch, 2000). A ce titre, de tels systèmes supportent des interactions différentes de celles classiquement offertes par le clavier et la souris, procurent de nouveaux moyens de communication et disposent de présentations adaptées aux machines qui en permettent l'utilisation (comme par exemple les GUI adaptées à l'usage tactile ou encore les affichages en 3D). L'usage de ces dispositifs est par ailleurs en partie dévolu aux domaines des loisirs interactifs et aux télécommunications.

2) Les différents types d'interfaces utilisateur modernes

De nos jours, il est ainsi possible de réaliser une même activité de différentes manières. En effet, un grand nombre de systèmes sont maintenant portables, connectés et permettent une interaction de plus en plus naturelle. A ce titre, différentes approches d'IHM cohabitent et nous proposons de les caractériser un peu plus en détail. Mais avant cela, il paraît

primordial de présenter ce que ces différents types d'IHM partagent. Effectivement, ils sont tous utilisés dans des systèmes interactifs et ainsi relèvent tous de la gestion de modalités d'interaction (Nigay, 1994). On pourra distinguer deux types de telles modalités (**Figure 14**) :

- Les modalités d'entrées, qui sont les moyens d'action de l'utilisateur pour interagir avec le système informatique (Bouchet, 2006; Kamel, 2006). Ces modalités s'appuient sur différents types de « représentations externes ». Elles sont désignées de la sorte car elles sont externes à l'environnement cognitif de l'utilisateur. Il s'agit des éléments de l'interface virtuelle/logicielle et/ou physique/tangible.
- Les modalités de sorties, c'est-à-dire les moyens de perception de l'état du système informatique par l'utilisateur (Bernsen, 1994; Mansoux, Nigay, & Troccaz, 2006; Vernier & Nigay, 2000). Ces modalités visent à provoquer chez l'utilisateur des effets appelés « représentations internes » des modalités. Ces effets peuvent être de plusieurs sortes :
 - Les effets sensoriels
 - Les effets perceptuels
 - Les effets moteurs
 - Les effets cognitifs

L'interface, au travers de modalités d'IHM, permet à l'utilisateur d'utiliser le noyau fonctionnel d'un système informatique (Coutaz & Balbo, 1991; Nigay, 1994). Ce noyau fonctionnel est la partie d'une application qui permet la modélisation et l'exécution des concepts de la tâche informatisée. Son utilisation par l'utilisateur peut être relativement directe ou se faire au travers de middlewares qui agiront comme des médiateurs sur les modalités. Les middlewares offrent une couche d'abstraction supplémentaire, relevant du domaine de l'intégration horizontale, facilitant les interactions entre multiples composants.

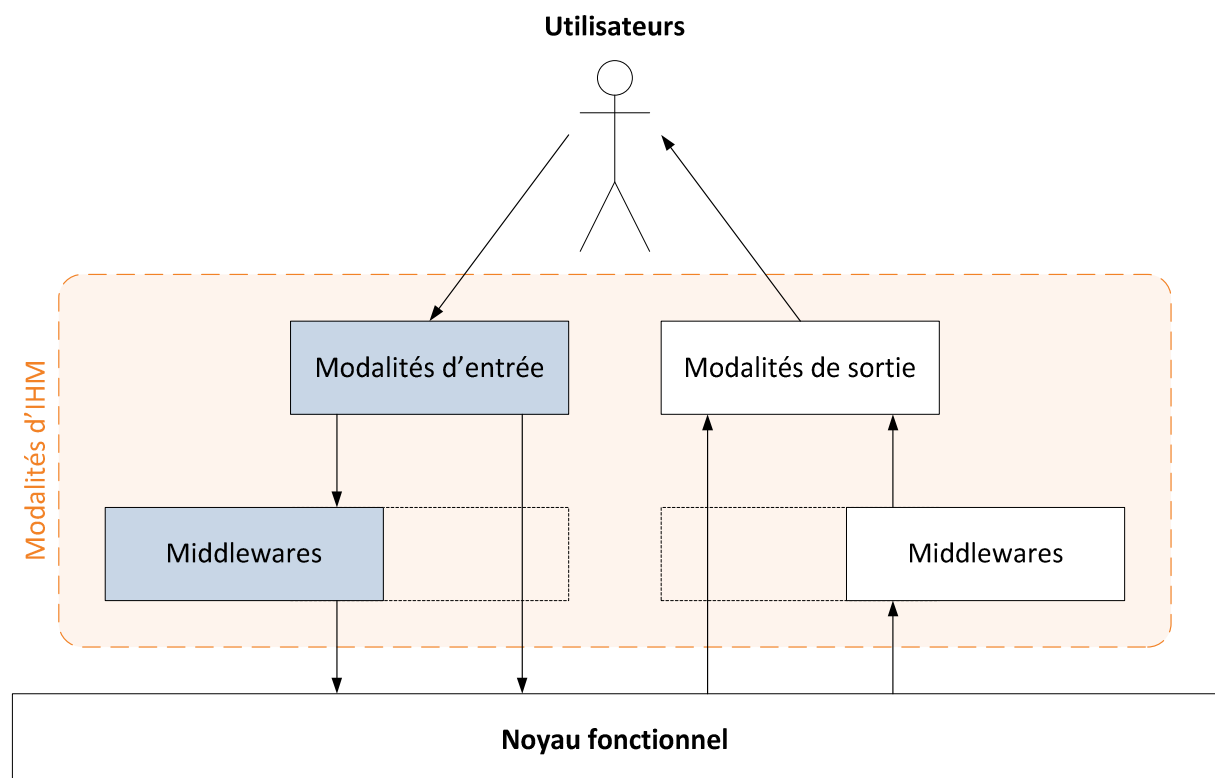


Figure 14 : Schéma général des interactions modales en IHM

a) Les interfaces graphiques

Il s'agit des dispositifs de dialogue homme-machine, que nous avons précédemment introduits, dans lesquels les objets sont représentés à l'écran sous forme de pictogrammes. L'utilisateur peut alors manipuler ces objets en imitant une manipulation physique à l'aide d'un dispositif de pointage tel que la souris. On parle dans ce cas de « manipulation directe » (Shneiderman & Plaisant, 2010). Ces interfaces prônent le style d'interaction WIMP – Windows, Icon, Menu, Pointing devices – et utilisent classiquement la métaphore du bureau.

b) Les interfaces perceptuelles

Depuis la fin des années 90, il est communément admis que les interactions homme-machine, qui dépendent uniquement du clavier, de la souris et d'un moniteur, ne peuvent satisfaire les besoins des futurs types d'interactions homme-machine (Turk, 1998). Un mode d'interaction basé sur le passage de commandes dans l'attente de réponses du système n'est pas un mode naturel de dialogue. On remarquera aussi que le type d'interfaçage existant jusqu'alors nécessite un apprentissage particulier (Zhu & Wang, 2006), ce qui force l'utilisateur à s'adapter au système alors que, dans le cas d'un système utilisateur-centré, ce devrait être au système de s'adapter à son utilisateur. Les interfaces utilisateur perceptuelles cherchent à tirer parti des capacités perceptives des utilisateurs afin de présenter de l'information et son contexte de façon significative et naturelle. Ainsi, le développement de telles interfaces passe par la compréhension de la vision humaine, la perception auditive, les conventions de conversation, les capacités tactiles, etc. (Turk & Kölsch, 2003; Turk, 1998). Il est intéressant de constater que tels systèmes peuvent être appliqués à l'identification des émotions de leurs utilisateurs (Clay, Couture, & Nigay, 2009; Mäkinen, 2007). Effectivement, ceci traduit une forte volonté de produire des systèmes offrant des moyens d'interaction de plus en plus proches de ceux affichés lors d'interactions naturelles entre êtres humains.

De plus, si nous voulons pouvoir évaluer le comportement naturel de notre utilisateur, pouvant souffrir de troubles spécifiques du langage, il paraît pertinent que l'interface permettant cela mette à contribution principalement ses compétences en langage et communication. Ce dernier point nous conforte dans l'idée qu'il est primordial de procéder à notre recueil de données à travers une interface utilisateur à la fois naturelle et transparente : de telles interfaces sont couramment appelées interfaces perceptuelles (Turk & Kölsch, 2003). On remarquera d'ailleurs qu'il existe de nombreux outils adaptés, avec plus ou moins de réussite, à la conception de telles interfaces (Myers et al., 2000).

c) Les interfaces multimodales

Le premier système multimodal intégrant des modalités perceptuelles fut présenté en 1980 par Rich Bolt (Beaudouin-Lafon, 2001). De tels systèmes exploitent au moins deux modes combinés d'entrée utilisateur, pour lesquels une analyse sémantique est effectuée (Nigay & Coutaz, 1993), qui sont coordonnés avec des sorties multimédia (Oviatt, 2007). Les systèmes multimodaux sont radicalement différents de ceux reposant sur des interfaces graphiques

classiques, principalement en raison de la nature de la communication humaine (Turk, 2014). Tandis que les entrées des GUI sont atomiques (c'est-à-dire indivisibles), discrètes et certaines, la perception et la reconnaissance par la machine de langages naturels, tels que la parole ou le geste, est incertaine (Oviatt & Cohen, 2000). Ainsi, une approche multimodale peut pallier les lacunes des systèmes classiques en recueillant des informations complémentaires via différents canaux d'information (Tessendorf et al., 2010). Effectivement, la multimodalité va prendre tout son sens dès lors que nous allons procéder à une fusion des données acquises et/ou analysées (voir la sous-section **La fusion de données**). On signalera qu'elle facilite également l'identification d'émotions humaines (Picard, 2000)(pour un exemple d'utilisation, consulter (Gonzalez-Sanchez, Chavez-Echeagaray, Atkinson, & Bursleson, 2011)). De telles capacités d'interactions tendent à rendre les systèmes informatiques qui les utilisent plus « intelligents ».

Indépendamment des modalités d'interaction impliquées, les interfaces perceptuelles et multimodales peuvent être (Oviatt & Cohen, 2000; Oviatt, 2007) :

- Passives, si le système les utilise de manière proactive en récupérant continuellement des données, pouvant d'ailleurs provenir plus ou moins involontairement de l'utilisateur.
- Actives, si le système se contente de réagir au travers de ces modalités à des entrées intentionnelles de l'utilisateur.
- Hybrides, quand elles combinent ces deux aspects.

Il est à signaler que les interfaces multimodales peuvent minimiser la charge cognitive des utilisateurs. Ceci induit une amélioration de leurs performances lors de la réalisation de tâches, aussi bien en perception qu'en production (Oviatt, 2007). Ces interfaces présentent donc des avantages indéniables autant pour assister les praticiens dans l'accomplissement de multiples processus de soins (Aigner & Miksch, 2006) que pour assurer la remédiation de patients souffrant de troubles neuropsychologiques. Elles peuvent également être utilisées avec succès pour les personnes souffrant de handicaps (par exemple, consulter (Tessendorf et al., 2010)). Ainsi, dans la suite de nos développements, nous supposerons que ce type d'IHM est celui qui est utilisé.

3) L'utilisabilité et les fonctionnalités

L'utilisabilité étudie la manière qu'a un système d'interagir avec ses utilisateurs dans le but d'en optimiser la productivité et le confort d'utilisation (Juristo, Windl, & Constantine, 2001; Kamel, 2006). Pour cela, il faut analyser, à l'aide de métriques, l'interface ou le comportement d'utilisateurs potentiels (ou simulés (De Loo, Le Bodic, Calvet, & Tisseau, 2006)) pendant qu'ils effectuent des tâches supportées par le système. De cette manière, il devient possible de concevoir des systèmes en adéquation avec les capacités de leurs probables utilisateurs. Pour atteindre cet objectif, il est également possible d'utiliser des « usability patterns » afin d'intégrer directement des recommandations de bonnes pratiques (Seffah & Forbrig, 2002; van Welie & van der Veer, 2003). Ferré et al. (2001) attribuent à cette discipline cinq attributs (qui sont aussi repris dans (Shneiderman & Plaisant, 2010)) :

- La facilité d'apprentissage, c'est-à-dire le temps qu'il faut à un utilisateur pour réaliser, avec le système, des tâches avec une rapidité similaire à celle d'un expert.

- L'efficacité, qui renvoie à la vitesse maximum à laquelle l'utilisateur peut accomplir des tâches à l'aide du système.
- La rétention de l'utilisateur au fil du temps, qui représente la capacité de l'utilisateur à conserver des acquis concernant l'utilisation du système après une période où il ne l'aurait pas utilisé.
- Le taux d'erreur, c'est-à-dire le taux d'erreur que l'utilisateur commet lorsqu'il utilise le système.
- La satisfaction, qui reflète l'impression générale de l'utilisateur vis-à-vis du système.

L'analyse de l'utilisabilité vise à fournir, à l'aide d'évaluations récurrentes lors du processus de développement, un dispositif qui soit bien adapté à l'utilisateur (pour un exemple de technique d'évaluation de l'utilisabilité, basée sur le suivi oculaire, consulter (Jacob & Karn, 2003)). En effet, ce n'est pas à ce dernier d'avoir à s'adapter au système développé à son intention. Dans le contexte médicale, on peut, par exemple, signaler l'existence d'expériences visant à éprouver l'intérêt d'une symbolique visuelle pour représenter des informations appartenant à des DMP (Müller, Maurer, Reihls, Sauer, & Zatloukal, 2011; Müller, Sauer, Zatloukal, & Bauernhofer, 2010). Ces dernières illustrent le fait que la conception d'un système devrait être centrée sur l'utilisateur (Anderson, Fleek, Garrity, & Drake, 2001). Afin de pouvoir commencer la conception d'une IHM intégrant ceci, il faut pouvoir apporter des réponses aux questions suivantes (Ferré et al., 2001) :

- Quels sont les utilisateurs du système ?
- Qu'ont-ils besoin d'effectuer ?
- Que doit leur fournir le système pour y arriver ?
- Comment le système devrait leur fournir cela ?

Répondre à ces questions doit permettre d'identifier les fonctionnalités, ainsi que les moyens mis en œuvre pour les assurer, au travers de l'établissement des besoins, c'est-à-dire d'une analyse fonctionnelle (Anderson et al., 2001). Dans la méthode de développement que nous avons choisie (voir le chapitre **Le choix d'un modèle de développement**), l'identification correspond à l'établissement des cas d'utilisation. Le langage UML peut d'ailleurs bénéficier d'extensions afin de mieux retranscrire et intégrer les considérations liées à l'utilisabilité. Ainsi, lors de l'analyse fonctionnelle, Obrenović et Starčević (2006) insistent sur le fait qu'à l'aide de leur extension d'UML on peut expliciter :

- La description détaillée des utilisateurs, en précisant bien les propriétés qui les caractérisent en tant qu'acteurs du système. On mettra ainsi en évidence les différents groupes d'utilisateurs à même d'utiliser le système.
- La description détaillée de l'environnement d'utilisation, en spécifiant les caractéristiques et spécificités de l'endroit où le système est utilisé. On mettra en exergue de la sorte les différents contextes d'utilisation du système.

De tels modèles d'analyse, matérialisés par des diagrammes UML, permettent de répondre à la première question soulevée par Ferré et al. (2001). Il faut d'ailleurs être conscient qu'il peut y avoir des différences entre les différents types d'utilisateurs d'un système dont il faut tenir compte lors de la conception des interfaces utilisateur. Ceci se traduit par le développement d'interfaces adaptées aux usages de ces derniers. En particulier, il est capital de filtrer les données qui sont présentées à l'utilisateur au regard de ses compétences et attributions (Martins-Baltar et al., 2013). Une telle restriction d'accès aux données est d'une part nécessaire pour ne pas perturber l'utilisateur avec des informations ne le concernant

pas, et d'autre part indispensable pour des raisons de confidentialité. Dans notre cas, il y a par exemple un écart évident, en termes de connaissances médicales, entre les patients et les praticiens (Sainfort et al., 2007). Ceci est également vrai pour des praticiens appartenant à des disciplines différentes. Pour répondre aux deuxième et troisième questions de Ferré et al. (2001), le langage UML propose des diagrammes de cas d'utilisation qui servent à établir clairement les fonctionnalités que le système doit exhiber. Afin d'apporter des réponses à la dernière question, Obrenović et Starčević (2006) ajoutent, au cours de l'analyse du système, des classes adjointes d'attributs, décrivant :

- Les interactions, c'est-à-dire les effets que l'on souhaite provoquer chez l'utilisateur. Ils peuvent de cinq types : sensoriels, perceptuels, affectifs, cognitifs ou linguistiques.
- Les éléments de l'interface, c'est-à-dire les composants interactifs utilisés pour réaliser les IHM. Chacun de ces composants supporte d'ailleurs des entrées et des sorties.

Dans le cadre du processus de développement UP7, le dernier type d'activités que nous traiterons, qui relève de la conception, peut aussi être adapté pour tenir compte des considérations liées aux interfaces utilisateur. Il est effectivement possible, pour décrire des interfaces multimodales, d'utiliser des modèles UML et non uniquement des outils de conception et d'analyse spécialisés (Bouchet, 2006; Obrenović & Starčević, 2004, 2006). Ainsi, nous pourrions définir nos propres modèles à l'aide d'un méta-modèle destiné à la représentation des modalités d'IHM (Figure 15). D'autre part, nous utiliserons certains modèles, tels que les schémas de contexte ou de principe, ne se conformant à aucun formalisme particulier ainsi que des maquettes d'interfaces. En effet, ce type de modélisation, bien que peu formel, paraît assez simple pour ne pas être ambigu, et vient compléter efficacement les diagrammes UML.

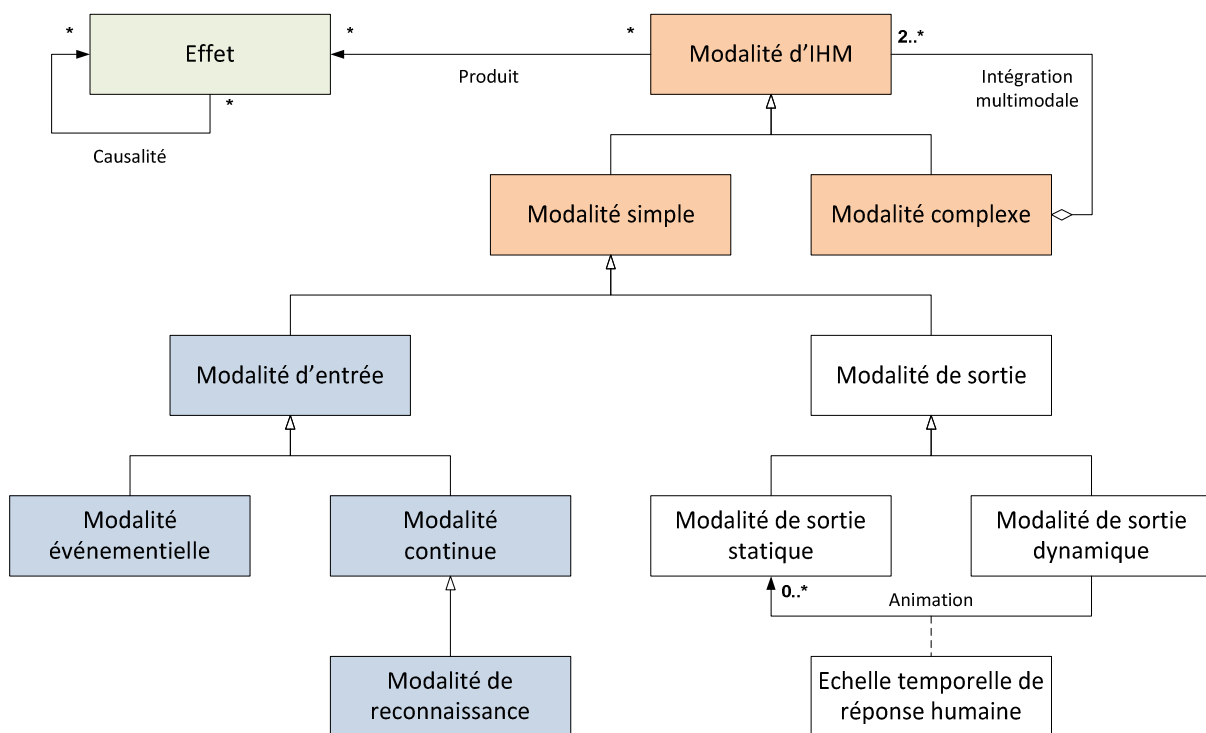


Figure 15 : Diagramme UML du méta-modèle pour représenter des modalités d'IHM (d'après (Obrenović & Starčević, 2004))

Pour décrire les informations destinées à être échangées entre l'utilisateur et le système au cours de la réalisation de tâches, l'identification des modalités d'entrées et de sorties impliquées dans ces échanges est nécessaire (Bernsen, 1994). La **Figure 15**, qui utilise un code couleur similaire à la **Figure 14**, présente un méta-modèle d'analyse de ces modalités.

Pour finir, on pourra noter que l'utilisabilité ne s'applique pas uniquement à l'interaction avec un logiciel mais également à celle avec ses fonctionnalités d'aide, sa documentation utilisateur ou encore à ses instructions d'installation (Ferré et al., 2001).

B) L'architecture du système

Nous allons maintenant entreprendre la description de l'architecture de notre PAM. C'est un point important de l'UP étant donné que cette méthodologie est centrée autour de l'architecture. Pour cela, nous allons présenter les design patterns et paradigmes de programmation que nous utiliserons. Nous comptons ainsi procéder incrémentalement à des choix architecturaux en accord avec les besoins que nous avons déjà identifiés (voir le chapitre **Les axes d'amélioration technique de la prise en charge**). On fera remarquer que l'application de design patterns est primordiale afin de pouvoir anticiper de futures modifications du système (Ng et al., 2006). Notre étude va, à ce titre, nous permettre de détailler plusieurs des exigences techniques que notre dispositif final devrait satisfaire. Nous aborderons d'ailleurs des exigences relatives à la macro-architecture du système aussi bien qu'à sa microarchitecture (Hollingsworth & Weide, 1994).

1) Le design pattern Modèle-Vue-Contrôleur

Parmi les design patterns conditionnant les choix architecturaux, un en particulier s'attache à résoudre le problème de l'accès par l'utilisateur aux données selon de multiples points de vue (Bragge, 2013) : il s'agit du design pattern **Modèle-Vue-Contrôleur** (Krasner & Pope, 1988). Étant donné que notre PAM est destinée à plusieurs types d'utilisateurs différents, les données qu'il manipule vont devoir être présentées selon le point de vue de chacun. En conséquence, une modélisation en accord avec les concepts du MVC semble particulièrement appropriée. On notera que ce design pattern décrit à haut niveau l'architecture du système et donc qu'il vise à formater la macro-architecture de ce dernier. Remarquons également qu'on peut retrouver un découpage architectural suivant une logique similaire dans les modèles Arch (Bass et al., 1992).

Le modèle MVC est un design pattern qui permet d'isoler la couche de présentation de l'application logique c'est-à-dire du noyau fonctionnel. Une telle séparation a pour but de faciliter les modifications/évolutions de l'IHM (Nigay, 1994) et donc de minimiser l'impact d'améliorations liées à l'utilisabilité sur le reste de l'application. Il est ainsi fréquemment utilisé dans un contexte de développement de GUI, mais il reste un paradigme général de programmation (Bragge, 2013). Cette modélisation préconise de décomposer l'application logicielle en trois parties qui communiquent les unes avec les autres et qui forment une triade. Chacune d'entre elles, composée d'objets qui lui sont propres, a ainsi un domaine limité de fonctionnalités à gérer (Burbeck, 1992) :

- Le Modèle, qui régit les données et leur comportement. En particulier, il peut servir à réaliser les traitements métier (Potel, 1996).
- La Vue, qui assure les sorties graphiques représentatives de l'état du Modèle. C'est par ce biais que l'interaction avec l'utilisateur est supportée.
- Le Contrôleur, qui analyse les entrées de l'utilisateur afin de réaliser des traitements sur le Modèle. Cette couche fait véritablement la jonction entre les deux autres.

Nous pouvons d'ailleurs apporter quelques précisions à ces définitions. Les objets du Modèle peuvent être complètement passifs s'ils se contentent d'être manipulés par ceux du Contrôleur et de la Vue : ils peuvent ainsi n'avoir aucune conscience des autres parties de la triade. Cependant, il leur est aussi possible d'être actifs s'ils ont en charge de notifier des objets de la Vue des modifications qu'ils pourraient avoir subies. Il est aussi important de signaler que les éléments du Modèle peuvent être partagés par plusieurs triades (Burbeck, 1992; Krasner & Pope, 1988), c'est-à-dire qu'ils peuvent être associés avec différents objets de la Vue et du Contrôleur. Ceci ouvre la voie au développement d'applications collaboratives disposant de Modèles partagés (Qiu, 2005). Qu'ils soient passifs ou actifs, les éléments appartenant au Modèle exhibent des interfaces permettant l'accès aux données (Buschmann et al., 1996). A contrario, chaque objet de la Vue est associé avec un seul et unique objet du Contrôleur, et la réciproque est vraie. Les éléments de la Vue ont la faculté d'être encapsulés les uns dans les autres selon un système hiérarchique auquel leurs homologues, appartenant au Contrôleur, se conforment également. Pour finir, nous précisons que l'objet du Contrôleur qui se situe au sommet de cette hiérarchie a la responsabilité de gérer l'ensemble des autres objets du Contrôleur. En effet, il doit déterminer, à chaque instant, lequel de ces objets a la main sur les entrées de l'utilisateur (Burbeck, 1992). La **Figure 16** représente certaines de ces considérations ainsi que l'ensemble des interactions en présence dans ce paradigme.

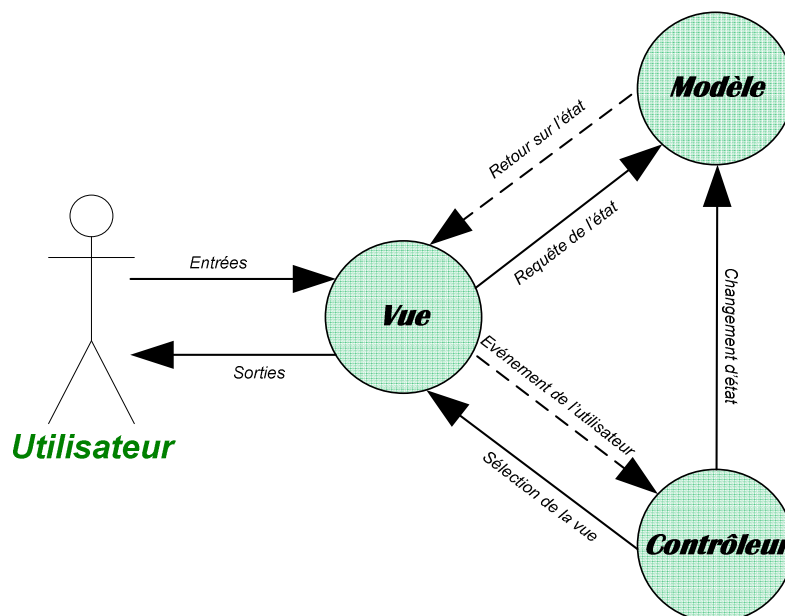


Figure 16 : Schéma des interactions au sein du modèle MVC

La couche de présentation de l'application et la seule partie du logiciel avec laquelle l'utilisateur va pouvoir directement interagir. L'étude d'un tel pattern confirme donc qu'il

peut s'avérer précieux pour la conception de GUI. D'autre part, la définition de nombreuses interfaces entre les couches favorise la réutilisabilité des composants développés ainsi que leur portabilité (Krasner & Pope, 1988; Potel, 1996). On rappellera la nécessité pour notre PAM de pouvoir être distribuée et de supporter l'interopérabilité des composants logiciels et du matériel qu'elle intègre (voir le chapitre **Les axes d'amélioration technique de la prise en charge**). De précédents développements (Qiu, Pallickara, & Uyar, 2004; Qiu, 2004) ont montré que le paradigme MVC pouvait satisfaire à ces exigences par la mise en œuvre d'une communication explicite entre ses composants via des messages asynchrones (par l'intermédiaire d'un broker par exemple). Cet aspect lié à la communication entre les composants est un élément conditionnant fortement les performances globales du système (Sauer & Engels, 1999).

Ce pattern permet finalement de déterminer les caractéristiques techniques de l'architecture, c'est-à-dire des caractéristiques n'étant pas fonctionnelles. En contraignant nos modèles d'analyse à se conformer au pattern MVC, nous comptons répondre aux questions que pose le modèle de développement mais auxquelles l'analyse métier ne peut pas répondre. On remarquera que le design pattern MVP (Potel, 1996), directement dérivé de MVC, est bien adapté aux applications de type client-serveur ainsi qu'à la POO. On fera d'ailleurs remarquer que la **Figure 16** intègre les améliorations apportées par MVP. Ceci semble correspondre favorablement aux exigences fixées lors de la modélisation préliminaire de notre système et nous conforte dans le choix de l'utilisation de ce design pattern. De la sorte, plutôt que sur le MVC originel, nous nous appuyerons particulièrement sur les améliorations du design pattern MVP telles que présentées par Bower et McGlashan (2000). Par ailleurs, on remarquera que le formalisme UML (Object Management Group, 2014e) est bien adapté à la modélisation des éléments composant ces différentes couches logicielles (Mahmoud & Maamar, 2006). En effet, certains de ses stéréotypes sont similaires aux notions mises en avant par le modèle MVC.

2) Les systèmes multi-agents

Nous avons déjà précisé à quel point, dans le contexte de développement d'une PAM, il était important de garantir l'interopérabilité des matériels supportés. D'autre part, on observe que cela doit aussi se vérifier au niveau de l'architecture logicielle du framework (Hatcliff et al., 2012). En effet, les composants logiciels, en fonction de la couche du framework auxquels ils appartiennent, doivent implémenter des interfaces spécifiques à leur niveau de granularité ainsi que d'autres relatives aux fonctionnalités qu'ils assurent. Les composants doivent ainsi pouvoir être remplacés par d'autres, de la même couche logicielle, aptes à assurer les mêmes fonctionnalités. Cette propriété des composants est primordiale pour notre système distribué destiné à être adaptable à des multiples contextes d'utilisation et organisations différentes. On peut d'ailleurs remarquer que le paradigme MVC est adapté à la modélisation d'applications distribuées (Gupta & Govil, 2010), cependant ce design pattern n'offre qu'un découpage assez grossier de l'architecture, et donc ne peut permettre de traduire tous les niveaux d'interopérabilité auxquels nous aspirons. Nous avons donc cherché et trouvé un paradigme de développement ayant démontré sa pertinence pour la conception de systèmes distribués tels que le nôtre : la modélisation multi-agents (pour des revues consulter (Green et al., 1997; Nwana, 1996; Wooldridge & Jennings, 1995a)). Ces

agents étant des composants du système, leur étude nous conduit à nous intéresser à la microarchitecture de ce dernier (Boissier, Gitton, & Glize, 2004; Hollingsworth & Weide, 1994; Noël, 2012; Shehory, 2001).

Dans le domaine du logiciel, le concept d'agent, qui est apparu avec le développement de l'intelligence artificielle (Hewitt, 1977), peut s'avérer un peu difficile à définir (Franklin & Graesser, 1997). Il consiste généralement à disposer d'un programme (ou fragment de programme) autonome qui peut exhiber un comportement présentant des similarités avec celui d'un humain (Zhen Zhu & Wang, 2006). Les agents logiciels permettent de partitionner efficacement l'espace du problème d'un système complexe (Cao, Li, Zhang, & Dai, 2003; Jennings, 2001; Kirn & Gasser, 1998). Par agrégation, s'appuyant explicitement sur une organisation collective où les agents jouent des rôles particuliers (Picard, 2004), ces composants forment des **Systèmes Multi-Agents**. De tels SMA peuvent s'appliquer à trois grandes familles de problèmes (Boissier et al., 2004) :

- Simuler, c'est-à-dire effectuer la modélisation et la reproduction de phénomènes dans le but d'en expliquer ou comprendre le comportement (Meurisse, 2004; Treuil et al., 2008).
- Résoudre, c'est-à-dire procéder à la résolution de problèmes, pour lesquels il n'existe pas de modèle analytique global et satisfaisant, au travers d'une intelligence artificielle distribuée (Ferber, 1995; Shehory, 1998).
- Intégrer, c'est-à-dire réaliser l'hybridation entre des systèmes informatiques, mécaniques et humains d'une manière flexible et cohérente (Huhns & Singh, 1998; Ocello, 2003; Sabouret, 2009).

A ce titre, les SMA sont fréquemment utilisés en contexte clinique à des fins d'aide à la décision (comme par exemple (Czibula, Czibula, Cojocar, & Guran, 2008; Sayyad Shirabad, Wilk, Michalowski, & Farion, 2012)) et en particulier pour effectuer des simulations (comme par exemple (Shtiliyanova, 2012)). Les agents qui les constituent se veulent d'ailleurs être réutilisables, partageables, et indépendants d'une plate-forme particulière (Lanzola et al., 1999; Nwana, 1996), et ils peuvent exister indépendamment les uns des autres (Oja, Tamm, & Taveter, 2001). Chaque agent poursuit un objectif et opère dans un environnement où il interagit avec d'autres agents au moyen de langages et de protocoles d'interaction (Bernon, Gleizes, & Picard, 2009; Galland et al., 2014). Sa perception de l'environnement, dont tout agent fait aussi partie, s'effectue au travers de capteurs (Franklin & Graesser, 1997). D'autre part, les agents ont la capacité d'agir relativement intelligemment (Fox, Beveridge, & Glasspool, 2003; Wooldridge & Jennings, 1995b) et d'être potentiellement capables d'apprendre, d'une façon plus ou moins similaire à celle d'un être humain. En effet, une telle entité autonome et flexible peut récupérer des informations de l'environnement et les traiter pour parvenir à décider de ce qu'elle doit faire (Krüger, Schäfer, Tewes, & Würtz, 2004). Il est d'ailleurs possible que les agents conservent des représentations de leur environnement ainsi que de leur propre état mental (Shoham, 1993). Les agents offrent un niveau d'abstraction au-dessus des objets ainsi qu'une vision fonctionnelle du système en dissimulant les technologies qu'ils utilisent (Wooldridge, 1997). Ils peuvent être organisés en vue d'accomplir des tâches de manière collaborative, ainsi les intelligences individuelles des agents peuvent s'agréger pour constituer une intelligence collective et distribuée (Ferber, 1995; Green et al., 1997) et offrir des fonctionnalités émergentes (Picard, 2004). Par ailleurs, il est important de signaler que le concept d'agent, au travers de sa capacité à paralléliser les traitements, est apte à la réalisation d'applications temps réel (Ocello, 2003).

Il faut bien comprendre qu'un agent est différent d'un objet (Shoham, 1993). En effet, un objet est une entité passive qui se trouve dans l'attente d'être invoqué par des messages ou appels, alors qu'un agent est une entité active qui peut apprendre au sujet de son environnement et est apte à réagir aux changements de ce dernier (Mahmoud & Maamar, 2006). D'autre part, alors que le sens des messages peut varier d'un objet à l'autre, les agents utilisent un langage commun (Genesereth & Ketchpel, 1994). Les tâches effectuées par les agents peuvent, d'autre part, être assimilables à des processus cognitifs (Zhen Zhu & Wang, 2006) d'où le mimétisme possible entre les performances du système et un comportement humain (Franklin & Graesser, 1997). Le concept d'agent peut ainsi être vu comme une extension, de plus haut niveau d'abstraction, de celui d'objet (Wagner, 2003b). On pourra aussi signaler qu'un comportement d'un agent se résume en une séquence d'actions effectuées par l'agent en réponse à un événement particulier, d'une provenance externe ou interne à l'agent (**Figure 19**). Ces actions peuvent être de contacter d'autres agents ou attendre que ces derniers le fassent, ou alors d'effectuer un quelconque traitement pouvant par exemple mener à la manipulation de l'environnement à l'aide d'effecteurs. Nous remarquerons qu'en UML on peut représenter un tel comportement d'agent à l'aide du concept d'activation (Oja et al., 2001). Faire avantageusement réaliser des traitements aux agents passe par la combinaison de quatre facteurs (Cao & Dai, 2003; Cao et al., 2003) :

- La multiplicité des types d'agents, qui est présentée dans la sous-sous-section **Les différents types d'agents et d'organisations d'agents**.
- Le couplage d'algorithmes et d'interactions, que nous abordons dans la sous-sous-section **La communication entre agents**.
- La multiplicité des mécanismes de fonctionnement, qui est l'objet de la sous-sous-section **De la microarchitecture à la macro-architecture**.
- La multiplicité des design patterns, qui sera finalement traitée dans la sous-section **La perspective du méta-système**.

a) Les différents types d'agents et d'organisations d'agents

Les SMA sont particulièrement appropriés dans le cas d'interactions multimodales (Gonzalez-Sanchez et al., 2011; Krüger et al., 2004; Zhu & Wang, 2006) et dans le cas de systèmes distribués (Cao et al., 2003; Lanzola et al., 1999; Singh, Ismail, Haron, & Yong, 2005). Leur mise en œuvre assure une grande durée de vie au logiciel en permettant l'évolution continue grâce à l'interopérabilité qu'offrent les agents (Genesereth & Ketchpel, 1994; Genesereth, 1997). Ces propriétés sont d'ailleurs celles d'un système qu'on pourrait qualifier d'« ouvert » (Cohen, Cheyer, Wang, & Baeg, 1994; Demeyer, Meijler, Nierstrasz, & Steyaert, 1997), c'est-à-dire d'un système dans lequel les composants ne sont pas a priori connus et peuvent évoluer, apparaître ou disparaître au cours du temps (Shehory, 1998)(pour exposé complet sur ce type de systèmes, consulter (Sabouret, 2009)). La mise en place de tels systèmes est évidemment plus complexe que dans le cas de systèmes fermés (Rodríguez, Pérez-Lancho, De Paz, Bajo, & Corchado, 2009).

Les agents sont communément rangés en différents types d'entités disposant de caractéristiques propres, comme nous l'avons présenté brièvement. Ces multiples

caractéristiques peuvent être décrites selon différents axes (Franklin & Graesser, 1997; Giorgini & Henderson-Sellers, 2005; Nwana, 1996; Wooldridge, 1997) :

- La capacité ou non à transiter à travers un réseau, les séparant respectivement en agents mobiles (pour un exemple consulter (Mitrović & Mena, 2003)) et statiques.
- La présence ou non d'une représentation interne de leur environnement et d'états mentaux, les répartissant respectivement en agents cognitifs et réactifs. D'autre part, leur niveau d'aboutissement de la compréhension de leur environnement va conditionner :
 - Leurs croyances, leurs désirs et leurs intentions (Rao & Georgeff, 1991, 1995).
 - Leur niveau d'autonomie par rapport à leur environnement, pouvant dans le meilleur des cas leur permettre d'être des agents proactifs.
 - Leur habilité à coopérer et/ou être en compétition.
 - Leur disposition à apprendre avec le temps.
- Les fonctions qu'assure l'agent au sein du système en accord avec son ou ses désirs/buts, qui peuvent être diverses et multiples ainsi qu'évoluer dynamiquement.

L'ensemble de ces caractéristiques, au travers de toutes les possibilités de combinaisons qu'elles offrent, rend les agents utilisables dans un vaste éventail d'applications. Un agent peut d'ailleurs être qualifié d'agent hybride s'il adopte simultanément des comportements relevant de plusieurs combinaisons différentes de ces catégories (Wooldridge & Jennings, 1995a). Un SMA peut être qualifié de système d'agents hétérogènes s'il est constitué d'agents appartenant à plusieurs de ces différentes catégories (Nwana, 1996). On précisera également que les comportements d'agents sont le fondement de la réalisation d'actions par les agents.

Les organisations d'agents permettent de caractériser les ensembles d'agents formant une certaine unité et dont les différents éléments sont subordonnés entre eux dans un ensemble solidaire et dans une activité convergente (Ferber, Gutknecht, & Michel, 2004; Ferber, 1995; Zambonelli, Jennings, & Wooldridge, 2003). Pour apporter des solutions au problème complexe que tente de résoudre le système dans sa globalité, chaque sous-système poursuit un ou plusieurs objectifs et peut ainsi être vu comme lui-même un agent. Les agents ainsi organisés peuvent travailler en équipes de pairs ou sous l'égide d'un gestionnaire (Jennings, 2001)(comme, par exemple, dans les systèmes présentés dans (Cavaleri, Garetti, Macchi, & Taisch, 2000)). Nous avons identifié les types de structures d'organisations suivantes :

- Les hiérarchies : la complexité d'un système prend fréquemment la forme d'une hiérarchie (Adam & Mandiau, 2005) organisant les interactions entre (et dans) des sous-systèmes correspondant à des organisations d'agents (Hübner, Sichman, & Boissier, 2006). Les agents ainsi organisés n'ont d'échanges qu'au travers de cette structure hiérarchique, et il n'est pas nécessaire de les situer au sein du système.
- Les holarchies : une holarchie consiste en l'encapsulation d'agents les uns dans les autres selon des relations de composition (Giret & Botti, 2004; Pichler, 2000). L'organisation holarchique est relativement similaire à l'organisation hiérarchique hormis le fait que l'agent de plus haut niveau contrôle totalement les agents qu'il encapsule (Adam, 2000; Shehory, 1998).
- Les hétérarchies : des agents autonomes évoluant dans un système ouvert peuvent être amenés à s'auto-organiser au sein de structures égalitaires. De la sorte, les fonctions émergentes du système peuvent être définies ou ajustées dynamiquement en fonction de l'évolution de l'environnement (Shehory, 1998, 2001).

On pourra également signaler qu'une organisation d'agents peut être un mélange de ces différentes approches. D'autre part, les organisations d'agents manipulent au travers de ces approches des flux de contrôle et de données qui correspondent à ce qu'on retrouve dans les organisations humaines pour, respectivement, gérer les aspects stratégiques et organisationnels d'un côté et les aspects opérationnels de l'autre (**Figure 6**). Dans le contexte précis du développement de système intégrant des pratiques médicales, les SMA ont d'ailleurs prouvé leur efficacité (Isern, Sánchez, & Moreno, 2010; Nealon & Moreno, 2002, 2003; Sánchez, Isern, & Moreno, 2010). On notera effectivement qu'ils ont le bénéfice (Fox et al., 2003) :

- De permettre la construction des systèmes experts pour assister individuellement les cliniciens dans leurs pratiques
- D'améliorer la gestion de l'organisation des soins

Ces avantages répondent, respectivement, aux besoins d'intégration verticale et horizontale que nous avons précédemment évoqués (voir le chapitre **Les axes d'amélioration technique de la prise en charge**). Leurs modèles organisationnels, effectivement, les rendent naturellement adéquats pour implémenter des SI (Adam, 2000) et d'aide à la décision (Bouzguenda & Turki, 2012). En particulier, leurs capacités à se comporter de manière réactive ou proactive les rendent respectivement aptes à fournir une aide à la décision aussi bien passive qu'active. Nous sommes donc confiants quant au choix de ce paradigme de modélisation pour supporter les fonctionnalités de notre propre PAM, comme le confirme le framework conçu par Wilk et al. (2008).

Les organisations d'agents servent à définir des accointances entre agents, c'est-à-dire les relations qu'ils entretiennent, qui ne sont pas nécessairement réciproques/symétriques (Hewitt, 1977). Pour s'affirmer, ces relations nécessitent que les agents puissent communiquer. C'est pourquoi, nous allons maintenant présenter les façons qu'ont les agents de communiquer et comment cela se traduit au niveau de leur architecture.

b) La communication entre agents

Dans le but de coopérer, les agents ont besoin de communiquer, de converser, de négocier (Green et al., 1997; Nwana, Lee, & Jennings, 1996; Nwana, 1996). On parlera d'ailleurs de sociabilité des agents. Effectivement, pour pouvoir évoluer au sein d'un système, un agent a besoin de prendre en considération l'activité des autres agents (Koning & Pesty, 2001). Leur communication s'effectue au travers de messages principalement asynchrones relevant d'un langage déclaratif de communication entre agents (Boissier et al., 2004; Genesereth, 1997; Oja et al., 2001) ou par l'intermédiaire de modifications de leur environnement. On signalera qu'une communication par messages est adaptée aux architectures distribuées alors qu'une communication par appels favorise plutôt la performance. Il sera alors possible de choisir le moyen de communication le plus adapté en fonction de la nature d'une interaction. Les messages sont autant d'événements représentatifs de l'état, à différents niveaux de granularité, du système (Qiu, 2005). Les composants du système, que sont les agents, vont ainsi générer et consommer ces événements afin d'induire le comportement d'ensemble du système. Les besoins d'interopérabilité et de réutilisabilité des composants de notre PAM nous poussent à les faire disposer d'interfaces fortement standardisées et génériques (Arney et al., 2014; Jalender et al., 2012). Ces interfaces standardisées doivent être ainsi applicables

à différents niveaux de granularité du système et permettre, par assemblage de composants les exhibant, la description de processus métiers. Un composant peut être réutilisable seulement si son comportement est complètement spécifié de telle sorte à être appréhendable, de façon non ambiguë, par des développeurs.

La communication entre agents peut être strictement directe (Koning & Pesty, 2001) ou se faire au travers de l'environnement à l'aide d'un médiateur/facilitateur (Finin, Fritzson, McKay, & McEntire, 1994; Genesereth, 1997; Shehory, 1998) tel qu'un routeur, un broker (FIPA, 2002c)(c'est-à-dire que l'on utilise le pattern « Publish-Subscribe » (Qiu, 2005)) ou encore un tableau noir ou blanc (Corkill, 2003; Occello, 2003; Reilent, 2012; Rudenko & Borisovs, 2007)(**Tableau 4**). Comme les agents du système, ces médiateurs/facilitateurs peuvent être distribués (Cohen et al., 1994). Un médiateur/facilitateur peut typiquement être utile lorsqu'un agent doit envoyer un message mais ne « sait » pas exactement à quel autre agent l'adresser. Il faudra néanmoins noter qu'une communication utilisant un routeur peut être considérée comme relevant de la communication directe dans la mesure où le ou les destinataires des messages sont identifiés. On signalera finalement que ces différents moyens de communication entre agents font l'objet de descriptions formelles à l'aide de patterns (Juziuk, Weyns, & Holvoet, 2014). Dans ce manuscrit, nous ne détaillerons pas plus en profondeur les différents types de fonctionnement de ces médiateurs/facilitateurs.

Type de médiateurs/facilitateurs	Type de communication	Mode d'interaction
Routeur	Directe	Messages
Broker	Indirecte	Messages
Tableau noir/blanc	Indirecte	Environnement

Tableau 4 : Les différents types de médiateurs/facilitateurs utilisés pour les SMA

Les agents se placent alors comme des médiateurs des interactions (comme par exemple dans (Cesta & D'Aloisi, 1996; Gleizes & Glize, 2002)), à la fois entre les composants du système qu'entre le système et ses utilisateurs, qui peuvent aussi être vus comme des agents, ou encore entre différents utilisateurs. A ce titre, ces agents doivent disposer de référentiels de connaissances communs pour pouvoir communiquer efficacement (Coiera, 2001). De tels référentiels prennent fréquemment la forme d'ontologies (Bergenti & Poggi, 2000; Labrou, Finin, & Peng, 1999). Il faut aussi préciser que la communication s'étudie selon différents aspects (Finin et al., 1994) :

- La syntaxe, qui est caractéristique du langage employé pendant l'interaction.
- Le pragmatisme, qui est spécifique à un contexte d'interaction.
- La sémantique, qui est propre à l'interprétation faite par un agent de messages échangés lors d'une interaction dans un contexte donné.

Basés sur la théorie des actes de langage (I. A. Smith & Cohen, 1996), qui permet la modélisation de processus de travail coopératifs (Nurcan, 1996), il existe différents formats standard de communication disposant chacun de leur syntaxe propre (Chaib-draa & Dignum, 2002; Green et al., 1997; Koning & Pesty, 2001; Labrou et al., 1999). Ces standards sont issus aussi bien du milieu académique que du milieu industriel. On pourra citer, non exhaustivement, les langages suivants :

- ARCOL (Sadek, 1991) développé à France Telecom.

- Cool (Kolb, 1995) développé dans le cadre du projet CoMMA (DFKI GmbH, 1998).
- FIPA ACL (FIPA, 2002b; Poslad, 2007) développé par FIPA (IEEE Computer Society, 2014).
- IL (Demazeau, 1995) développé au LIG par l'équipe Magma (Magma Team, 2011).
- KQML (Finin et al., 1993, 1994; Genesereth, 1997) développé par DARPA KSE (Defense Advanced Research Projects Agency, 1993).

Les deux plus utilisés sont KQML (par exemple utilisé dans (Hawryszkiewicz & Debenham, 1998; Hübner et al., 2006; Lanzola et al., 1999)), qui n'est cependant plus maintenu, et FIPA ACL (par exemple utilisé par la plate-forme JADE (Bellifemine, Caire, Poggi, & Rimassa, 2003, 2008)). Ils sont relativement identiques syntaxiquement, cependant ils diffèrent du point de vue de leur sémantique/philosophie, ce qui ne permet pas de disposer d'un mécanisme systématique de traduction de l'un à l'autre (Chaib-draa & Dignum, 2002; Labrou et al., 1999). Néanmoins, d'une manière générale, on retiendra qu'un message impliqué dans des interactions entre agents est caractérisé par :

- Un langage/standard de communication, qui précise la syntaxe utilisée dans le cadre de l'interaction.
- Une performative/primitive/intention, qui doit appartenir véritablement au lexique du langage employé.
- Un protocole/discussion, qui plus précisément devrait faire référence à une instance de classe de protocole d'interaction permettant ainsi de désambigüiser les échanges entre des agents qui seraient engagés simultanément dans plusieurs échanges se conformant à des protocoles similaires, voire identiques.
- Un expéditeur, qui fait référence à un agent particulier, à l'aide d'un nom/identifiant, localisé dans un environnement d'exécution précis.
- Un destinataire, ou éventuellement plusieurs.
- Des contenus/informations/données, qui peuvent, par exemple, être structurés à l'aide d'un langage comme KIF (Genesereth & Fikes, 1992), FIPA SL (FIPA, 2002d), OKBC (SRI International, 1997), Prolog (Colmerauer & Roussel, 1996; Kowalski, 1988), RDF (World Wide Web Consortium, 2014) ou encore SQL (International Organization for Standardization, 2011).

L'aspect pragmatique de la communication est régi par des protocoles d'interaction qui sont des spécifications des interactions entre agents (Huget, 2001; Koning & Pesty, 2001). On signalera que les langages d'interaction, précédemment énumérés, en permettent généralement la description. Ces protocoles impliquent au moins deux agents (comme précisé dans (Barbuceanu & Fox, 1995; Coriat, 1993)). Effectivement, chaque message échangé entre agents a un expéditeur et au moins un destinataire (Boissier et al., 2004). Un protocole définit les rôles tenus par les agents dans le cadre d'une interaction (Zambonelli et al., 2003). En particulier, il stipule par lequel d'entre eux il peut être initié et il précise les comportements que les agents peuvent adopter dans le contexte de cette interaction ainsi que les conditions de début et de fin des échanges (Huget & Koning, 2003). On signalera que, dans le cadre d'une interaction, il existe un seul et unique agent initiateur (Huget & Odell, 2005). Nous signalerons que le design pattern MVC, qui définit un certain nombre de règles concernant les échanges entre des agents du Modèle, du Contrôleur et de la Vue (**Figure 16**), conditionne fortement la définition des protocoles entre agents appartenant à des couches applicatives différentes (Krasner & Pope, 1988). On fera finalement remarquer que définir efficacement une telle description de la coordination entre agents (Nwana et al.,

1996), en terme d'utilisation du réseau au travers de la fréquence des échanges, n'est pas une chose triviale, en particulier si l'environnement évolue dynamiquement (Lesser & Corkill, 2014).

Le critère sémantique de la communication va, quant à lui, être géré au niveau des agents et leur est spécifique. Effectivement, chaque agent a à sa charge de prendre les mesures qui s'imposent par rapport à une interaction donnée en fonction de ses états internes. Ces prises de décision de l'agent vont s'exprimer à travers l'expression de comportements (Figure 17). Ces derniers vont avoir pour effet la réalisation d'actions par l'agent (Figure 19).

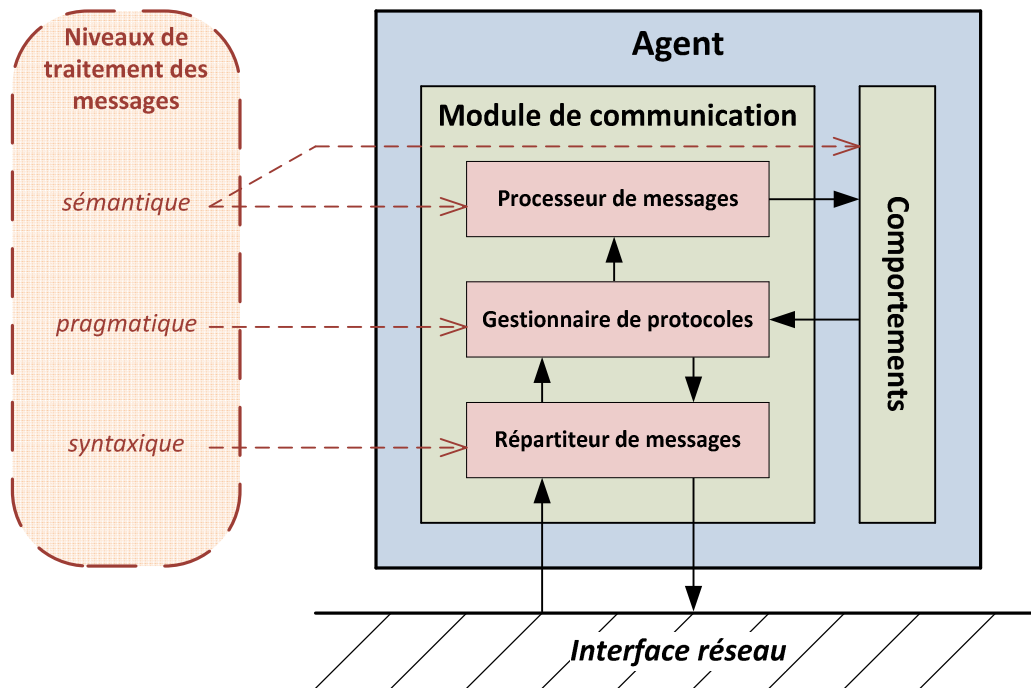


Figure 17 : Schéma de la microarchitecture du module de communication entre agent

Pour tenir compte de ces différents aspects de la communication inter-agent, la microarchitecture des agents évoluant au sein de notre PAM doit permettre la gestion des messages aux niveaux syntaxique, pragmatique et sémantique (Lanzola et al., 1999; Wangler & Backlund, 2005). C'est en partie au travers de ces échanges de messages que les états des agents vont évoluer (Peng et al., 1998). Dans cette optique, nos agents vont être dotés des composants suivants (Figure 17) :

- Un répartiteur de messages, permettant l'envoi et la réception de messages. Ces traitements s'effectuent au regard d'un langage donné.
- Un gestionnaire de protocoles, permettant de replacer les messages dans différents modes de conversations structurées. Ces analyses sont propres aux protocoles d'interaction employés.
- Un processeur de messages, permettant d'interpréter le sens des messages et de déterminer quelle(s) action(s) l'agent doit réaliser. Ces actions sont naturellement propres à l'agent en question et leur ordre de traitement peut dépendre d'un niveau de priorité.

L'interface réseau n'implique pas nécessairement de faire appel à une communication à l'aide de sockets : il peut aussi s'agir, par exemple, d'une communication reposant

directement sur des fonctions du noyau de l'OS (comme, par exemple, en utilisant les fonctions d'envoi et de réception de messages de Win32 (Microsoft, 2010)).

c) De la microarchitecture à la macro-architecture

Au travers de notre framework, nous voulons proposer des classes représentant l'agent et ses composants afin qu'ils puissent être spécialisés en vue d'un usage spécifique. A ce titre, la **Figure 18** présente l'ensemble des éléments, décrits dans cette sous-section, sous la forme d'un diagramme de classes (on peut retrouver des prototypages similaires dans (Cesta & D'Aloisi, 1996; Lanzola et al., 1999; Oja et al., 2001)).

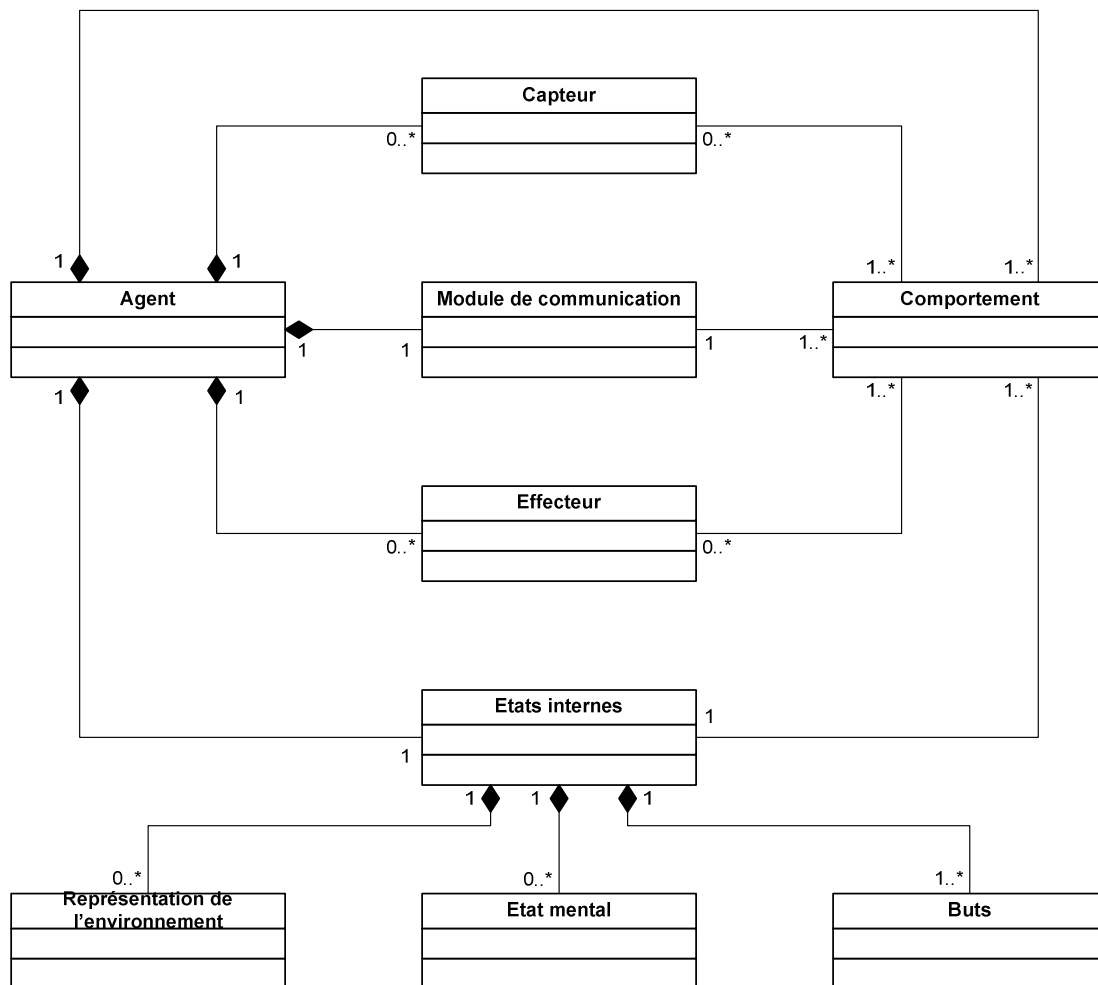


Figure 18 : Diagramme des classes constituant un agent

Ce diagramme comprend ainsi (pour une proposition de modélisation fine de certains de ces composants, consulter (van Breemen, 2003)) :

- L'agent, c'est-à-dire l'entité anthropomorphe de plus haut niveau conceptuel.
- Le module de communication, qui est le composant permettant les interactions entre agents au travers de messages.
- Les capteurs, qui sont les modules permettant à un agent de recueillir des données dans son environnement.

- Les effecteurs, qui sont les modules au travers desquels un agent agit sur son environnement.
- Les comportements, c'est-à-dire les modules permettant à un agent d'effectuer des tâches pouvant requérir l'utilisation de capteurs et/ou d'effecteurs.
- Les états internes, qui se composent de divers types d'informations indispensables à l'exécution de ses comportements :
 - o La représentation de l'environnement, qui correspond à un modèle personnel qu'un agent a de son environnement.
 - o L'état mental, qui permet à un agent de conserver des représentations propres à sa condition et à l'évolution de cette dernière.
 - o Les buts, qui représentent les motivations profondes d'un agent.

La **Figure 19** montre la microarchitecture complète d'un agent, dont une description similaire est proposée par Nahm et Ishikawa (2005), en accord avec notre diagramme de classes. On pourra remarquer qu'un méta-modèle complet et générique est aussi proposé par Treuil et al. (2008) ainsi qu'une implémentation par Nwana, Ndumu, Lee et Collis (1999).

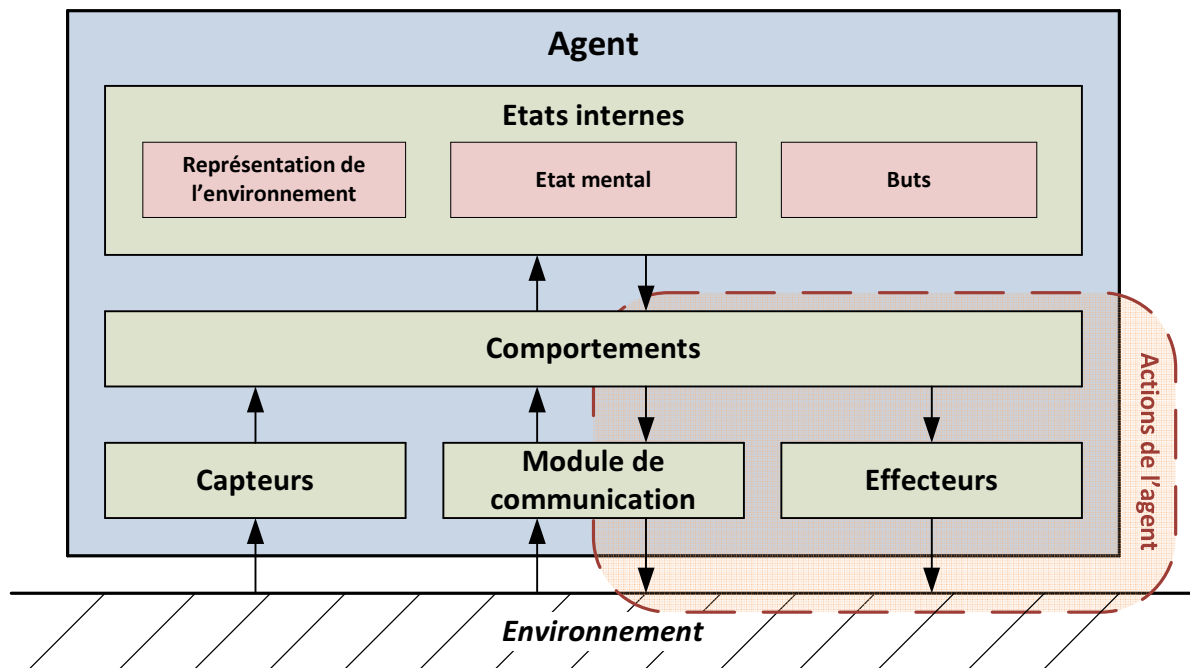


Figure 19 : Schéma de la microarchitecture globale d'un agent

On remarquera que cette microarchitecture est composée d'éléments de granularités variées. Ainsi, d'un point de vue de la macro-architecture du système il est important de pouvoir cantonner ces différents types d'éléments à des couches applicatives appropriées. Il est pour cela possible de concevoir une architecture MA s'appuyant sur trois couches (Oja et al., 2001; Taveter & Tamm, 2001) :

- Une couche binaire
- Une couche orientée objet
- Une couche orientée agent

Les couches de bas niveau, c'est-à-dire binaire et orientée objet, permettent de définir les éléments utilisés dans la description des modules et états internes des agents ainsi que les

informations échangées au cours d'interactions (comme, par exemple, dans (Yanhong, Fang, Caowan, & Yantao, 2002)). Nous ne détaillerons pas ces couches de bas niveau qui sont hors du périmètre de nos travaux. La couche orientée agent offre, quant à elle, une abstraction de la couche orientée objet qui à son tour est une abstraction de la couche binaire. D'autre part, l'expérience montre que le design pattern MVC est propice à l'utilisation d'agents (Gangat, Payet, & Courdier, 2012; Mahmoud & Maamar, 2006). Ainsi nous allons pouvoir intégrer chacune de ces couches à l'architecture déjà définie (**Figure 16**) à l'aide de ce design pattern. Les interfaces entre les éléments des couches du modèle MVC sont ainsi totalement assurées par des agents (**Figure 20**). D'une manière similaire, au sein d'une même couche applicative, au sens MVC, les agents seuls sont habilités à directement interagir.

D'autre part, il est intéressant de signaler que chacune des couches du modèle MVC ainsi affinée ressemble au modèle de collaboration de groupe proposé par Bharadwaj et Reddy (2003). Dans leur framework, visant à soutenir la collaboration dans des environnements hétérogènes, apparaissent trois différents types d'entités :

- Les informations, qui représentent toutes les données impliquées dans la collaboration. Du point de vue de notre système, il s'agit d'informations binaires.
- Les systèmes, qui sont les applications collaboratives utilisées pour un projet. Dans notre cas, ce sont les objets utilisés pour composer les modules des agents.
- Les participants, qui collaborent pour atteindre les buts d'un projet. Dans notre contexte, ces participants sont évidemment les agents.

Conséquence naturelle de l'anthropomorphisme des agents, notre modélisation s'avère conforme à leur framework. Nous pouvons alors affirmer que l'architecture de notre système va permettre d'assurer une collaboration dans un environnement d'agents hétérogènes. D'autre part, d'une manière générale, il est possible de considérer le système comme un gros agent réalisant des actions (conformément à la **Figure 19**) de communication avec l'utilisateur, c'est-à-dire à supporter l'IHM, et d'opérationnalisation sur le reste de l'environnement, ce qui revient à supporter les traitements métiers.

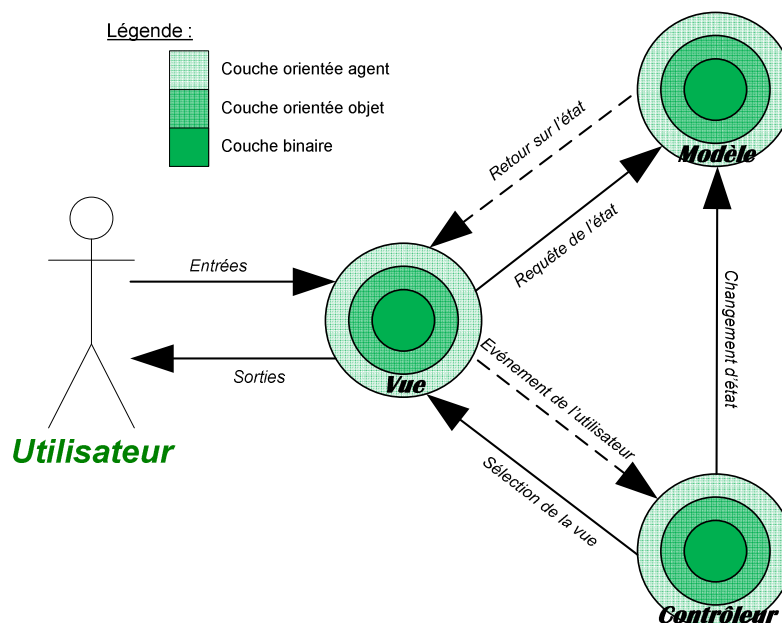


Figure 20 : Schéma des interactions au sein du modèle MVC composé d'agents

A ce point de notre raisonnement, la macro-architecture du framework de notre PAM est ainsi constituée de neuf couches permettant au système :

- D'être distribué et ouvert.
- De disposer d'une séparation logique entre sa présentation et son noyau fonctionnel.
- D'assurer l'organisation collaborative, extensible et robuste d'agents.

On pourra également ajouter que ce découpage garantit un haut niveau d'interopérabilité entre les nombreux composants du système. D'autre part, pour pouvoir s'exécuter, les SMA doivent disposer d'une infrastructure, c'est-à-dire d'un cadre de définition des agents et de leurs organisations (Shehory, 1998, 2001). Cette infrastructure s'accompagne pour cela de différents services (Poslad, 2007). A ce sujet, les standards de la FIPA (FIPA, 2002a, 2004) définissent trois services d'agents indispensables à tout SMA :

- « Agent Management Service », en charge de la gestion des agents dans le système. Il fournit un service de pages blanches c'est-à-dire il référence tous les agents du système (par exemple, comme dans la plate-forme ZEUS (Nwana et al., 1999)).
- « Agent Directory Service », fournissant un service de pages jaunes. Ce service consiste à référencer tous les services du système (c'est un service similaire à ceux présentés dans (Finin et al., 1994; Kolb, 1995)).
- « Message Transport Service », responsable de la communication entre agents. Il s'agit d'un service de médiateur/facilitateur (voir la sous-sous-section **La communication entre agents**).

Ces services font partie de l'environnement commun à tous les agents du SMA (Galland et al., 2014). A ce titre, faire appel à eux, du point de vue d'un agent, relève de la conscience de de ce dernier et de ses capacités perceptives (Sycara, Giampapa, Langley, & Paolucci, 2002). Il est donc possible, pour un agent, d'utiliser ces services de manière synchrone (Sabouret, 2009). En effet, la perception des agents est une tâche temps réel strict (Occello, 2003).

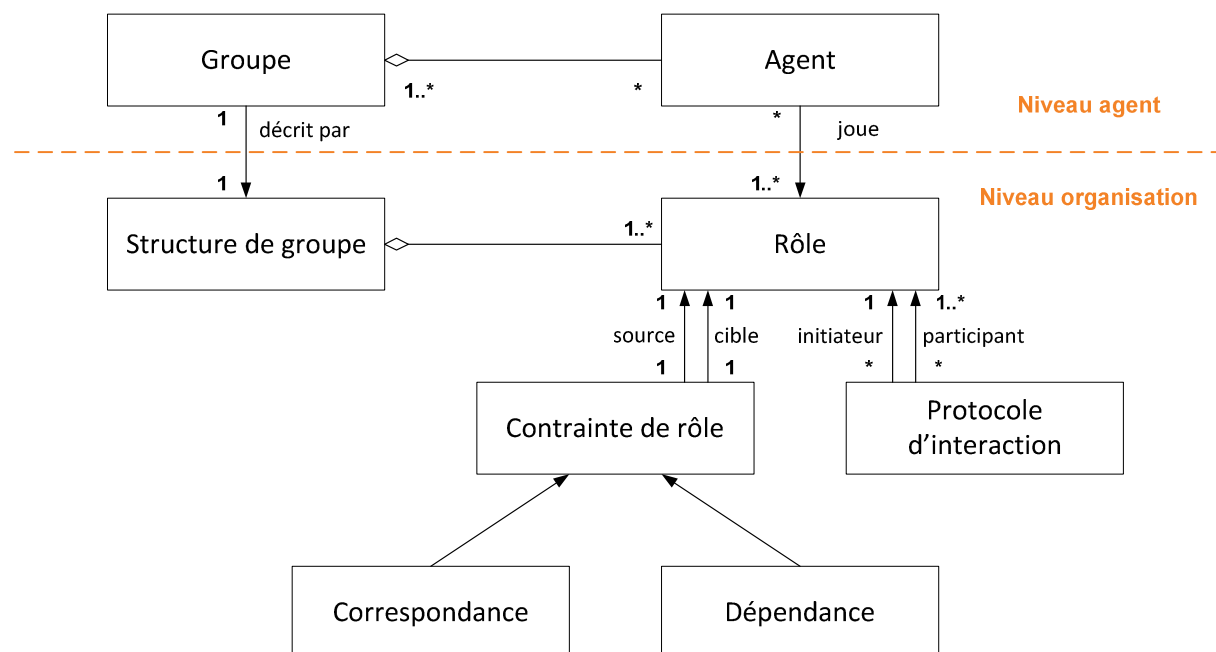


Figure 21 : Diagramme UML du méta-modèle pour représenter des SMA (d'après (Ferber et al., 2004))

Pour finir, nous signalerons qu'il existe des extensions d'UML (telles que MAS-ML (da Silva, Choren, & de Lucena, 2004; da Silva & de Lucena, 2004), AORML (Wagner, 2003a, 2003b), AMOLA (Spanoudakis & Moraitis, 2008) ou encore AML (Červenka, Trenčanský, Calisti, & Greenwood, 2005; Trenčanský & Červenka, 2005)) pour représenter les aspects structurels et dynamiques des agents et des organisations d'agents (Kavi, Kung, Bhambhani, Pancholi, & Kanikarla, 2003; Odell, Parunak, & Bauer, 2003). Ces extensions ont pour vocation d'adapter/étendre des diagrammes UML existants ou d'en proposer de nouveaux. La plus populaire de ces extensions est certainement AUML (Bauer, Müller, & Odell, 2001; Bauer, 2002; Huget & Odell, 2005; Huget, 2004a, 2004b; Odell, Parunak, & Bauer, 2000). Néanmoins, il est possible d'utiliser uniquement les mécanismes d'extension intrinsèques à UML (Lind, 2002). En particulier, on fera remarquer qu'il est possible, depuis la version 2.0 d'UML, de représenter de nombreuses caractéristiques d'agents à l'aide, par exemple, des stéréotypes et des contraintes (Guedes & Vicari, 2009). La norme UML actuelle supporte ainsi nativement des mécanismes d'extensions qui auparavant devaient faire l'objet d'extensions spécifiques (Bauer & Odell, 2005; Bergenti & Poggi, 2000; Dillon, Dillon, & Chang, 2008). Tous les diagrammes définis à l'aide d'UML et de ses extensions destinées aux SMA sont autant de modèles d'analyse que nous pourrions être amenés à utiliser pendant notre application de l'UP7. Ainsi, nous pourrions définir nos propres modèles à l'aide d'un méta-modèle destiné à la représentation des agents et de leurs organisations (comme, par exemple, le modèle AGR (Ferber et al., 2004) , anciennement appelé Aalaadin (Ferber & Gutknecht, 1998), qui est représenté sur la **Figure 21**). Il est à signaler que nous évoquerons, un peu plus loin dans ce mémoire, des méthodes de développement de SMA reposant sur ces types de modèles d'analyse (voir la section **La modélisation globale de notre solution**).

3) La perspective du méta-système

La solution logicielle que nous comptons proposer est un framework, elle devra donc s'appuyer sur des analyses et composants réutilisables (Fayad & Schmidt, 1997; Fortune & Valerdi, 2013; Johnson, 1997). Un framework est ainsi un méta-système dans la mesure où il permet la conception et l'exécution de systèmes. Ce méta-système repose sur un méta-modèle, c'est-à-dire un modèle de langage de modélisation permettant l'expression des modèles dynamiques qui sous-tendent le fonctionnement des systèmes que nous venons d'évoquer (**Figure 10**) (Treuil et al., 2008). Le méta-modèle définit des concepts de modélisation ainsi que leurs propriétés et relations. D'autre part, notre framework n'est pas seulement un framework MA mais un framework de plates-formes d'applications MA. A ce titre, il doit disposer d'une division en couches fonctionnelles similaire à celle d'un SI, où différents traitements/applications de domaines métiers différents peuvent coexister. Par ailleurs, il est notable que la capacité des agents à supporter le fonctionnement d'un SI est avérée, en particulier si ce dernier implique la gestion de processus métiers (Wagner, 2003b). Afin d'assurer cette division en couches fonctionnelles, notre framework devrait disposer d'une architecture comprenant (Bäumer et al., 1997) :

- Les couches du domaine métier, contenant les éléments permettant de décrire tous les concepts fondamentaux impliqués dans les processus métiers. Ces couches visent à éviter des duplications aisément évitables.

- Les couches de la section métier, contenant les éléments propres aux différents types de fonctions accessibles aux utilisateurs du système. Les éléments de cette couche utilisent ceux de la couche du domaine métier.
- Les couches applicatives, correspondant à une projection des différentes fonctions de la section métier à un usage conditionné par l'utilisateur ou le lieu d'utilisation. Les éléments appartenant à ces couches offrent des vues ouvertes sur le système.

De cette manière, notre framework va devoir, pour chacune de ces couches, proposer des méta-structures aptes à supporter la description puis l'instanciation des éléments constitutifs de processus métiers. L'architecture précédemment présentée peut aussi être étendue par l'adjonction de ces nouvelles couches (**Figure 22**). Du point de vue de l'implémentation, on pourra parler de packages pour désigner les regroupements d'éléments appartenant à un même concept (pour une illustration, dans notre contexte d'application, voir la section **La conception**). L'architecture ainsi proposée se présente comme un moyen efficace d'assurer un bonne modularité des éléments du système et faciliter son évolution (Ferrara, 1996). Une telle approche permet ainsi de concevoir un système complexe voué à évoluer, à s'étendre, et qui sera apte à s'adapter à de nouvelles exigences fonctionnelles et techniques au fur et à mesure de son développement (Hanseth & Lytinen, 2004). Notre framework se présentera alors comme une véritable « boîte à outils » de composants réutilisables. Une sorte d'indexation de ces composants peut d'ailleurs être réalisée afin d'en faciliter l'exploitation (Fortune & Valerdi, 2013).

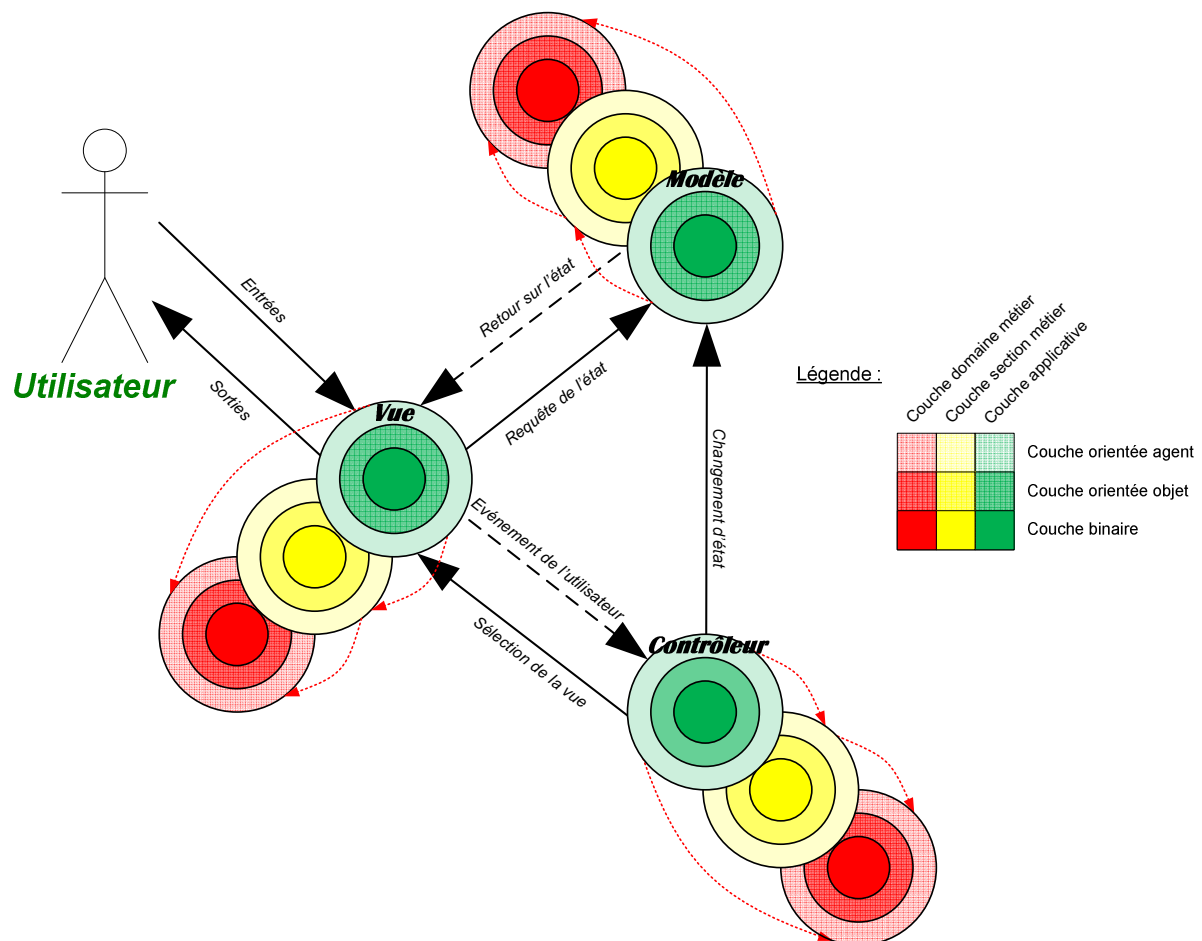


Figure 22 : Schéma des interactions dans un framework composé d'agents reposant sur MVC

Dans les faits, certains éléments, en particulier ceux des couches binaire et orientée-objet, pourront se retrouver à l'identique dans plusieurs couches. A ce titre, le framework pourra disposer d'un ensemble de packages dit « commun » dans lequel des templates d'éléments génériques – appartenant indépendamment aux couches binaire, orientée-objet ou orientée-agent – pourront servir de base à n'importe quels éléments des autres packages et types de packages.

Un tel framework trouve spécialement un intérêt auprès des systèmes existants, qui sont possiblement intégrables (Madni & Sievers, 2014; Maier, 1998). Effectivement, il est possible à l'aide d'agents, tels que nous les avons définis, d'offrir des abstractions de ces systèmes (consulter par exemple (Cesta & D'Aloisi, 1996; Peng et al., 1998)). La nette séparation entre le comportement, qui s'exprime au travers de traitements, et les facultés de communication de l'agent permettent d'encapsuler, de réutiliser, tout ou partie de systèmes préexistants et les intégrer à un réseau d'agents relativement aisément (Lanzola et al., 1999). Ce constat prend tout son sens dans la mesure où notre PAM va devoir intégrer les fonctionnalités d'applications déjà développées, et peut-être déjà en cours d'exploitation. Ainsi, afin de servir de mécanisme d'intégration, un agent peut prendre la forme d'un wrapper ou d'un transducteur permettant d'encapsuler un service ou une technologie existante (Genesereth, 1997; Jennings, 2001). Les nouvelles couches que nous venons d'introduire nous permettent, d'autre part, une intégration à différents niveaux de granularité. De tels agents pourront être vus comme servant véritablement de middleware (Ferrara, 1996). Cela nous amène à constater que notre framework est un framework orienté agent tel que défini par Shoham (1993). En effet, il dispose des trois éléments :

- Un langage formel pour décrire les états mentaux (tel qu'un langage orienté-objet).
- Un langage de programmation et de communication interprété (tel que KQML ou FIFPA ACL).
- Un moyen d'intégrer d'autres composants sous la forme d'agents.

Dans le cadre de notre PAM, cette définition d'un framework MA doit être adaptée afin de prendre en considération notre contexte d'application. A ce titre, nous devons proposer des modèles d'analyse permettant de clarifier certaines organisations architecturales dans le but de permettre le développement efficace d'applications aux composants interopérables. En particulier, nous allons, en nous appuyant spécifiquement sur le modèle MVP (Potel, 1996), répondre aux questions fondamentales suivantes :

- Comment les utilisateurs interagissent-ils avec les données ?
- Comment les données sont-elles gérées ?

Dans cette optique, nous allons présenter des modèles d'analyse prenant la forme de design patterns d'agents volontairement très génériques afin de pouvoir systématiquement être utilisés. L'ingénierie logicielle propose de nombreux patterns destinés à la programmation de SMA (pour une revue systématique consulter (Juziuk et al., 2014)) en grande partie parce que les agents sont utilisés dans de très nombreux contextes applicatifs différents (Boissier et al., 2004). Sauvage (2004a, 2004b) propose une classification de ces patterns, basée sur leurs niveaux d'application et leurs origines, contenant quatre types :

- Les méta-patterns, qui s'appliquent à un haut niveau conceptuel, comme par exemple pour décrire la macro-architecture.
- Les patterns métaphoriques, qui s'inspirent de modèles relevant de domaines spécifiques, tels que la biologie ou la physique.

- Les patterns architecturaux, qui s'appliquent à l'architecture des agents c'est-à-dire à la microarchitecture, telle que définie précédemment.
- Les anti-patterns, qui apportent des propositions de correction pour des conceptions et/ou implémentations existantes et présentant certaines faiblesses.

Dans ce domaine, Lind (2003) a proposé un patron pour décrire et documenter les patterns d'agents et de protocoles d'interaction entre agents. Effectivement, l'étude des patterns permet de répondre efficacement aux problèmes qui se posent lorsque l'on doit concevoir l'architecture d'un SMA (Noël, 2012). L'utilisation de patterns adaptés permet d'ailleurs d'intégrer les organisations à l'aide des technologies de l'information (Kirn & Gasser, 1998). Par l'intermédiaire de ces patterns, les agents se voient attribuer des rôles (Hübner et al., 2006) au sein de l'organisation intégrée par le système (Kirn & Gasser, 1998).

Dans les sous-sous-sections qui suivent, nous allons décrire des patterns concernant la méso-architecture de notre système (comme, par exemple, dans (Aridor & Lange, 1998)). Il s'agit ici de proposer des solutions architecturales locales à certaines parties du système sous la forme d'organisations d'agents (Schillo, Fischer, & Klein, 2000). Ces solutions sont effectivement intermédiaires entre celles macro-architecturales, car elles présentent des solutions architecturales locales et non globales, et celles micro-architecturales, car elles ne s'intéressent pas aux descriptions individuelles des agents. On signalera qu'elles prendront place dans l'architecture globale de notre solution sous la forme de middlewares disposant de comportements propres à leur périmètre fonctionnel (comme, par exemple, ceux de RETSINA (Sycara et al., 2002; Sycara, Paolucci, van Velsen, & Giampapa, 2003)).

a) Modéliser les interactions avec les utilisateurs

L'interface utilisateur est le moyen d'échange d'informations entre humains et ordinateurs, ce qui en fait une partie importante du système informatique. A l'heure actuelle, cette interface utilisateur tend à devenir perceptuelle et multimodale (voir la section **Les interfaces utilisateur**). Il s'avère que supporter de telles interactions à l'aide d'une modélisation multi-agents est particulièrement adapté (Gonzalez-Sanchez et al., 2011; Krüger et al., 2004; Oviatt & Cohen, 2000; Sanchis, Julián, Corchado, Billhardt, & Carrascosa, 2014). En effet, lorsqu'ils interagissent avec leur environnement, les humains utilisent de nombreux canaux de perception et d'intervention/action en parallèle. Une personne est ainsi équivalente à un supercalculateur traitant et produisant des masses d'informations à partir et sur ces différents canaux. Il paraît donc naturel de vouloir disposer d'une IHM apte à travailler simultanément sur ces différents canaux sans contraindre l'utilisateur à interagir avec la machine d'une manière séquentielle (Zhu & Wang, 2006). De telles interfaces sont ainsi plus naturelles, intuitives, adaptatives et transparentes car elles rendent l'interaction homme-machine plus proche de la façon qu'ont les gens d'interagir entre eux et avec leur environnement (Turk, 1998). Il en existe, par exemple, pour l'analyse de la parole de manière couplée audio-vidéo (Srivastava & Dev, 2013) ou la reconnaissance de gestes (Krüger et al., 2004). D'autre part, les SMA peuvent disposer d'agents intelligents d'interface (Lieberman & Selker, 2003), c'est-à-dire d'agents qui s'adaptent à leur utilisateur dans le but (Green et al., 1997) :

- De l'assister lorsqu'il interagit avec le système, c'est-à-dire de lui fournir une aide appropriée ou d'automatiser certaines tâches.

- D'établir son profil et d'adapter les composants et fonctionnalités présentes dans l'interface, pour cela ces agents ont besoin de disposer de mécanismes d'apprentissage.

Indépendamment de la nature des interactions supportées par l'IHM, notre framework doit permettre de décrire les interfaces qui seront utilisées par les applications de la PAM. A ce titre, il doit contenir (Myers et al., 2000) :

- Une bibliothèque de composants interactifs
- Un framework architectural pour gérer les interactions entre ces composants

Nous n'allons pas ici rentrer dans l'énumération des composants inclus dans la bibliothèque, cependant nous souhaitons présenter un pattern architectural générique pour les interactions entre composants directement impliqués dans l'interface utilisateur. L'application de ce pattern permettra d'organiser des agents sous la forme de hiérarchies et/ou d'holarchies en conformité avec la structuration qu'on retrouve au sein des éléments de la Vue et du Contrôleur dans le pattern MVC.

Pour clarifier notre discours à propos des interactions avec l'utilisateur, commençons par poser quelques définitions. Premièrement, on parlera de « fenêtre » pour désigner une surface de rendu destinée à l'affichage d'informations sur un écran physique. Les agents qui communiquent directement avec l'utilisateur (ou avec l'environnement), au travers des entrées ou des sorties du système, pourront être appelés « agent d'interface » (Lieberman & Selker, 2003) ou « interacteurs » (Coutaz, Nigay, & Salber, 1996).

Il s'agit ici de définir comment nous allons formellement modéliser les interfaces utilisateur, c'est-à-dire les éléments appartenant à la Vue et au Contrôleur. A ce propos, le design pattern MVP, variante de MVC, nous invite à nous poser trois questions (Potel, 1996) :

- Comment afficher les données ?
- Comment coïncident les événements utilisateur avec des révisions des données ?
- Comment appliquer ces modifications ?

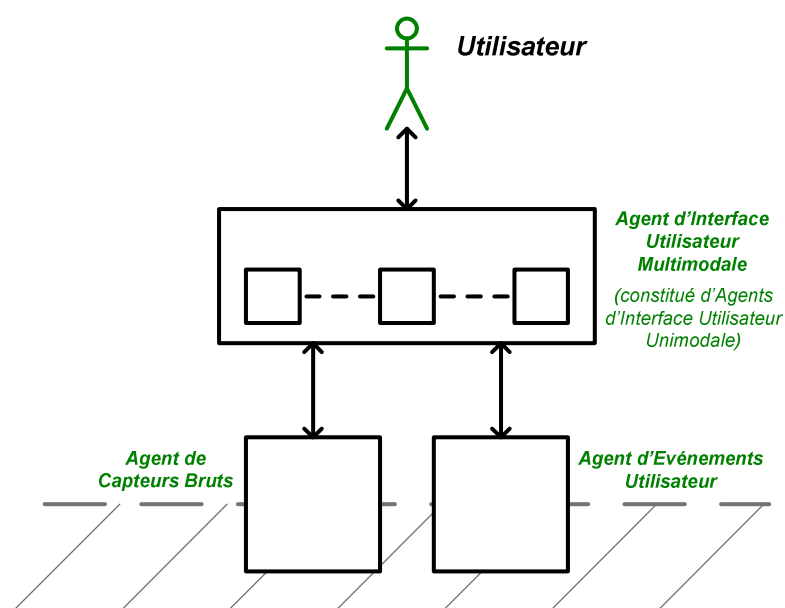


Figure 23 : Représentation d'agents d'interactions multimodales

Pour répondre à ces questions, nous proposons une organisation générique d'agents en charge des interactions multimodales avec l'utilisateur (**Figure 23**).

Dans cette organisation, un Agent d'Interface Utilisateur Multimodale est en charge de l'interaction directe avec l'utilisateur : il s'agit de l'interacteur de plus haut niveau. Il intègre une hiérarchie d'Agents d'Interface Utilisateur Unimodale en charge soit des modalités d'entrées soit de celles de sorties. Ainsi, ces agents supportent différents types d'interactions avec la partie matérielle de l'IHM :

- Les interactions entrantes, c'est-à-dire les actions actives ou passives de l'utilisateur sur le système via un matériel. Par exemple, l'utilisateur peut activement sélectionner des éléments affichés par le système à l'aide d'une souris, ou encore il est possible qu'il envoie passivement des données par l'intermédiaire d'une caméra.
- Les interactions sortantes, c'est-à-dire les actions du système sur l'utilisateur via un matériel. Il peut, par exemple, s'agir de l'affichage ou de tout autre appareil permettant la transmission d'informations en direction de l'utilisateur. C'est par ce biais que des données pourront, le cas échéant, être affichées.

Au travers de ces agents sont échangées les informations relatives à l'interaction entre l'utilisateur et le système. A ce titre, des Agents de Capteurs Bruts et des Agents d'Événements Utilisateur servent d'interface entre la Vue et l'ensemble Contrôleur-Modèle. Ces agents en charge des données d'interaction appartiennent au Contrôleur et peuvent réaliser des fusions de données. Par ailleurs, ils seront en charge de propager les requêtes de modification de données qui pourront porter soit sur les données du domaine soit sur des données d'affichage de l'application.

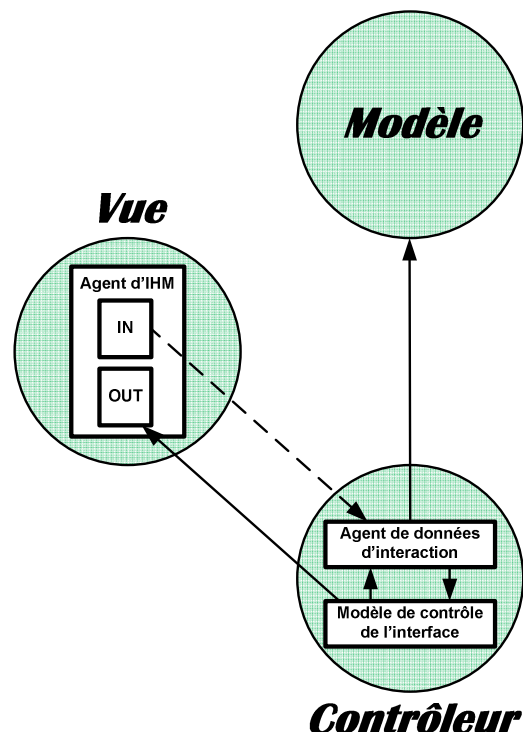


Figure 24 : Représentation d'agents de notre framework impliqués dans les interactions avec les utilisateurs

La **Figure 24** montre comment de tels agents sont répartis dans les couches du framework. D'autre part, on remarquera que, conformément au modèle MVP (Bower & McGlashan, 2000), le modèle de contrôle de l'interface devra être maintenu directement dans le Contrôleur.

Pour finir, nous évoquerons les agents de PAC (Coutaz et al., 1996; Coutaz, 1987; Nigay, 1994) qui ressemblent aux composants de MVC (Sauer & Engels, 1999)(pour un exemple d'utilisation des agents PAC consulter (Clay et al., 2009)). La principale différence entre PAC et MVC est que PAC rassemble toutes ces fonctionnalités au sein d'un même objet alors que MVC les dissémine dans plusieurs (Coutaz, 1987). De la sorte, le paradigme MVC permet de décrire un système plus modulaire qu'avec PAC. Néanmoins, les préceptes concernant l'interface utilisateur orientée agent de PAC peuvent tout à fait être appliqués à notre framework. En particulier, nous pourrions retenir plusieurs règles qui concernent la structure des fenêtres et leur contenu (Coutaz et al., 1996; Nigay, 1994) :

- Un agent, disposant de son propre thread, doit être utilisé pour modéliser la fenêtre principale. A ce titre, notre Agent d'Interface Utilisateur Multimodale devra disposer de son propre thread.
- Une hiérarchie d'agents doit permettre de traduire la relation de filiation instaurée entre fenêtres. Ainsi, les agents ayant des sous-agents doivent maintenir les relations sémantiques entre ces derniers. Ceci peut clairement s'appliquer aux agents de la Vue en charge des modalités entrantes et sortantes. Les agents n'étant pas des feuilles de cette hiérarchie peuvent appartenir au Contrôleur s'ils ne participent pas directement aux tâches de représentation.
- Un concept complexe et/ou particulièrement dynamique est représenté à l'aide d'un agent, sinon un objet utilisé comme attribut d'un agent existant pourra suffire. Bien que nous n'ayons ici représentés que les agents (**Figure 23** et **Figure 24**) ces derniers s'appuient sur des objets dédiés à la conception d'interface utilisateur comme nous l'avons décrit précédemment (**Figure 20**).

Nous allons maintenant nous concentrer sur la manière dont les données vont pouvoir être gérées. Ainsi nous expliquerons plus précisément comment des événements utilisateur provoquent des modifications de données.

b) Modéliser la gestion des données

Dans le domaine de l'accès à l'information, les agents peuvent être utilisés pour accéder à des données distribuées (Kook, Lee, Park, Kim, & Choi, 2011; Wooldridge & Jennings, 1995b). En effet, la morphologie des agents leur permet, par définition, d'être interchangeables et d'offrir ainsi une interopérabilité des données auxquels ils permettent d'accéder. Il est ainsi possible de concevoir, par exemple, un système de bases de données dont les évolutions pourront s'effectuer moyennant un coût et un temps réduits. InfoSleuth™ est un exemple de SMA appliquant ce genre de principes pour la gestion de données (Bayardo et al., 1997; Nodine et al., 2000).

En ce qui concerne la gestion des données, le design pattern MVP nous invite à nous poser trois questions (Potel, 1996) :

- Que sont les données ?

- Comment sélectionne-t-on les données ?
- Comment modifie-t-on les données ?

Pour répondre à la première de ces questions, nous proposons une classification des données selon leur type et selon leur utilité en termes de modélisation (**Figure 25**).

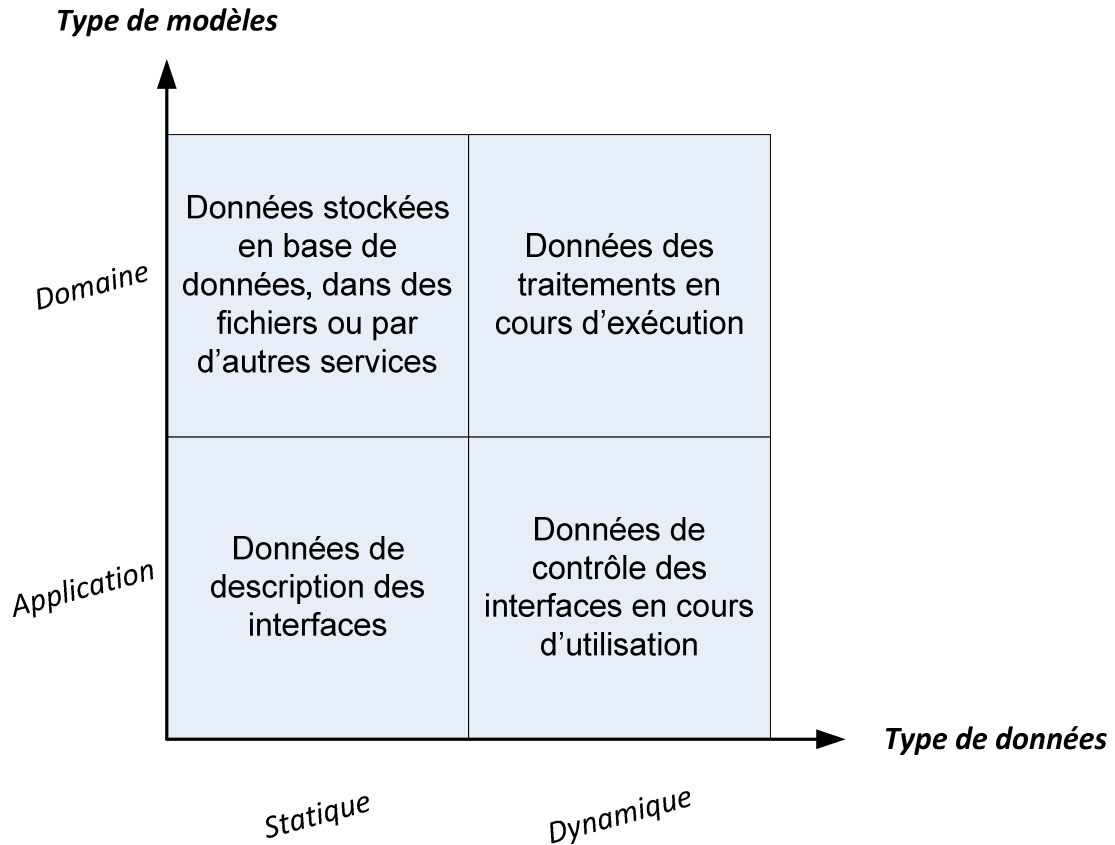


Figure 25 : Schématisation de la classification des différentes données du framework

Il existe de multiples manières de gérer les données dynamiques du domaine. Ces techniques sont propres à la nature des traitements effectués et, à ce titre, il serait délicat et réducteur de vouloir imposer un quelconque pattern. On pourra juste préciser que ces données devront être conservées au sein d'agents appartenant au Modèle ou éventuellement au Contrôleur. Le problème est à peu près le même concernant les données de contrôle des interfaces, c'est-à-dire les données dynamiques liées à l'application. Cependant, elles seront gérées exclusivement dans le Contrôleur (**Figure 24**).

Les données statiques, quant à elles, peuvent aisément faire l'objet de patterns. Effectivement, de telles données peuvent être dissociées du contenu d'exécution. Ainsi, elles peuvent être stockées à l'aide de fichiers, des bases de données ou d'autres types de services. La **Figure 26** présente un design pattern reposant sur des agents pour exploiter de telles données statiques. Ce pattern a pour but de répondre aux deux dernières questions préliminairement énoncées.

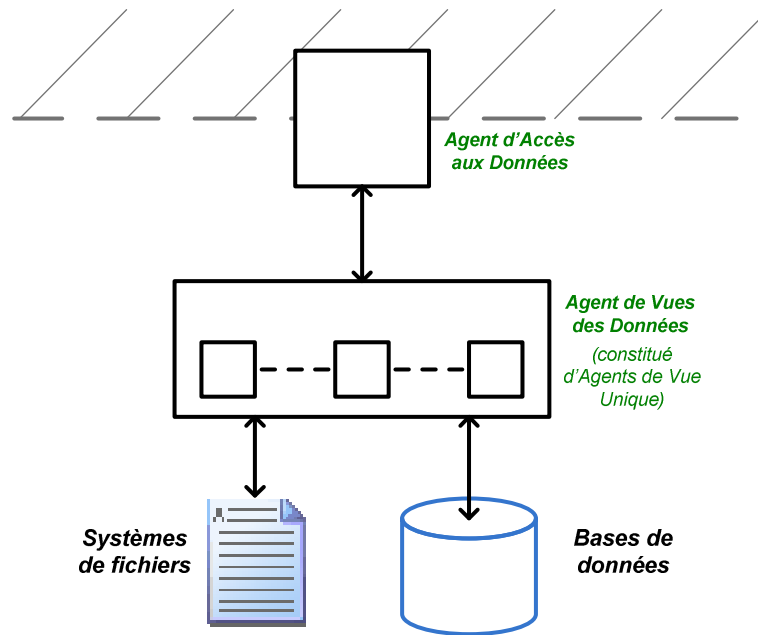


Figure 26 : Représentation d'agents d'abstractions de données

Dans cette organisation, l'agent de vues de données est particulièrement intéressant car il constitue une abstraction des méthodes et technologies de stockage des données. En effet, ce dernier est constitué, au sens holarchique du terme, d'agents de vue unique. Ce sont ces derniers qui procèdent à la lecture et l'écriture des données par et pour le SMA. Ainsi, chacun d'entre eux est spécifique à un type de requêtes, à une technique et à une technologie. Effectivement, chaque agent de vue unique va satisfaire à un besoin particulier de données en utilisant un langage adapté à la récupération de ces données (comme par exemple le SQL (International Organization for Standardization, 2011)) dans des fichiers ou auprès de services (comme par exemple ceux de bases de données telles que MySQL (Oracle Corporation, 2015)). En s'appuyant sur ces agents, l'agent de vues des données n'a alors qu'à exhiber, au travers des messages qu'il accepte en entrée, les différentes requêtes auxquelles il lui est possible de satisfaire. Cet agent centralise de la sorte les fonctionnalités de ses agents subalternes, et c'est lui qui « choisit » auquel ou auxquels il va transférer la requête. Nous sommes ainsi dans la possibilité de changer de technologie, par le biais de cette holarchie d'agents, de façon totalement transparente pour le reste du système, à commencer pour l'agent d'accès aux données. Ce dernier agent peut servir d'interface entre une partie du Modèle, matérialisée par des agents de vues des données, et le conglomerat formé par le Contrôleur et la Vue (**Figure 27**). Il a pour rôle de leur présenter les données dans un formalisme adapté à leur logique de fonctionnement, comme par exemple une représentation objet.

La différence entre les données du domaine et celles définissant l'interface est sémantique. En effet, les unes comme les autres pourront être gérées à l'aide du pattern que nous venons de présenter mais elles ne relèvent pas du même niveau d'abstraction. Alors que l'accès aux données du domaine sera réalisé au niveau d'une application de la PAM, l'accès aux données définissant les interfaces sera effectué par le gestionnaire d'applications. On peut affirmer que la description complète des applications sera stockée et gérée de la même manière.

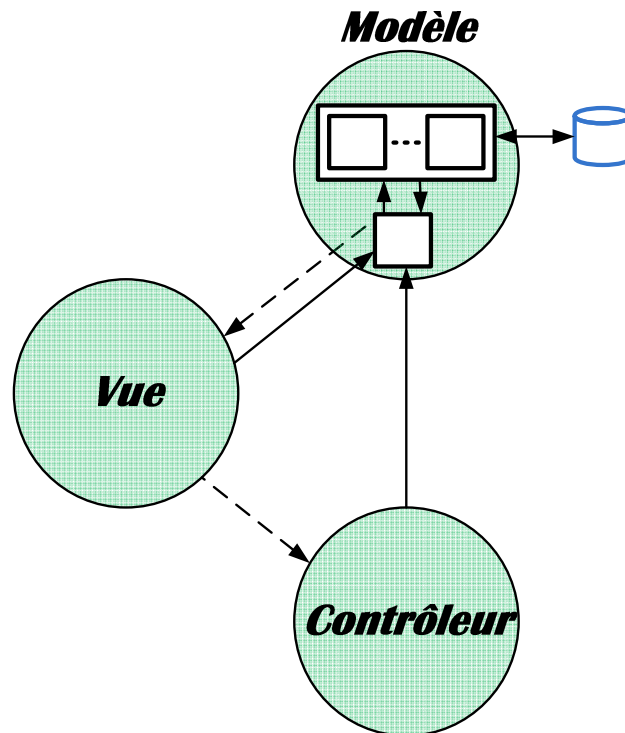


Figure 27 : Représentation d'agents de notre framework impliqués dans la gestion des données du domaine

C) La modélisation globale de notre solution

Dans la précédente section, nous avons décrit les éléments architecturaux de notre framework aux échelles macro, méso et micro. Dans la présente section, nous allons replacer dans le contexte de notre PAM les différents modèles d'analyse évoqués. Ainsi, l'ébauche du système précédemment exposée (voir chapitre **La division en sous-problèmes**) va pouvoir être enrichie et précisée. Commençons par rappeler brièvement les types de modèles d'analyses que nous avons pu mettre en avant :

- Des diagrammes de contexte et de principe (sans formalisme particulier).
- Des diagrammes UML et de ses extensions (aussi bien pour les SMA (Guedes & Vicari, 2009) que pour les IHM (Obrenović & Starčević, 2004)).
- Des maquettes/prototypes « off line » d'IHM (Beaudouin-Lafon & Mackay, 2007).

Nous pouvons aussi replacer ces choix de modèles d'analyse dans le contexte du méta-modèle de modélisation déjà évoqué (**Figure 10**). Effectivement, les types de modèles d'analyse retenus forment un méta-modèle d'analyse dont l'ensemble de nos spécifications formera une instance. Conformément à notre méta-modèle de modélisation, les relations entre ces modèles, de différents niveaux d'abstraction, et la possible implémentation de la PAM sont représentés sur la **Figure 28**.

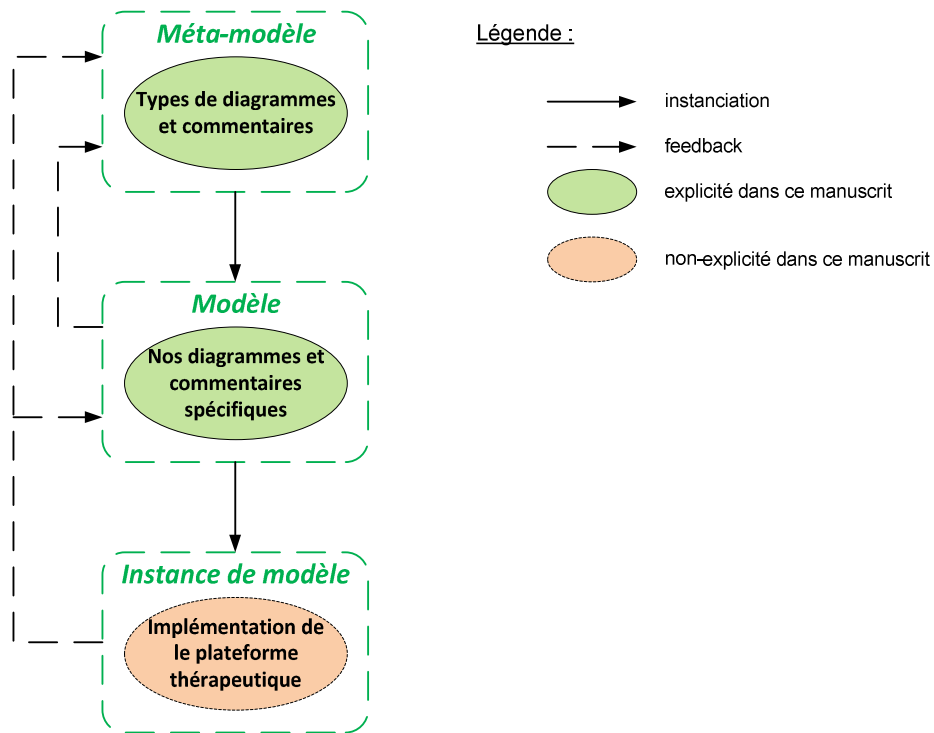


Figure 28 : Représentation de la relation entre le langage UML, nos diagrammes UML et notre PAM

Rappelons également que le but de notre PAM est de soutenir le travail coopératif assisté par ordinateur en fournissant un outil unique pour la construction, l'assistance et le suivi d'actes médicaux et/ou paramédicaux. L'approche que nous avons adoptée vise à faire reposer cette PAM sur une plate-forme multi-agent, qui elle-même repose sur le framework MA que nous avons décrit dans ce chapitre (**Figure 29**). Aussi, nous signalerons qu'une plate-forme multi-agent est un ensemble d'outils, méthodologiques et logiciels, conçu pour aider à la construction de systèmes multi-agents (Ricordel & Demazeau, 2002). Comme nous l'annonçons avec notre objectif **MODEL1**, ces différents outils font l'objet d'une proposition de modélisation dans le cadre de ce travail :

- Pour le framework MA, consulter la section **L'architecture du système**.
- Pour la plate-forme MA, consulter cette même section ainsi que la présente section.
- Pour la PAM, consulter la partie **L'intégration horizontale : support des processus métiers collaboratifs**.

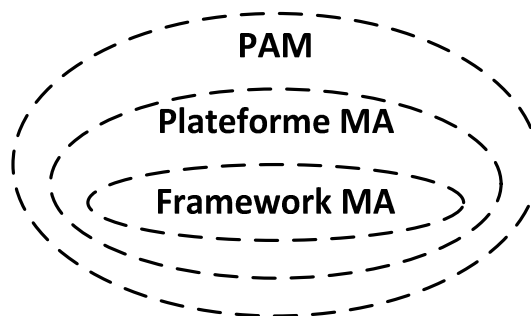


Figure 29 : Représentation de l'encapsulation des outils proposés

On pourra d'ailleurs maintenant renommer la PAM en **Plateforme d'Applications Médicales Multi-Agents**. Cette nouvelle appellation permet de rendre simultanément compte des aspects aussi bien fonctionnels que techniques de la plate-forme.

D'une manière très simplifiée, le but de notre système est de permettre à ses utilisateurs d'accéder à des données au travers d'IHM. A cette fin, nous avons décidé d'utiliser le design pattern MVC qui décrit une philosophie de développement permettant de découper « proprement » l'architecture logicielle (**Figure 30**). L'architecture peut, de la sorte, être aisément adaptée pour correspondre aux besoins spécifiques d'un framework ou d'un système particulier (comme par exemple dans (Bower & McGlashan, 2000; Potel, 1996; Sørensen & Mihailesc, 2010)). Nous ne nous sommes d'ailleurs pas restreints à l'utilisation des seuls concepts décrits dans le modèle original (c'est-à-dire (Burbeck, 1992)). Le but de ce design pattern est de clarifier les fonctions, au sens algorithmique du terme, des différents composants logiciels du système afin de les encapsuler et/ou hiérarchiser logiquement. En aucun cas MVC ne vise à complexifier ou rigidifier inutilement l'architecture logicielle, au contraire il définit un cadre structurant et est lui-même enclin à évoluer (Bragge, 2013).

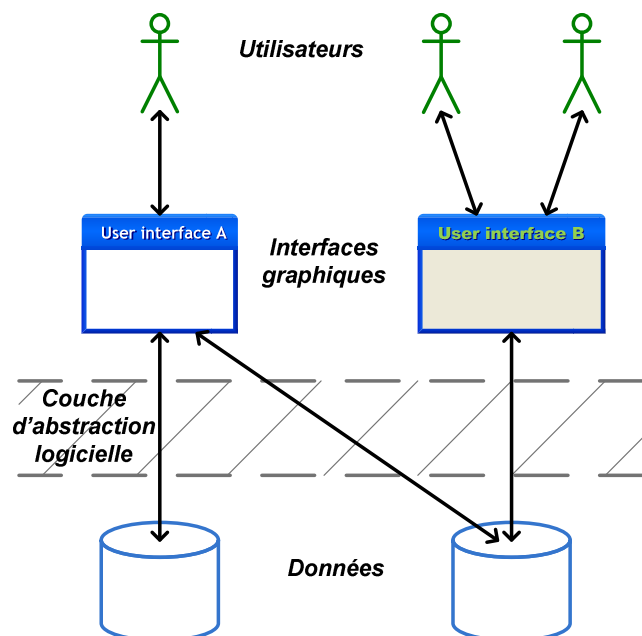


Figure 30 : Schéma d'ensemble d'un système supportant la modélisation MVC

En appliquant le modèle MVC à une modélisation MA, nous pouvons garantir la généricité et la réutilisation future de toutes les parties de l'outil (Gangat et al., 2012). Dans le cadre d'un SMA, ce design pattern va donc avoir pour avantage de pouvoir aussi bien être utilisé au niveau de la solution d'un point de vue global (Mahmoud & Maamar, 2006) qu'au niveau de ces briques constituantes que sont les agents (Gangat et al., 2012). Ceci signifie, comme nous l'avons présenté précédemment, qu'une approche MA peut être utilisée avec succès dans le Modèle, la Vue et le Contrôleur. Ainsi, afin d'offrir une vision plus détaillée du fonctionnement de notre système dans son ensemble, nous avons proposé des design patterns propres à être utilisés dans notre framework (voir la sous-section **La perspective du méta-système**). Ces patterns peuvent dès lors être appliqués à la vision simplifiée du système (**Figure 31**). Il est intéressant de noter que les organisations d'agents utilisées

intègrent à la fois les principes d'holarchie et de hiérarchie. En effet, il s'agit d'une bonne approche pour gérer avec les acteurs hautement interopérables du système, c'est-à-dire avec les utilisateurs et les sources des données. Dans notre cas spécifique, ceci est fondamental dans la mesure où notre système est censé interagir avec différents types d'utilisateurs et des sources de données hétérogènes. Le système est effectivement destiné à évoluer en fonction des besoins d'un vaste personnel médical pouvant déjà utiliser une grande variété de systèmes informatiques existants.

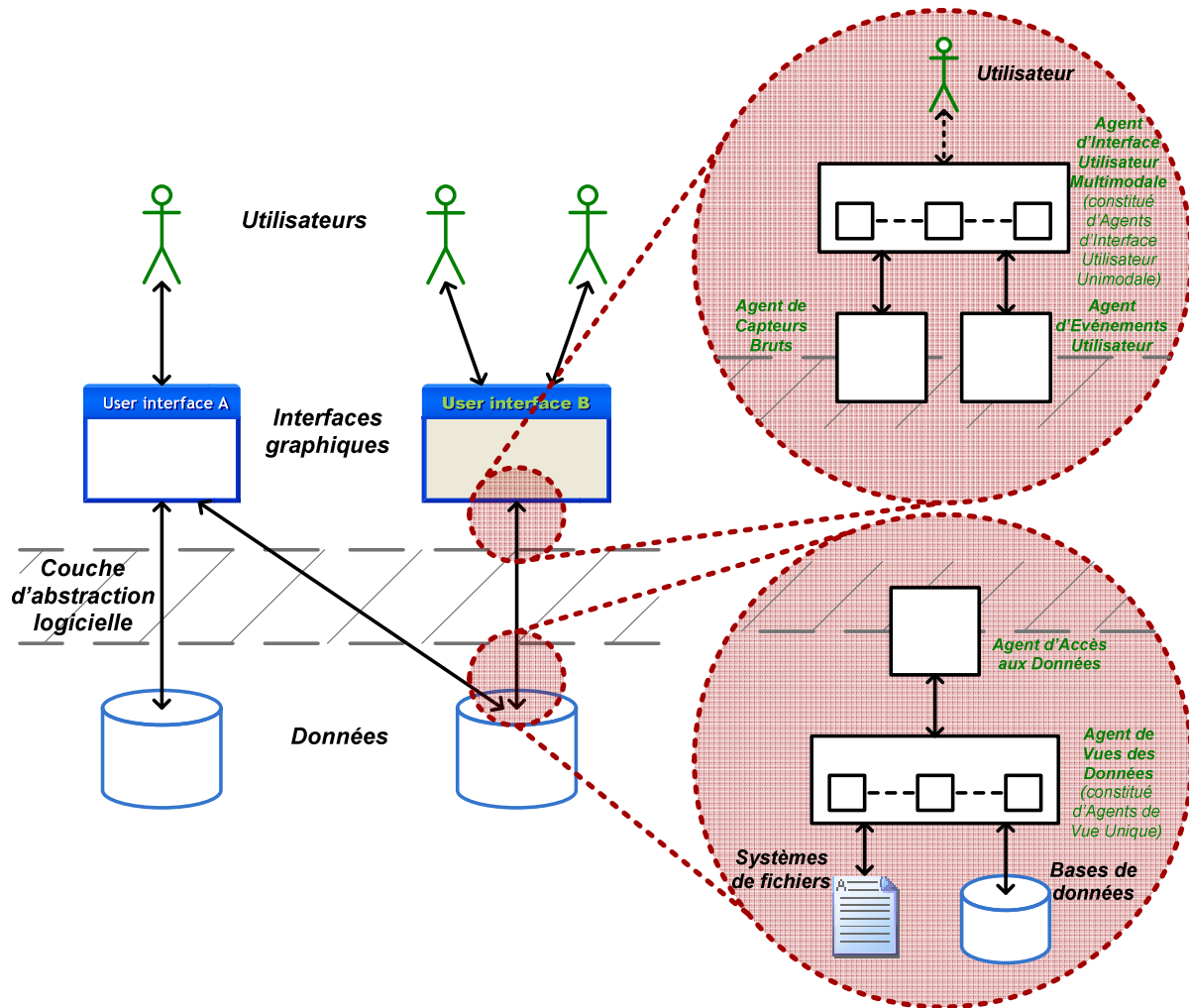


Figure 31 : Schéma d'ensemble d'un système multi-agents supportant la modélisation MVC

L'ébauche préliminaire de l'architecture de la PAMMA peut maintenant être précisée. A ce titre, la **Figure 32** et le **Tableau 5** décrivent les principaux agents de haut niveau impliqués dans le système. Nous pensons, à ce stade de notre recherche, que notre interface multimodale devrait, dans un premier temps, permettre l'acquisition de trois modalités perceptuelles : la voix, le regard et l'activité cérébrale. Ces modalités sont effectivement liées aux causes et/ou effets de la dyslexie (voir le chapitre **La dyslexie**). A ce titre, on fera remarquer que l'ébauche du système, que nous présentons ici (**Figure 32**), embarque des références à plusieurs technologies « sur étagère ». Ces dernières doivent permettre la gestion partielle (au niveau des médias et/ou des modes d'interactions) des modalités retenues. Ces technologies feront l'objet de plus amples descriptions ultérieurement dans ce manuscrit. De plus, nous ne prétendons pas que ces technologies sont les meilleures dans

leur domaine mais simplement qu'elles permettent efficacement et à moindre coût de procéder à la réalisation de preuves de concepts.

Nom de l'agent	Hiérarchie	Description
Agent Humain (AH)	0	Il interagit avec le système informatique composé d'un environnement matériel (AM) et logiciel (AL).
Agent Matériel (AM)	0	Il s'agit de l'interface entre les agents AH et AL . Il regroupe tous les capteurs et transducteurs utilisés par l' AL , dont certains sont destinées à être agrégés en interface multimodale.
Agent Logiciel (AL)	0	Il communique avec l' AM afin de soumettre l' AH à des stimuli, de traiter et d'analyser des données issues de ce dernier avant de procéder à des retours d'informations.
Agent Logiciel de Traitements (ALT)	1	Il contient les différents composants et threads de traitement. En particulier, les données multimodales sont traitées par un framework spécifique qui procède, entre autres, à la détection et au suivi des yeux et de la tête du patient.
Agent Logiciel de Données (ALD)	1	Il manipule différentes sortes de données utilisables par l' AL . Ces données peuvent être spécifiques à l' AH , à l' AL ou encore à l' AM .

Tableau 5 : Description des agents de haut niveau impliqués dans le système

Bien que cette ébauche ne le représente pas, n'oublions pas que notre système doit s'adresser à différents types d'utilisateurs. Ces derniers pourront d'ailleurs être amenés à l'utiliser collaborativement et simultanément. A ce titre, la PAMMA doit pouvoir supporter une interaction de type client-serveur. Le serveur doit, par exemple, permettre au praticien de piloter le dispositif dans son ensemble et d'analyser les résultats que ce dernier va fournir. Le client a, quant à lui, pour fonction d'être une interface logicielle, assurant affichage et recueil de mesures, entre le patient et le dispositif. Dans les faits, l'attribution des rôles de client et de serveur pourra varier selon la situation. En effet, chacune de ces parties distribuées du système doit pouvoir prendre l'initiative d'un échange tout comme satisfaire une demande provenant d'une autre partie du système : on parlera alors d'interaction « pair-à-pair ». Ce type d'interaction rentre évidemment complètement dans la philosophie de la communication entre agents dès lors qu'on assimile ces différentes parties du système à des agents. Nous pouvons également ajouter que ces parties du système, qu'on appellera simplement des agents, qui sont des applications ou des parties d'applications médicales, seront potentiellement exécutées par différentes instances de la PAMMA connectées au travers d'un réseau (**Figure 33**). Chacune de ces instances doit pouvoir si elle est contactée par une autre instance l'ajouter à son réseau. Pour maintenir l'état de son réseau d'applications et d'agents, chaque PAMMA doit disposer de services pour communiquer ainsi que pour tenir des pages blanches et des pages jaunes (comme, par exemple, dans la plate-forme MA nommée JADE (Bellifemine et al., 2003, 2008; Telecom Italia SpA, 2014)). Ces impératifs sont d'ailleurs les mêmes que ceux spécifiés par la FIPA (FIPA, 2002a, 2004) comme nous les avons déjà énoncé (voir la sous-sous-section **De la microarchitecture à la macro-architecture**).

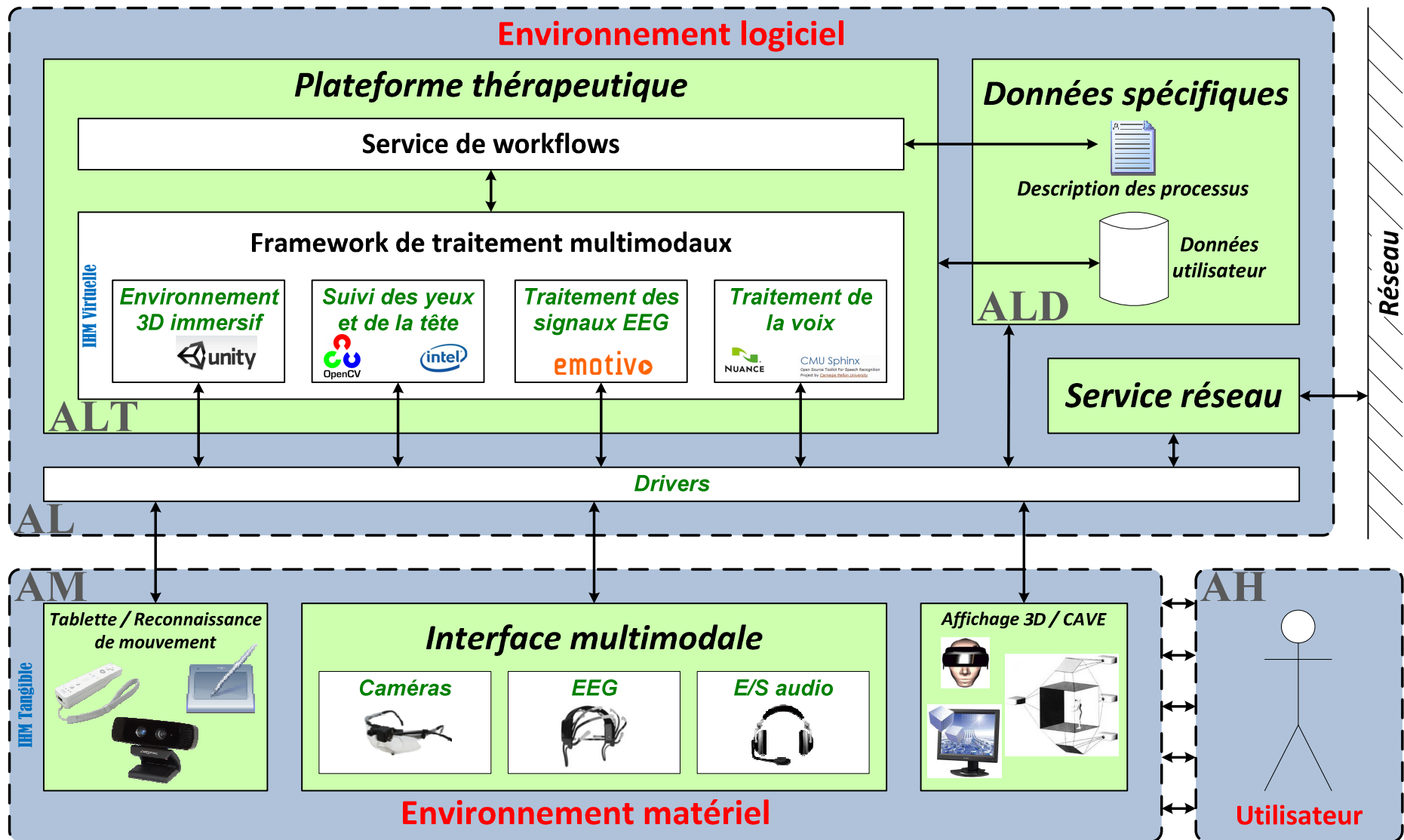


Figure 32 : Schéma global d'ébauche de l'architecture multi-agents du dispositif

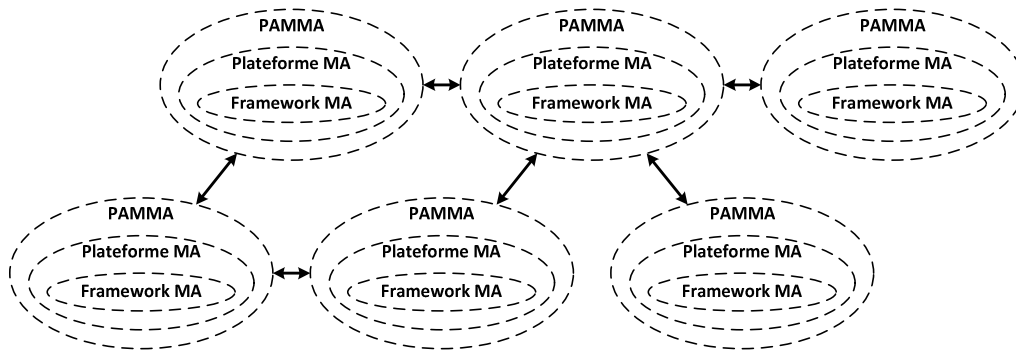


Figure 33 : Représentation de la constitution d'un réseau de PAMMA

Nous pouvons également préciser qu'un agent sert à représenter l'utilisateur. Du point de vue du système un tel agent signifie que l'utilisateur accède au système d'une manière protégée/sécurisée. A l'aide des modèles d'analyse retenus, nous n'avons qu'assez peu abordé cet aspect parce qu'il est fortement lié à des contraintes spatiales et temporelles propres à l'environnement de travail où le système va être utilisé. Nous pouvons cependant préciser qu'UML est adapté pour modéliser ce type de contraintes (Jürjens, 2002; Lodderstedt, Basin, & Doser, 2002; Maña, Montenegro, Sánchez, Ray, & Yagüe, 2003; Rodríguez, Fernández-Medina, & Piattini, 2006) et que les modèles, que nous allons proposer, pourraient être étendus en tenant compte de ces contraintes visant à sécuriser l'utilisation du système.

Nous avons maintenant une idée assez précise de ce que devrait être notre PAMMA et de son utilité concrète. Cette première modélisation de son architecture nous donne donc la matière pour instancier notre modèle de développement, c'est-à-dire commencer l'exécution de l'UP7. L'architecture du système est en effet un des principes fondateur de l'UP, qu'il faut prendre en considération le plus tôt possible (Gabay & Gabay, 2008). Nous nous appuyerons sur une méthode de développement orientée objet (Ramsin & Paige, 2008), mais il existe aussi différents modèles de développement adaptés au concept d'agents (Giorgini & Henderson-Sellers, 2005; Occello, 2003; Picard, 2004). Ces méthodes visent à définir des processus de développement de SMA, alors que nous cherchons à développer un framework de SMA (bien que lui-même un SMA). Néanmoins, on notera que les processus de développement impliqués dans ces méthodes suivent généralement un logique objet. A titre d'information, on pourra signaler non exhaustivement l'existence des méthodes suivantes (pour une présentation plus exhaustive de telles méthodes consulter (Noël, 2012)) :

- ADELFE (Bernon, Gleizes, Picard, & Glize, 2002; Picard & Gleizes, 2004; Picard, 2004) qui est dédiée aux systèmes MA auto-organiseurs par coopération. Cette méthode s'appuie sur RUP (Rational Software, 2001) et utilise UML ainsi qu'AUML (Bauer et al., 2001; Huget, 2004a).
- Gaia (Wooldridge, Jennings, & Kinny, 2000; Zambonelli et al., 2003) qui est une méthode générale et relativement complète. Elle permet la modélisation des agents et des organisations d'agents, cependant elle n'offre pas de représentation graphique riche ni n'envisage l'exécutabilité des modèles produits.
- INGENIAS/MESSAGE (Pavón & Gómez-Sanz, 2003) qui est une instance d'UP reposant sur une extension d'UML. La méthode s'accompagne de plusieurs outils permettant, entre autre, la représentation graphique du système et la génération de code.

- MaSE (DeLoach, Wood, & Sparkman, 2001) et O-MaSE (Garcia-Ojeda, DeLoach, Robby, Oyenon, & Valenzuela, 2008), qui se présentent comme des méthodes de spécialisation des objets en agents. On notera que la méthode O-MaSE met à contribution des diagramme UML et dispose d'une suite complète d'outils (Garcia-Ojeda, DeLoach, & Robby, 2009a, 2009b).
- MAS-CommonKADS (Iglesias, Garijo, González, & Velasco, 1998) est une méthode basée sur la conception d'intelligence artificielle issue de l'ingénierie des connaissances. Elle utilise, entre autres, les notations d'OOSE, de MSC et d'OMT, des ancêtres d'UML (Object Management Group, 2014e).
- PASSI (Cossentino & Potts, 2002; Cossentino, 2002) intègre à la fois des concepts orientés objet et de l'intelligence artificielle appliquée à la robotique. Cette méthode respecte les spécifications de la FIPA (IEEE Computer Society, 2014) et utilise fortement UML ainsi qu'AUML. Cette méthode s'accompagne d'outils proposés sous la forme d'un add-in pour Rational Rose (IBM, 2014).
- Prometheus (Padgham & Winikoff, 2003) est une méthode de développement adaptée à sa plate-forme d'exécution cible, JACK (AOS Group, 2008, 2014). Elle cible principalement le développement d'agents intelligents.
- Tropos (Bresciani, Perini, Giorgini, Giunchiglia, & Mylopoulos, 2004) est une méthode de développement relevant de l'ingénierie des besoins. Il s'agit d'une des rares méthodes appliquant une logique orientée agent pour sa propre définition. Elle s'appuie sur sa propre extension d'UML (Mylopoulos, Kolp, & Castro, 2001).
- Voyelles (da Silva & Demazeau, 2002; Ricordel & Demazeau, 2002) est une approche complète, bien que de haut niveau, dont les phases qui la composent ne sont pas réellement détaillées. Elle a donc besoin d'être couplée avec une autre méthode afin d'être pleinement applicable.

Nous n'allons pas ici comparer ces différentes méthodes (pour un comparatif de certaines d'entre elles consulter (Bernon et al., 2009; Isern, 2008; Picard, 2004)). On remarquera cependant que la plupart d'entre elles reprennent des activités similaires à celles de l'UP et qu'elles s'appuient sur des modèles d'analyses en UML (Odell et al., 2003; Picard, 2004), dans lesquels le concept d'agent est une extension de celui d'objet (Oja et al., 2001; Gerd Wagner, 2003b). Une perspective pourrait être d'appliquer certaines de ces méthodes, ou d'en extraire la substantifique moelle pour proposer notre propre processus de développement de SMA, pour la réalisation d'applications supportées par la PAMMA (voir la partie **L'analyse des résultats et les perspectives**). Pour conclure, on pourra signaler l'existence de plates-formes MA, dont nous avons déjà fait référence à certaines, telles que JADE (Bellifemine et al., 2003, 2008; Telecom Italia SpA, 2014), MASK (Occello, Baeijs, Koning, & Demazeau, 2003; Occello, 2003), MaDKit (Ferber & Gutknecht, 1998; Gutknecht, Ferber, & Michel, 2000, 2001; LIRMM, 2014), ZEUS (Nwana et al., 1999) ou encore JACK (AOS Group, 2008, 2014). Ces plates-formes permettent l'exécution de SMA, qui auront pu être préalablement modélisés à l'aide de méthodes de développement telles que celles que nous venons de citer.

Ayant grossièrement défini le contexte d'application de notre système, nous allons maintenant pouvoir démarrer l'application systématique de l'UP7. De cette manière, en gardant toujours la perspective de réaliser des applications médicales destinées à la dyslexie, nous affinerons la modélisation de la PAMMA en détaillant la structure et le fonctionnement de ses composants. Ceci est l'objet de la partie suivante de ce manuscrit.

Les développements

Jusqu'à maintenant, nous avons focalisé notre étude sur la problématique qu'est la prise en charge de la dyslexie, c'est-à-dire le contexte d'application de notre système, et identifié des besoins fonctionnels génériques en découlant. Nous avons ainsi pu intégrer ces besoins à une ébauche de modélisation de notre dispositif informatique et mettre en évidence un certain nombre de besoins techniques. Afin de mieux formaliser et documenter cette analyse, nous avons décidé d'appliquer la méthode UP7 (voir le chapitre **L'application de l'UP7: un modèle générique de prise en charge de la dyslexie**) et, de la sorte, valider notre objectif **MODEL1**. Ainsi, nous comptons produire la documentation nécessaire à la modélisation de ce système destiné au traitement de la dyslexie. Dans le cas de la réalisation de notre framework, le recouvrement entre documentation technique et documentation utilisateur (**Figure 34**) est très importante. Il n'est donc pas possible de s'affranchir de la production d'une documentation formelle de l'architecture du système, comme c'est néanmoins souvent le cas de nombreux projets informatiques (Neill & Laplante, 2003).

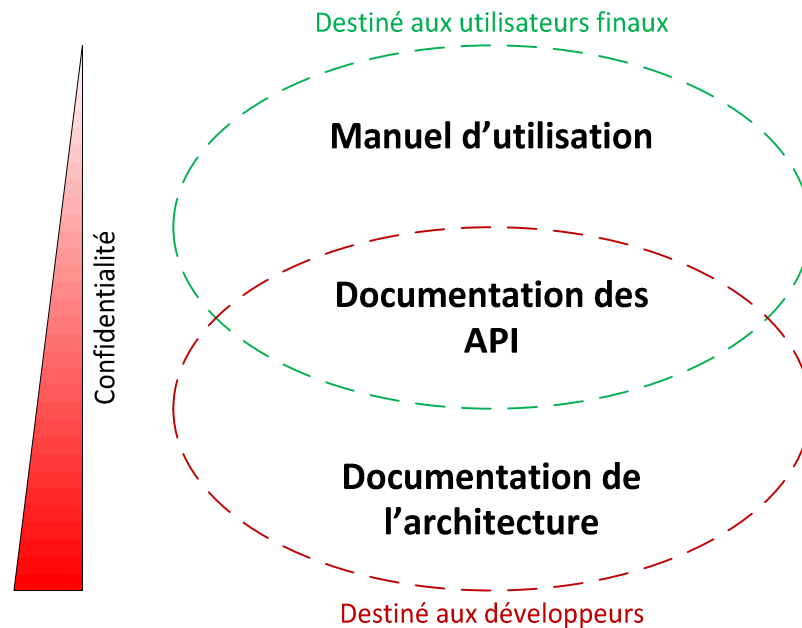


Figure 34 : Représentation des différents types de documentation accompagnant le logiciel

De ces spécifications du système, au travers de son application à la dyslexie, vont également émerger des besoins techniques permettant d'affiner l'architecture de notre PAMMA (**Figure 32**). Nous allons ainsi pouvoir fournir des éléments plus concrets liés aux facultés d'intégration horizontale que devra offrir notre plate-forme (voir le chapitre **L'intégration horizontale : support des processus métiers collaboratifs**). Nous pouvons d'ailleurs resituer notre modèle générique d'application dans le contexte du méta-modèle de modélisation déjà évoqué (**Figure 10**). Effectivement, cette modélisation appartient à un méta-modèle d'applications médicales dont une description spécifique à une application formera une instance. Conformément à notre méta-modèle de modélisation, les relations entre le framework de la PAMMA, les workflows et les applications médicales sont représentées sur la **Figure 35**.

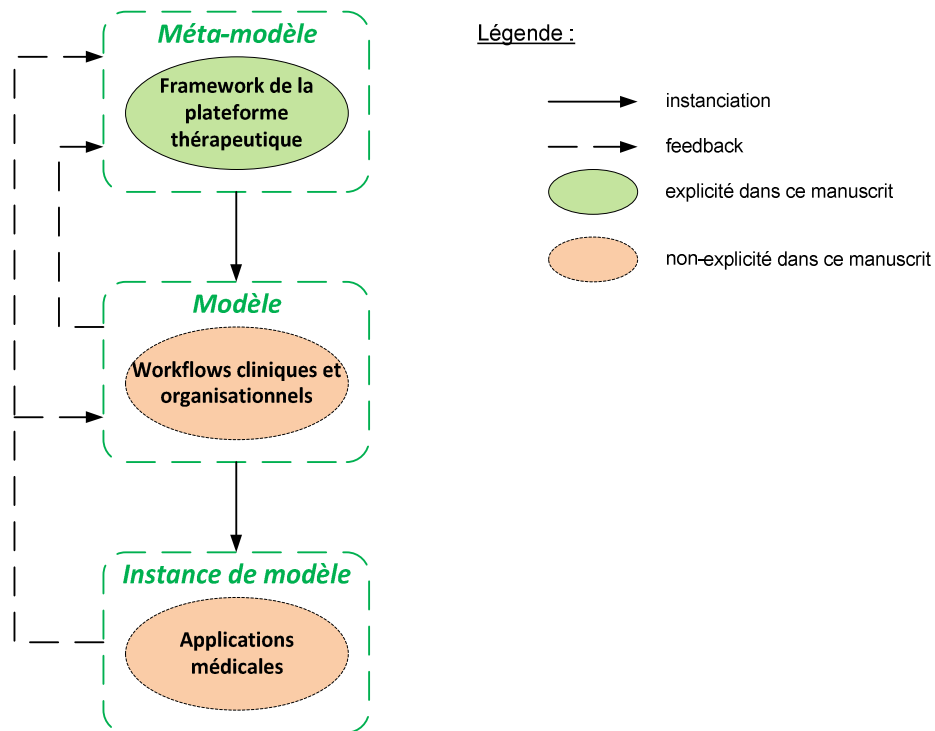


Figure 35 : Représentation de la relation entre le framework de la PAMMA, les workflows et les applications médicales

La modélisation seule n'étant pas suffisante pour éprouver la qualité de ces choix techniques, nous allons ensuite procéder à l'implémentation de plusieurs prototypes. La réalisation de ces prototypes correspond à nos objectifs **PROTO1**, **PROTO2** et **PROTO3**. L'implémentation de tels prototypes est indispensable pour démontrer la faisabilité de la solution que nous proposons (Arney et al., 2014). Les développements que nous allons conduire à cette occasion vont nous aider à apporter des réponses techniques précises à certaines de nos interrogations fonctionnelles les plus sensibles. Les deux premiers de ces prototypes concernent des modalités entrantes de notre système. A ce titre, nous ne pourrions pas débuter leur développement sans comprendre le contexte de l'acquisition et le traitement des données utilisateur issues de ces modalités. Effectivement, l'étude de telles modalités, qui vise à s'affranchir des moyens d'interactions classiques avec l'ordinateur, n'est pas triviale et fait l'objet de nombreuses recherches (par exemple, consulter (Sears & Jacko, 2007)). Dans un premier temps, nous allons donc étudier fonctionnellement les différentes modalités impliquées dans notre projet d'interface multimodale et ainsi préciser l'intérêt qu'a véritablement l'intégration de ces modalités à notre système dans le contexte des troubles neuropsychologiques. Ce sera d'ailleurs l'occasion de mieux comprendre comment notre système peut intégrer verticalement et horizontalement des processus métiers (pour une revue sur de telles méthodes d'intégration, consulter (Wangler & Paheerathan, 2000)). Ensuite nous étudierons comment gérer les aspects techniques liés à cette intégration. D'autre part, on signalera que le prototype visant à réaliser le suivi oculaire (dans le cadre de l'objectif **PROTO1**) va nous permettre également de définir un modèle de matériel (voir l'annexe **Le prototypage de l'interface multimodale**) adapté à l'usage envisagé pour notre système, conformément à notre objectif **MODEL2**.

I) L'application de l'UP7: un modèle générique de prise en charge de la dyslexie

Les résultats que nous allons maintenant présenter correspondent à ceux obtenus en fin de phase d'élaboration. A ce titre, on peut dire qu'il s'agit de la « milestone » relative à l'architecture du cycle de vie (Rational Software, 2001). La qualité de la modélisation de l'architecture ainsi dévoilée est prépondérante car elle conditionnera la qualité du futur logiciel (Jacobson, 2008a). Chaque activité que nous avons réalisée se compose de sous-activités pour lesquelles nous nous sommes inspirés des fiches guides proposées par Gabay et Gabay (2008). Ces fiches sont d'ailleurs reprises et adaptées pour chaque sous-activité traitée dans les parties qui vont suivre. On remarquera aussi que ces activités, bien qu'initialement dimensionnées pour la modélisation de logiciels orientés objet, mettront à profit une logique et des patterns d'agents (Juziuk et al., 2014). Néanmoins, nous ne décrirons pas formellement les workflows impliqués dans ce développement, nos travaux n'en requérant pas l'établissement strict, mais nous allons brièvement décrire les rôles et artefacts qu'ils impliquent. Les rôles impliqués dans ce travail de recherche, en dehors de nous-mêmes, sont ceux assurés par les praticiens et chercheurs que nous avons pu rencontrer tout au long de la période qu'a duré le projet de thèse. Ils nous ont principalement permis de répondre à nos différents questionnements fonctionnels et techniques. A ce titre, ils ont joué des rôles d'experts aussi bien dans les domaines métiers que dans d'autres plus techniques et relatifs au système informatique que nous voulions concevoir. Cela nous a permis d'identifier des besoins fonctionnels et non-fonctionnels du système. Un autre moyen pour recueillir les besoins des utilisateurs peut être de leur faire répondre à des questionnaires (Dieste, Juristo, & Shull, 2008). A ce titre, nous avons diffusé une enquête destinée aux parents d'enfants dyslexiques afin de mieux cerner leurs points de vue et attentes vis-à-vis d'un système tel que le nôtre (voir l'annexe **Enquête auprès des parents d'enfants dyslexiques**). Concernant les artefacts, s'agissant de modélisations en UML (Object Management Group, 2014e), ils seront présentés tout au long de cette partie. Ces réalisations constituent l'essence même de nos résultats et ils vont faire l'objet d'explications et représentations graphiques. Les activités, conduisant à la production de ces différents éléments statiques du processus, ont été effectuées de manière séquentielle comme indiqué sur la **Figure 36**.

Notre analyse, réalisée par une application d'UP7 adaptée à nos besoins par l'utilisation d'extensions dédiées à la modélisation d'IHM, d'agents et d'exigences en termes de sécurité, vise à capturer des besoins génériques relatifs à la prise en charge de la dyslexie. Il devient ainsi possible d'identifier des exigences relatives aux éléments que doit contenir la boîte à outils de la PAMMA, qui s'appuie sur notre framework MA précédemment exposé (**Figure 22**). De tels éléments doivent, par définition, être hautement réutilisables (comme, par exemple, cela est présenté dans (Adam, 2000)). On pourra également remarquer que l'architecture de notre PAMMA est conforme au découpage en trois couches proposé par Ferrara (1996). Ce découpage a pour but de garantir qu'un système à visée médicale soit suffisamment modulaire/flexible pour intégrer des systèmes existants tout en étant évolutif. D'autre part, nous pensons pouvoir déduire certains impératifs concernant le fonctionnement même de la PAMMA.

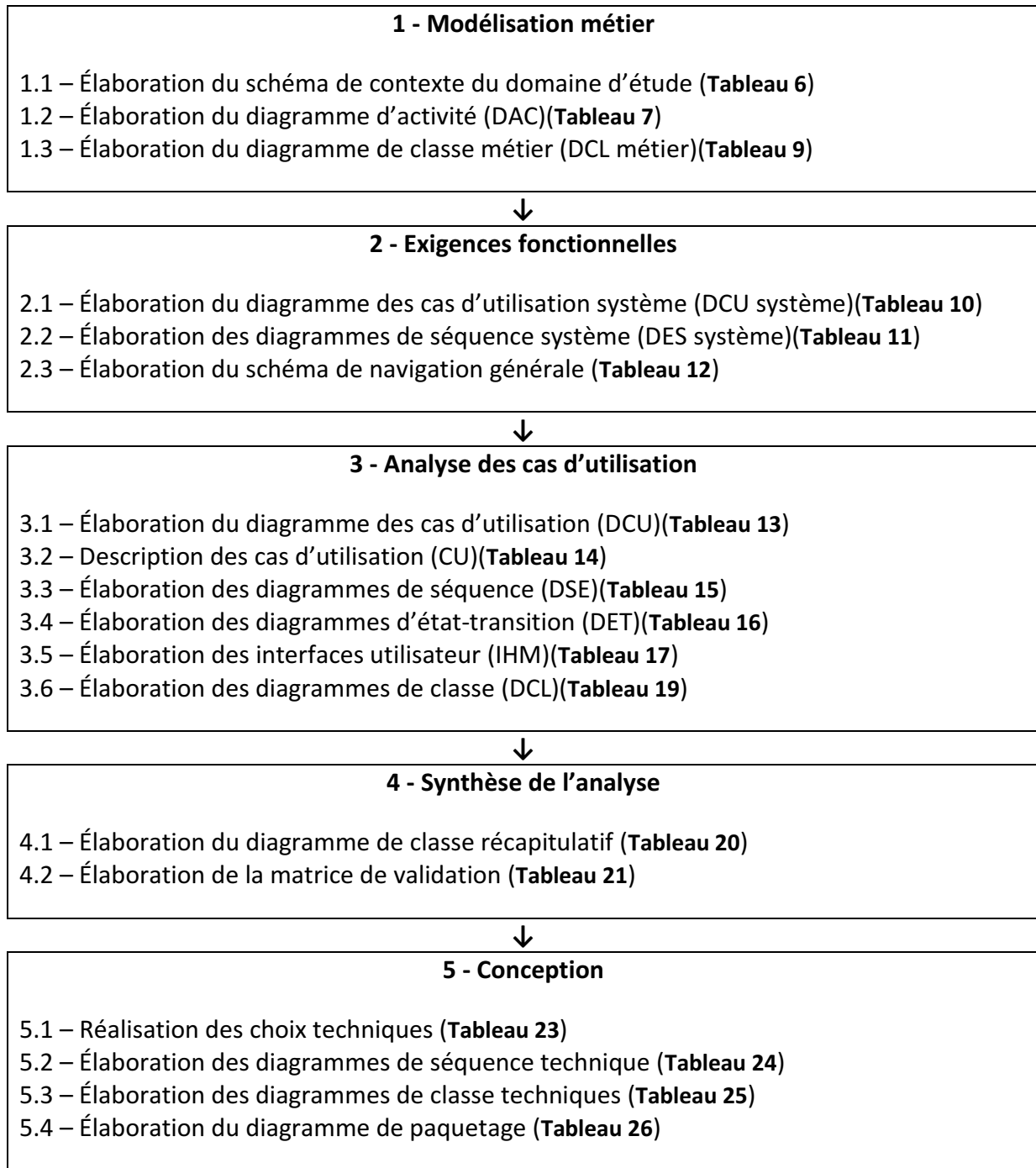


Figure 36 : Schéma de la séquence d'activités lors d'une phase d'UP7 (d'après (Gabay & Gabay, 2008))

Concernant notre analyse, on signalera que, pour des raisons de commodité et pour faciliter la compréhension, on assimilera la partie de la plate-forme utilisée par le professionnel de santé au serveur. Cependant, dans les faits, ce type d'utilisateur disposera d'un client, qui aura seulement plus de fonctionnalités que celui du patient. Les différents clients interagiront au sein d'un paradigme client-serveur étendu (Liang et al., 1994), et plus précisément, en mode « pair-à-pair ». Il est d'ailleurs intéressant de constater que, dans le cas d'un système client-serveur, le Modèle est typiquement une fonctionnalité du serveur, alors que la Vue se situe du côté client. Le Contrôleur, quant à lui, est traditionnellement

séparé en deux parties, une dans chacune des parties de l'application de manière à faire le lien entre elles. Mais il n'y a pas de restriction absolue en ce qui concerne la répartition des éléments du modèle MVC, ce qui permet de l'adapter à tout type d'applications distribuées (Potel, 1996). En ce point, le modèle MVC adopte une philosophie similaire à celle de Slinky et des modèles Arch (Bass et al., 1992). Pour en revenir au but premier de notre remarque, nous avons simplement choisi d'appeler « serveur » celui de ces clients qui a le plus de capacité d'un point de vue fonctionnel pour simplifier la notation et refléter l'asymétrie de l'interaction entre un praticien et un patient. Dans notre contexte, un client est une instance de la PAMMA consacrée à l'interaction avec l'utilisateur du système. Dans les sections suivantes, l'appellation « client » désignera un client destiné à être utilisé par un patient.

D'autre part, nous avons décidé de ne pas représenter les personnels du milieu de l'éducation parmi nos utilisateurs potentiels. Néanmoins, ces derniers pourront très bien disposer d'un client intégrant des éléments aussi bien de la partie concernant les patients, pour tout ce qui est lié à l'accès aux informations personnelles d'un enfant (comme, par exemple, dans (Mejía, Clara, et al., 2011)), que de celle des professionnels de santé, pour accéder aux connaissances relatives à la détection des troubles ou aux mesures d'aménagements possibles par exemple.

A) La modélisation métier

Cette première activité de la démarche vise à mieux connaître et comprendre les processus métiers des utilisateurs du futur système. Nous rappelons que nous souhaitons présenter ici une vision générique de la prise en charge des patients souffrant de dyslexie, afin de modéliser le côté fonctionnel du système. Nous verrons plus loin dans ce manuscrit comment une partie de ce travail de modélisation peut être utilisée dans l'implémentation systématique d'applications (voir la sous-sous-section **Le niveau 3 de l'éditeur**). Ainsi, nous allons définir un cadre à l'expression du besoin et ceci aboutira à trois résultats (Gabay & Gabay, 2008) :

- L'identification du système à étudier au sein de l'ensemble des processus de l'entreprise et la définition de son périmètre fonctionnel global.
- La définition des processus métiers destinés à être supportés par le système.
- La production de classes des concepts métiers, c'est-à-dire les informations créées, transformées ou manipulées dans le cadre de ces métiers.

Le but ici est de s'assurer que les utilisateurs finaux et les développeurs partagent une vision commune, dans un langage commun, du système à étudier (Rational Software, 2001). Ainsi, vont être produits des schémas de contexte ainsi que des diagrammes d'activité, des diagrammes d'environnement des cas d'utilisation et des diagrammes de classes, de sorte que le périmètre de ce système puisse être considéré comme défini à l'issue de cette activité. Dans la perspective de la conception de l'interface utilisateur, les différents profils d'utilisateurs seront explicités (Obrenović & Starčević, 2006). Par l'intermédiaire d'agents dédiés, ces profils devront :

- Garantir un accès sécurisé aux différentes fonctionnalités et ressources du système (Shakshuki, Luo, Gong, & Chen, 2004; Sulaiman, Sharma, Ma, & Tran, 2009).
- Offrir une personnalisation de l'expérience utilisateur (Boissier et al., 2004; Isern, 2008).

1) L'élaboration du schéma de contexte du domaine

FICHE GUIDE – FG1	
Activité 1 : Modélisation métier	
Sous-activité 1.1 : Elaboration du schéma de contexte du domaine d'étude	
Objectif	Positionner le système à étudier au sein des processus de soins
Point de départ	Esquisse fonctionnelle du besoin supposé
Point d'arrivée	Système à étudier positionné par rapport aux processus de soins et périmètre fonctionnel défini
Démarche d'élaboration	
1 - Identifier les processus connexes au système étudié.	
2 - Déterminer les interactions entre les processus connexes et le système étudié.	
3 - Préciser le périmètre du système à étudier (définition des sous-ensembles fonctionnels).	

Tableau 6 : Fiche Guide 1 pour UP7 (d'après (Gabay & Gabay, 2008))

Notre dispositif, permettant la prise en charge complète du patient, doit pouvoir se substituer aux outils préexistants ou, plus vraisemblablement, les intégrer en un tout cohérent (Madni & Sievers, 2014; Maier, 1998). Ainsi, nous identifions trois sous-ensembles fonctionnels mutuellement dépendants (**Figure 37**) :

- **Analyse et validation de théories**, qui permet de corréliser un ou plusieurs déficits à des analyses des observations comportementales et/ou électrophysiologiques cérébrales. En particulier, ce sous-ensemble devra disposer d'un éditeur qui, conformément à notre découpage en sous-problèmes (voir le chapitre **La division en sous-problèmes**), devra permettre de :
 - Définir les processus de diagnostic et de remédiation, mais aussi au niveau plus général des processus de travail inter et intra-organisationnel.
 - Définir précisément les tests/épreuves/examens.
- **Aide diagnostique**, qui permet de diagnostiquer exactement le type de pathologie du patient.
- **Environnement dynamique de remédiation**, qui permet d'effectuer les soins en fonction du diagnostic préétabli puis d'assurer le suivi.

Pour répondre aux besoins fonctionnels ainsi exprimés, nous devons définir certaines contraintes concernant la structure du projet informatique. Premièrement, il faudra distinguer deux plates-formes logicielles (Mandellos et al., 2009)(**Figure 38**) :

- **Un serveur** permettant au professionnel de santé de piloter toutes les fonctionnalités du dispositif. Cette plate-forme doit permettre d'éditer des workflows et de contrôler leur exécution.
- **Un client** permettant d'effectuer les relevés amenant au diagnostic ainsi qu'à prodiguer des soins au patient ou réaliser un monitoring. Un simple client web ne peut donc pas faire l'affaire (Magrabi et al., 2001).

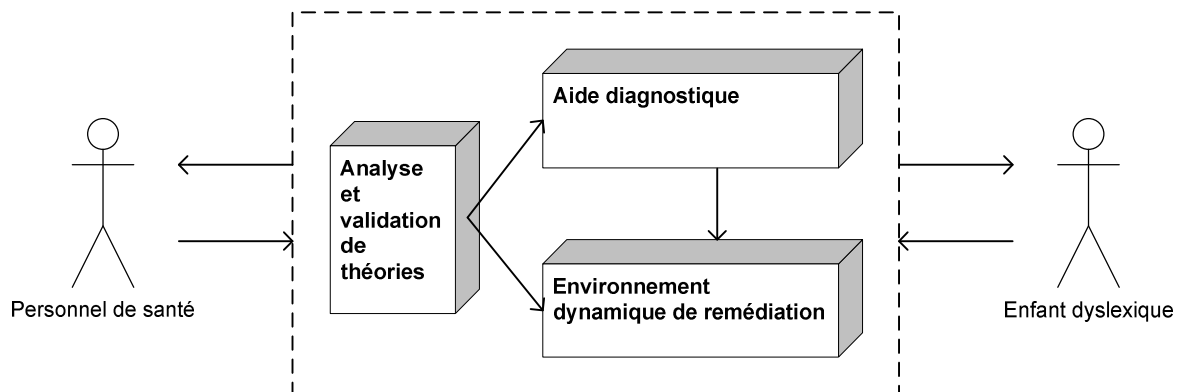


Figure 37 : Schéma de contexte fonctionnel

Cette structure distribuée doit permettre premièrement d'utiliser le système dans un contexte de télémedecine (Wootton, 2001), pour lequel les SMA peuvent être utilisés efficacement (Jørgensen, Hallenborg, & Demazeau, 2014; Laleci et al., 2008). Deuxièmement, cette séparation permet les deux modes d'utilisation distincts (comme ceci est le cas, par exemple, dans (Jovanov et al., 1999; Kilman & Forslund, 1997)) suivants :

- **Online**, pour avoir un contrôle en direct sur les stimuli envoyés au patient ainsi que sur les réponses qu'il va y donner et/ou pour travailler collaborativement avec d'autres praticiens. Les interactions se font de manière synchrone pour permettre une utilisation en temps réel.
- **Offline**, pour permettre une utilisation autonome de chacune de ces deux composantes logicielles. Les interactions se font de manière asynchrone pour permettre de stocker et de transmettre des données.

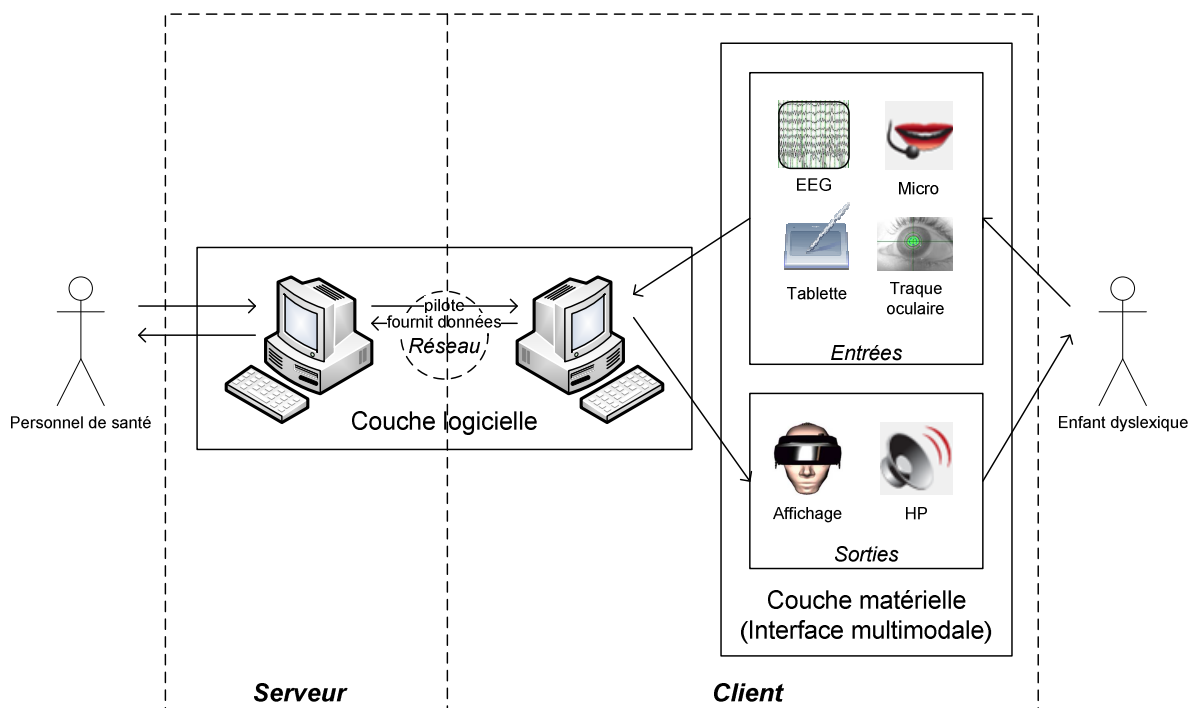


Figure 38 : Schéma de contexte structurel

L'étude de l'ensemble de ces contextes d'utilisation nous permet d'établir les différents environnements dans lesquels le système va être amené à être utilisé (Obrenović & Starčević, 2006). Nous avons représenté ces environnements, dans lesquels prennent place les cas d'utilisation du système, sur la **Figure 39**. Dans ces environnements vont s'inscrire les réalisations de processus métiers.

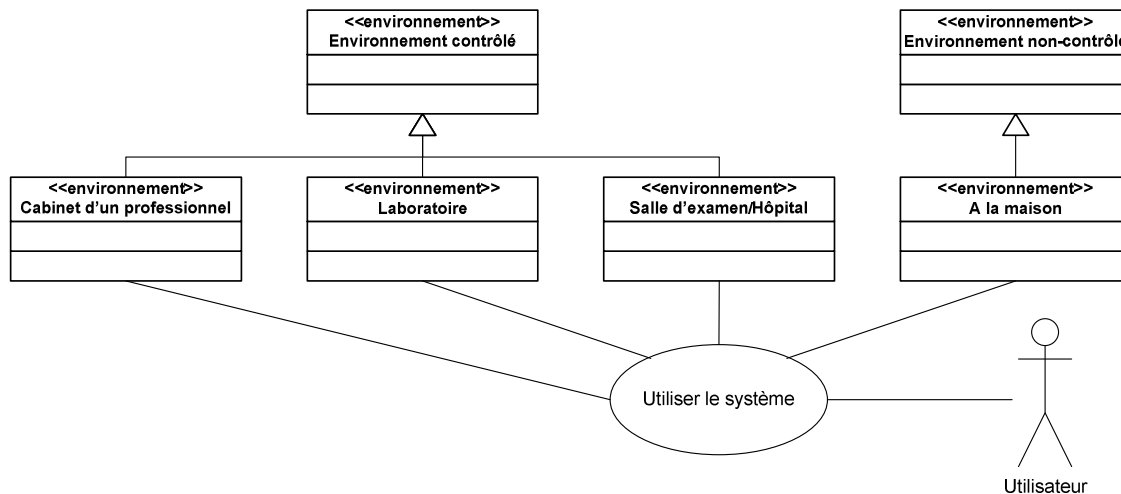


Figure 39 : Diagramme d'environnement des cas d'utilisation

Au sujet des environnements contrôlés présentés, nous pourrions, par exemple, signaler que :

- Le cabinet du professionnel correspond à l'environnement d'utilisation d'un clinicien exerçant en milieu libéral.
- Le laboratoire est typiquement l'environnement dans lequel le chercheur va travailler.
- La salle d'examen va cadrer avec l'environnement particulièrement contrôlé dans lequel certaines spécialités médicales (ou autres) peuvent seulement être pratiquées.

Les environnements non-contrôlés sont les endroits dans lesquels, par exemple, le patient peut se trouver lorsqu'il utilise le système dans un contexte de télé-médecine. Comme nous l'avons déjà évoqué précédemment et comme nous le préciserons un peu plus loin dans ce chapitre, il existe différents types d'utilisateurs qui seront amenés à utiliser notre système. On rappellera que le formalisme MA est particulièrement approprié pour ce genre d'applications où les utilisateurs et les lieux d'utilisation sont nombreux, divers et distribués, mais nécessitent de communiquer entre eux (Nealon & Moreno, 2002, 2003).

Pour finir avec notre présentation du contexte, nous souhaitons évoquer l'importance potentielle que peut avoir le domaine de la simulation. Effectivement, des modèles de simulations, mimant le comportement d'individus dyslexiques, existent et donnent des résultats similaires à ceux obtenus avec de réels être humains (Ziegler, 2011). Cependant, il faut se méfier car des modèles tels que le DRC (Coltheart et al., 2001) ou le CDP+ (Perry et al., 2007) ne tiennent pas compte de la dynamique, du développement, de l'enfant. Dans le but de quantifier les progrès des enfants dyslexiques, il serait donc pertinent que notre système propose un modèle de simulation capable de s'affranchir de ces limitations.

2) L'élaboration du diagramme d'activité

FICHE GUIDE – FG2	
Activité 1 : Modélisation métier	
Sous-activité 1.2 : Elaboration du diagramme d'activité (DAC)	
Objectif	Décrire les processus métiers du système à étudier
Point de départ	Système à étudier positionné par rapport aux processus de soins et périmètre fonctionnel défini
Point d'arrivée	Acteurs identifiés et processus métiers du système étudié définis (flot de contrôle, flot de données...) dans le DAC
Démarche d'élaboration	
<p>1 - Identifier les acteurs internes et externes du système étudié.</p> <p>2 - Identifier des actions du processus.</p> <p>3 - Définir le flot de contrôle (enchaînement des actions) :</p> <ul style="list-style-type: none"> - transitions automatiques - transitions gardées - synchronisation - début/fin du flot <p>4 - Représenter les données liées aux actions. Ces données sont décrites à l'aide des concepts.</p> <p>5 - Déterminer le flot de données c'est-à-dire l'enchaînement des données entre elles et/ou avec des actions.</p> <p>6 - Décrire les rôles des acteurs du système.</p>	

Tableau 7 : Fiche Guide 2 pour UP7 (d'après (Gabay & Gabay, 2008))

Conformément à notre schéma de contexte fonctionnel (**Figure 37**), nous allons réaliser trois DAC, un pour chaque sous-système fonctionnel précédemment identifié. Nous allons ainsi mettre en évidence les acteurs et les flux pour les sous-ensembles ci-dessous :

- **Analyse et validation de théories**, implémentation des scénarii de diagnostic et remédiation de patients souffrant d'un Trouble Spécifique du Langage.
- **Aide diagnostique**, utilisation des scénarii de diagnostic.
- **Environnement dynamique de remédiation**, utilisation des stratégies de remédiation.

Notre analyse s'appliquera à proposer des processus relativement génériques qui pourront, par la suite, être partagés et adaptés aux exigences spécifiques de certaines pratiques et/ou institutions (Boxwala et al., 2001; Hripcsak, Ludemann, Pryor, Wigertz, & Clayton, 1994). On pourra remarquer qu'on retrouve ce genre de modélisation générique d'un processus clinique, par exemple, avec le modèle Domino (Bury, Fox, & Sutton, 2001; Das, Fox, Elsdon, & Hammond, 1997; Fox et al., 1997) ou encore dans des produits de Shared Medical Systems

(Anderson et al., 2001)). Notre modélisation est d'ailleurs similaire à celle proposée par Sayyad Shirabad et al. (2012) dans le cadre de leur système d'aide à la décision, néanmoins la nôtre propose un niveau de détails plus fin. Nos processus génériques peuvent être appliqués en réalité à d'autres pathologies que la dyslexie, et même à d'autres domaines que celui des TSL. La particularité des pratiques autour de la dyslexie est qu'il s'agit de soins ambulatoires pour lesquels l'utilisation de certains matériels spécifiques intervient.

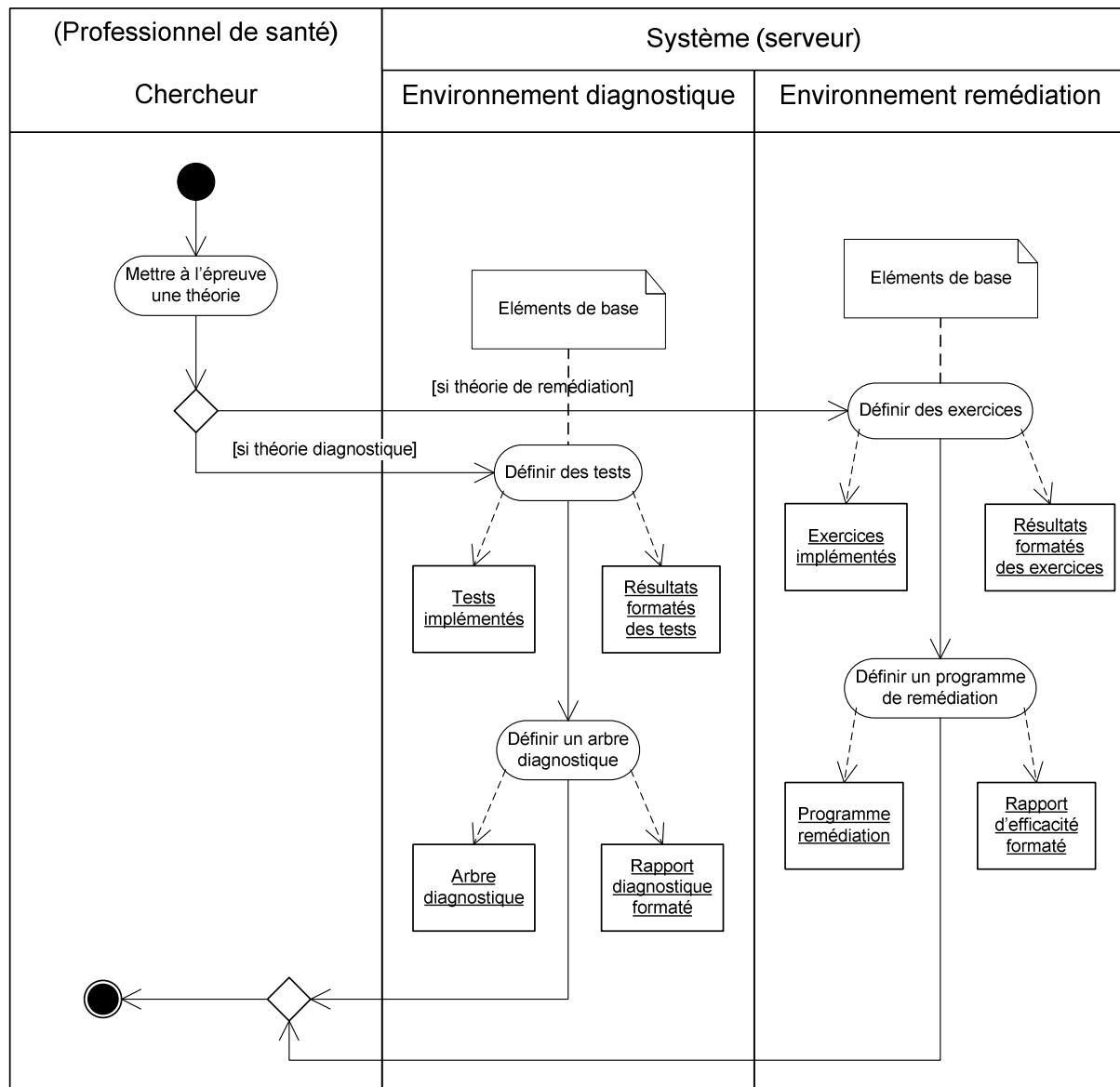


Figure 40 : Diagramme d'activité de la partie d'analyse et validation de théorie

Les acteurs identifiés pour ce sous-système fonctionnel sont (**Figure 40**) :

- **Professionnel de santé** – Il va créer du contenu sous forme d'épreuves (tests ou exercices) qu'il agrégera en processus de soins (arbres diagnostiques ou programmes de remédiation).
- **Environnement diagnostique (serveur)** – Il s'agit de la partie du système permettant le diagnostic du patient.
- **Environnement remédiation (serveur)** – Il s'agit de la partie du système permettant de mettre en place et suivre une stratégie de remédiation.

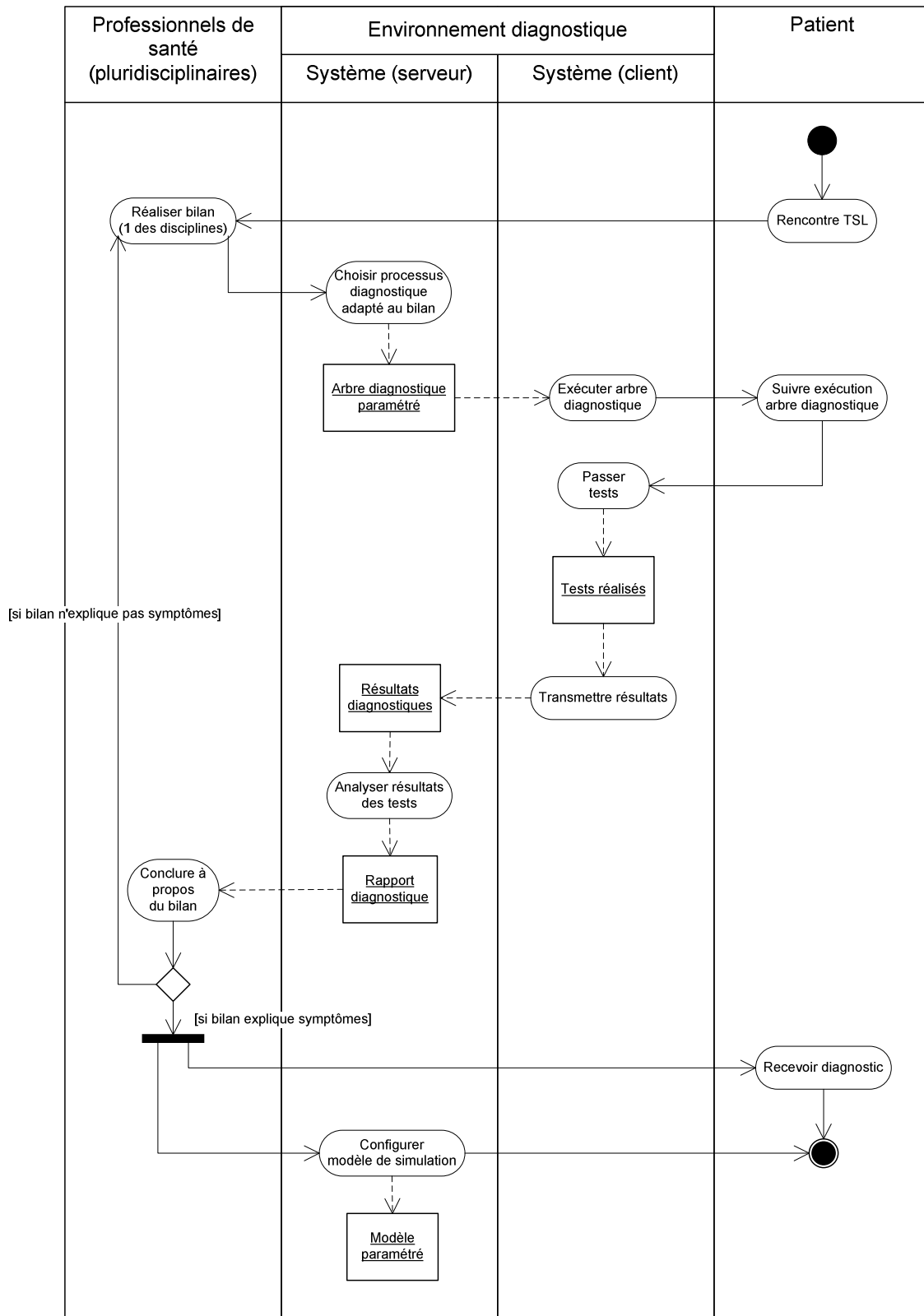


Figure 41 : Diagramme d'activité du sous-système d'aide diagnostique

On peut remarquer que les tests et exercices seront définis par agrégations d'éléments de base. La version du produit qui sera distribuée devra d'ailleurs disposer initialement d'un certain nombre d'épreuves et processus de soins pré-implémentés. D'autre part, on signalera que les arbres diagnostiques sont des arbres décisionnels tels que fréquemment

employés pour proposer de l'aide à la décision en contexte clinique (Hadorn, 1995)(pour un exemple d'utilisation, consulter (Sayyad Shirabad et al., 2012)). Chacun de ces arbres est en réalité une application destinée à être exécutée par la PAMMA, comme le sont également les programmes de remédiation. On remarquera également que ces applications peuvent être représentées à l'aide de diagrammes d'activité ou tout autre formalisme permettant la représentation de workflows (voir la sous-sous-section **Le niveau 3 de l'éditeur**).

Ce sous-système d'aide diagnostique comporte les acteurs suivants (**Figure 41**) :

- **Professionnels de santé** – Les membres de l'équipe pluridisciplinaire établissant le diagnostic final.
- **Environnement diagnostique (serveur et client)** – La partie du système permettant la sélection et l'exécution des différents arbres diagnostiques associés à chaque bilan. Ces bilans permettent ainsi de soumettre le patient à des batteries de tests.
- **Patient** – Ici il s'agit de l'enfant dyslexique.

On signalera que Hammermüller et Miksch (1999) décrivent des étapes similaires de spécification, de planification, d'exécution et d'analyse dans leur système d'aide à la décision destiné au milieu médicale.

Les acteurs impliqués dans ce sous-système destiné à la remédiation sont (**Figure 42**) :

- **Professionnel de santé** – Ici il va piloter la remédiation en choisissant un programme adapté à la pathologie du patient. Ce pilotage peut se faire en direct (utilisation online) ou de manière différée (utilisation offline). On pourra préciser que ce pilotage correspond principalement à la journalisation des informations et processus entourant le patient ainsi qu'au monitoring des processus en cours d'exécution. D'autre part, les programmes de remédiation que le serveur envoie au client sont de véritables applications médicales. On peut même noter qu'il s'agit d'agents mobiles qui seront exécutés par la plate-forme qu'est le client plutôt que sur le serveur (Mahmoud & Yu, 2006).
- **Environnement remédiation (serveur et client)** – Il s'agit de la partie du système permettant le choix et la mise en place d'un programme de remédiation côté serveur, puis son exécution côté client tout en permettant un suivi de l'évolution de la remédiation par le serveur.
- **Patient** – Dans notre périmètre d'étude, il s'agit de l'enfant souffrant de dyslexie développementale.

On peut noter que les pointillés séparent le diagramme en sections. Ils représentent soit un passage du mode online à offline (si le serveur et le client sont impliqués tous les deux dans la section précédente) soit l'inverse, à savoir un passage du mode offline à online (si seulement le serveur ou le client ont été impliqués dans la section précédente).

Pour des raisons évidentes de sécurité, liées au caractère confidentiel des données manipulées, il convient de mettre en place (Anderson, 1996; Forslund et al., 1996; Vasserman & Hatcliff, 2014) :

- Un système d'authentification, garantissant l'identité des personnes se connectant au système.
- Des protocoles de communication sécurisés, permettant de faire transiter les données par un réseau de façon sûre.

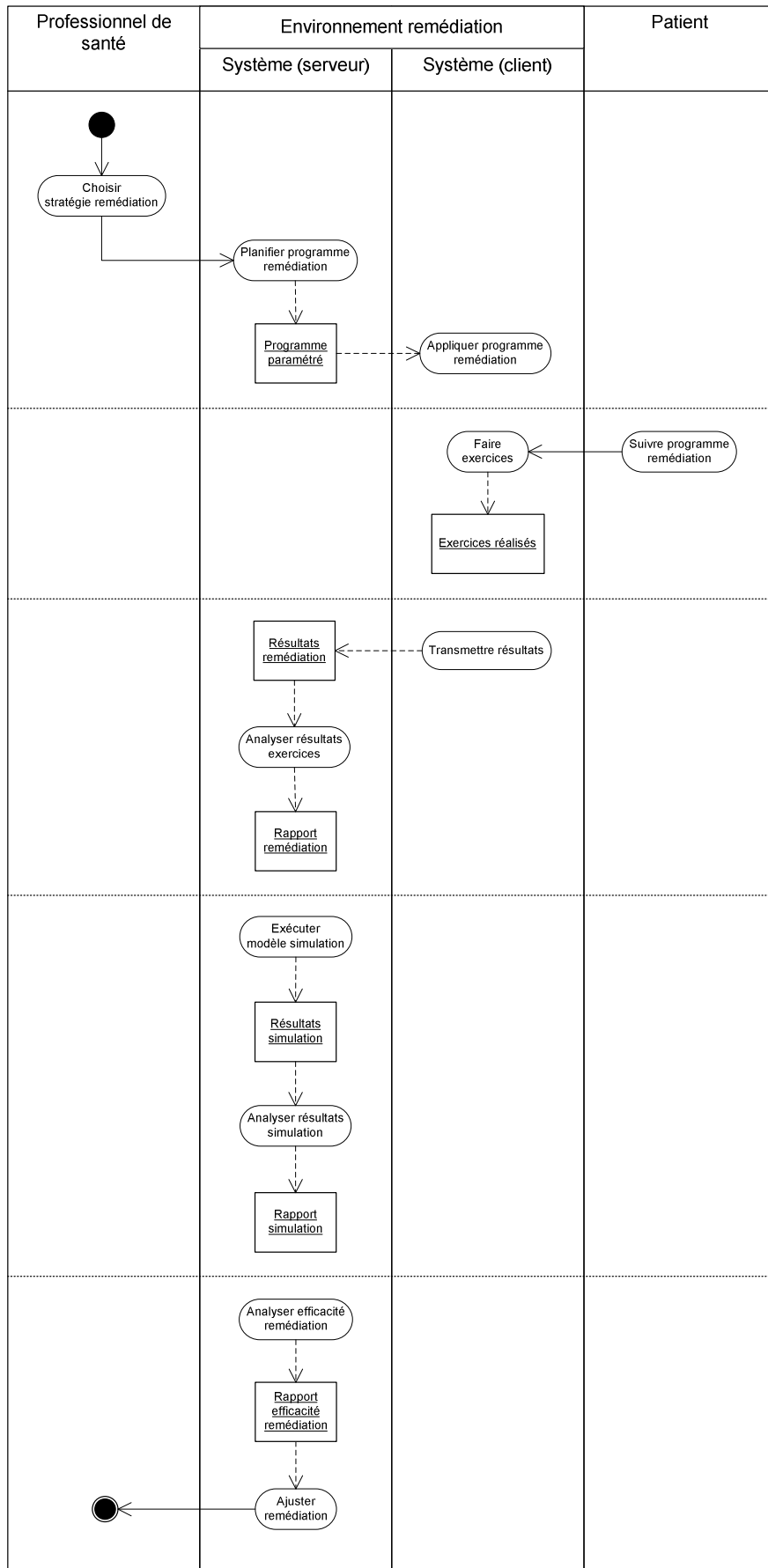


Figure 42 : Diagramme d'activité dans l'environnement dynamique de remédiation

Types de profils	Valeurs étiquetées
Professionnel de santé	Droit d'accès aux données du patient, Domaines de compétences
Personnel médical	Droit d'accès en exécution aux processus médicaux
Personnel paramédical	Droit d'accès en exécution aux processus paramédicaux
Clinicien	Niveau de compétences, Niveau hiérarchique
Chercheur	Droit d'accès aux différents niveaux du gestionnaire de processus
Patient	Date de naissance, Sexe, Pathologie(s)

Tableau 8: Liste de valeurs importantes des profils d'utilisateurs

Pour mettre en place un système d'authentification, il faut pouvoir identifier les utilisateurs et attribuer des droits propres à chacun de ces utilisateurs ou à chaque groupe d'utilisateurs. C'est pourquoi, nous allons maintenant présenter les différents acteurs du système au travers de leurs profils d'utilisateurs (Obrenović & Starčević, 2006). Des attributs, parmi les plus importants, pour chacun de ces différents profils sont résumés dans le **Tableau 8**. Ces valeurs ont pour but de restreindre l'utilisation que chaque type d'utilisateur va pouvoir faire du système dans le but de ne lui proposer que des fonctionnalités le concernant. On rappellera qu'il existe des extensions d'UML pour représenter les aspects liés à la sécurité (Rodríguez et al., 2006). Dans notre cas, nous nous contenterons de décrire le déroulement de l'authentification des utilisateurs puis nous signalerons les utilisateurs authentifiés à l'aide d'un cadenas (**Figure 43 (b)**).



Figure 43 : Représentation **(a)** d'un utilisateur et **(b)** d'un utilisateur authentifié

D'autre part, ces profils devraient contenir des informations permettant une personnalisation du système adaptée à chaque différent type d'utilisateur ainsi qu'aux conditions dans lesquelles ils l'utilisent (Kaschek et al., 2006). Une description des différents types de profils est présentée au travers d'un diagramme de classe sur la **Figure 44**. Sur ce diagramme, nous avons présenté un certain nombre de spécialisations possibles pour les différents profils d'utilisateurs relevant des domaines médicaux et paramédicaux impliqués dans le traitement de la dyslexie. Cette énumération est là afin d'illustrer notre propos et ne se veut en aucun cas être exhaustive. En outre, on signalera que de tels profils vont pouvoir être étoffés dynamiquement au cours de l'emploi du système par l'utilisateur de façon à pouvoir lui procurer une expérience personnalisée (Cesta & D'Aloisi, 1996; Lieberman & Selker, 2003). En effet, nous avons évoqué, dans notre première partie (voir la sous-sous-section **Les aménagements et la compensation en milieu scolaire** et la sous-section **Impliquer le patient dans une remédiation centrée autour de lui**), qu'il était important de pouvoir réaliser un profilage intelligent des utilisateurs. La modélisation MA répond positivement à ce besoin en proposant des agents chargés d'offrir une personnalisation de

l'expérience d'utilisation adaptée à chaque utilisateur (Boissier et al., 2004; Lieberman & Selker, 2003; Rodríguez, Favela, Preciado, & Vizcaíno, 2005). Cette gestion des profils doit permettre d'affranchir les utilisateurs de tâches de bas niveau (Cesta & D'Aloisi, 1996). On signalera également que des agents en charge de cette personnalisation sont « programmés par l'exemple », c'est-à-dire que leur apprentissage est un effet de bord de l'utilisation continue du système par ses utilisateurs (par exemple, consulter (Mitrović & Mena, 2003)). Cependant, nous nous ne détaillerons pas ce versant de la gestion des profils qui nous semble hors de notre périmètre d'étude.

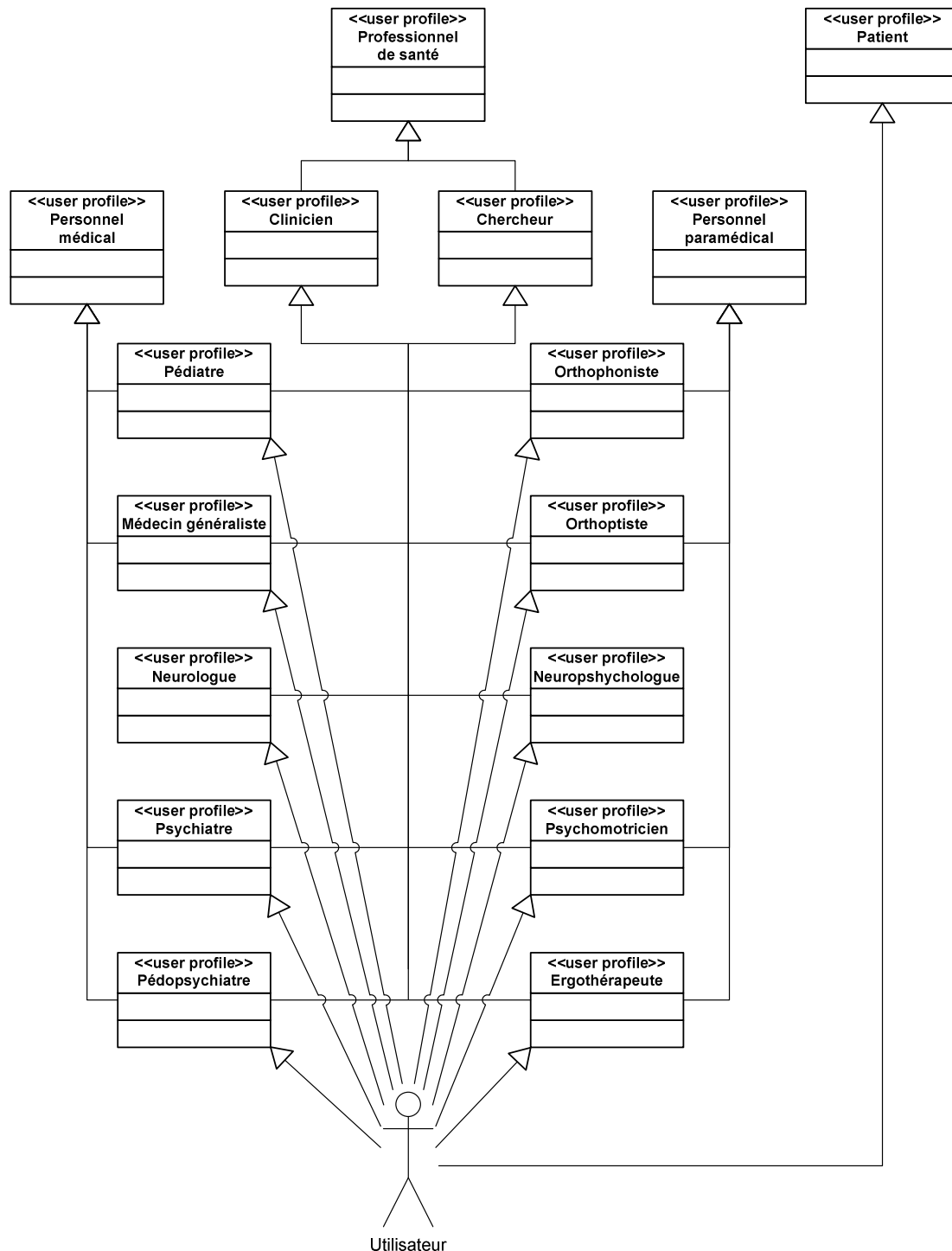


Figure 44 : Diagramme de classe des profils d'utilisateurs

3) L'élaboration du diagramme de classe métier

FICHE GUIDE – FG3	
Activité 1 : Modélisation métier	
Sous-activité 1.3 : Elaboration du diagramme de classe métier (DCL)	
Objectif	Définir les concepts métiers du domaine sous forme de classe
Point de départ	Acteurs identifiés et processus métiers du système étudié définis (flot de contrôle, flot de données)
Point d'arrivée	Concepts métiers identifiés définis dans le DCL métier et le glossaire métier
Démarche d'élaboration	
<p>1 - Identifier les concepts du domaine sous forme de classes en prenant comme base ceux définis dans le diagramme d'activité.</p> <p>2 - Préciser les principaux attributs utiles à la compréhension des experts métiers.</p> <p>3 - Déterminer les relations entre les classes :</p> <ul style="list-style-type: none"> - nom de l'association - multiplicité <p>4 - Décrire de manière générale les concepts du domaine afin d'obtenir un glossaire métier.</p>	

Tableau 9 : Fiche Guide 3 pour UP7 (d'après (Gabay & Gabay, 2008))

Comme nous pouvons diviser le dispositif en deux parties, serveur et client (**Figure 38**), nous allons proposer un DCL pour chacune d'elles. Ces classes du domaine métier seront utilisées pour l'accomplissement des cas d'utilisation (Khamis & Abdelmonem, 2002). Côté serveur, nous avons ainsi identifié les concepts du domaine suivants (**Figure 45**) :

- **Utilisateur (Professionnel de santé, Patient, Modèle Simulation)** – Ce concept englobe tous les acteurs utilisant l'application. On sépare ces acteurs en deux sous-classes, selon qu'ils sont professionnels de santé ou patients. De plus, il existe un modèle de simulation qui se présente sous la forme d'une extension possible d'un dossier patient.
- **Type Élément de base** – Description des éléments de base permettant la production de stimuli, le recueil de données issues des différentes modalités, ainsi que toutes les opérations de connexion entre ces éléments.
- **Élément de base** – Ce sont les instanciations des éléments de base tels qu'ils sont utilisés dans des épreuves.
- **Type Épreuve (Type Test, Type Exercice)** – Il s'agit du concept correspondant aux descriptions d'agrégations d'éléments de base. Ces descriptions devront, tout comme l'ont été les éléments de base pour construire ces types épreuves, être paramétrées/spécifiées et connectées afin de constituer un processus de soins. Ce concept sera décliné aussi bien dans l'environnement de diagnostic que dans celui de remédiation. Un type épreuve peut être composé d'autres objets du même type.

- **Epreuve (Test, Exercice)** – Ici on retrouve les instanciations, corrélées à un patient particulier, des types épreuves telles qu’elles ont été précédemment décrites.
- **Type Processus de soins (Type Arbre diagnostique, Type Programme de remédiation)** – Ce concept correspond aux descriptions des processus pouvant être mis en œuvre lors de la prise en charge du patient. Tout comme pour les descriptions des types épreuves, ces descriptions concernent aussi bien l’environnement de diagnostic que celui de remédiation. Un type processus de soin peut être composé d’autres objets du même type.
- **Processus de soins (Arbre diagnostique, Programme de remédiation)** – Ce sont les instanciations des types processus. Elles se rapportent bien évidemment à un patient spécifique.
- **Type Résultat** – Il s’agit du concept permettant la description des résultats tels qu’ils pourront être retournés par des types épreuves.
- **Résultat** – Les réalisations consécutives à l’accomplissement de certaines épreuves par un patient.
- **Type Rapport** – Concept de description des données résultant de l’exécution d’un type processus de soins.
- **Rapport** – Instance incorporant tout ou partie des résultats obtenus en fin de processus de soins par le patient.

Hormis la possibilité de mimer le comportement d’un patient, le modèle de simulation pourrait permettre d’évaluer l’utilisation même du système (pour un exemple d’une telle approche à l’aide d’agents, consulter (De Loor et al., 2006; Le Bodic, De Loor, Calvet, & Kahn, 2004; Le Bodic, De Loor, & Kahn, 2003)). Nous ne développerons pas plus cet aspect du futur système, mais nous garderons à l’esprit qu’avoir recours à ce type de simulation doit être fonctionnellement possible.

Dans le cas présent, on notera que les éléments de base font partie d’un framework de type « white-box » alors que les deux autres, les processus et épreuves, sont de type « black-box » (Fayad & Schmidt, 1997). C’est-à-dire que ces derniers sont plus faciles à manipuler pour les utilisateurs mais aussi plus difficiles à concevoir pour les développeurs du framework, car ils doivent anticiper sur tous les types de besoins des utilisateurs. Conformément à Bäumer et al. (1997), on signalera d’ailleurs que (**Figure 22**):

- Les types éléments de base constituent les couches du domaine métier.
- Les types épreuves constituent les couches de la section métier.
- Les types processus constituent les couches applicatives.

On peut également remarquer qu’une application doit d’abord être conçue/spécifiée avant de pouvoir être exécutée (voir la section **L’intégration des processus métiers**). A ce titre, on distingue deux composants du système qui manipuleront les classes du domaine :

- L’éditeur qui permet de gérer l’édition des processus, c’est-à-dire qui offre la possibilité de définir leurs classes et méta-classes constituantes. Cet éditeur est impliqué dans la spécification des processus de soins et fait partie du serveur.
- Le runtime qui permet d’en gérer l’exécution, c’est-à-dire qu’il instancie et monitore des classes (Arney et al., 2014). Ce composant est présent aussi bien sur la plateforme côté serveur que sur celle côté client.

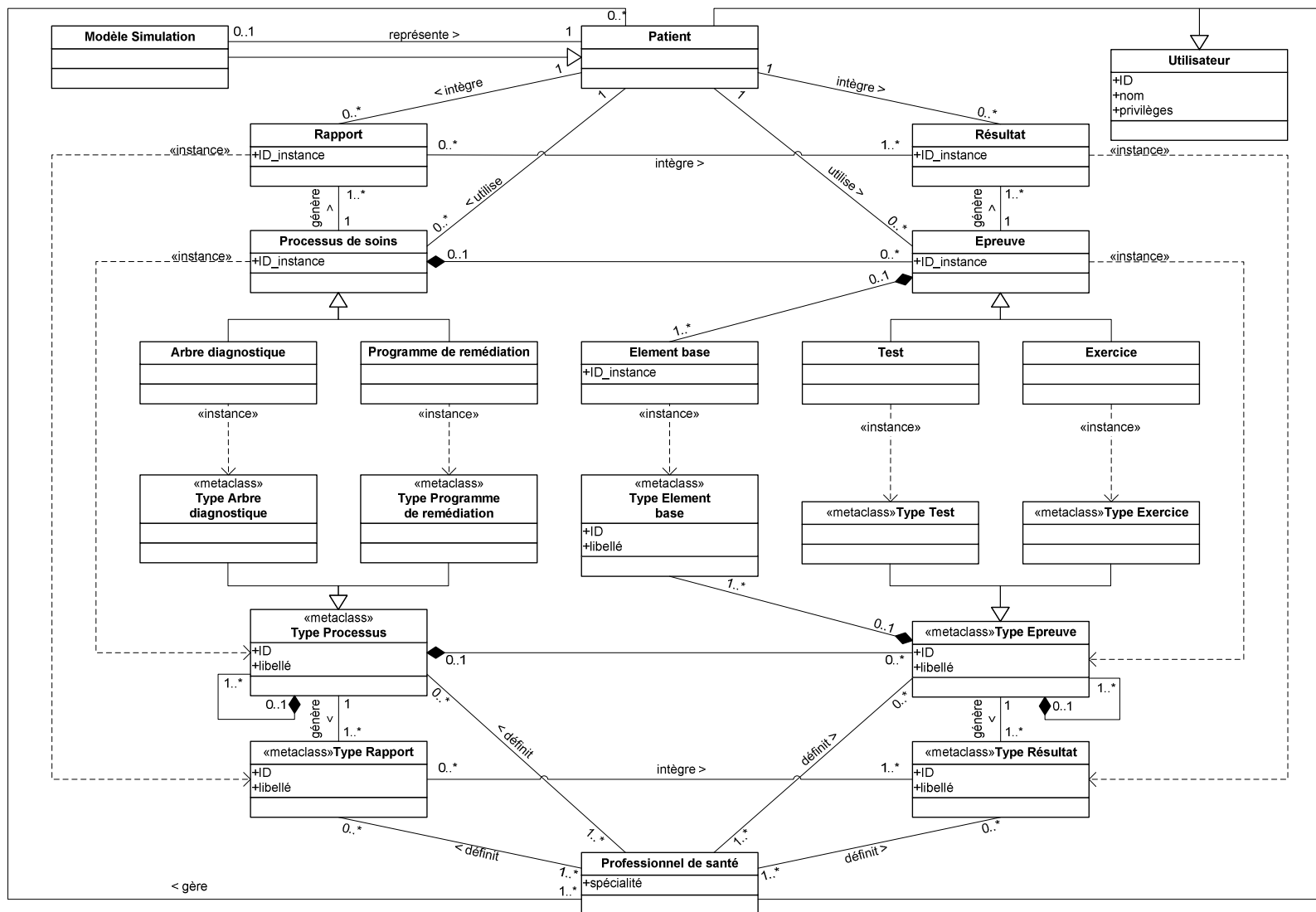


Figure 45 : Diagramme de classe métier (serveur)

Du côté du client, nous avons identifié les concepts du domaine suivants (qui forment une restriction de ce qu'on trouve côté serveur)(**Figure 46**) :

- **Utilisateur (Patient)** – Ce concept englobe les acteurs utilisant l'application.
- **Type Élément de base** – Description des éléments de base permettant : la production de stimuli, le recueil de données issues des différentes modalités, ainsi que toutes les opérations de connexion entre ces éléments.
- **Élément de base** – Ce sont les instanciations des éléments de base tels qu'ils sont utilisés dans des épreuves.
- **Type Epreuve (Type Test, Type Exercice)** – Il s'agit du concept correspondant aux descriptions d'agrégations d'éléments de base. Ces descriptions devront, tout comme l'ont été les éléments de base, être paramétrées/spécifiées et connectées afin de constituer un processus de soins. Ce concept sera décliné aussi bien dans l'environnement de diagnostic que dans celui de remédiation.
- **Epreuve (Test, Exercice)** – Ici on retrouve les instanciations, corrélées à un patient particulier, des épreuves telles qu'elles ont été précédemment décrites.
- **Type Processus de soins (Type Arbre diagnostique, Type Programme de remédiation)** – Ce concept correspond aux descriptions des processus pouvant être mis en œuvre lors de la prise en charge du patient. Tout comme pour les descriptions des épreuves, ces descriptions concernent aussi bien l'environnement de diagnostic que celui de remédiation.
- **Processus de soins (Arbre diagnostique, Programme de remédiation)** – Ce sont les instanciations des processus. Elles se rapportent bien évidemment à un patient précis.
- **Type Résultat** – Il s'agit du concept permettant la description des résultats tels qu'ils pourront être retournés par des épreuves.
- **Résultat** – Les réalisations consécutives à l'accomplissement de certaines épreuves par un patient.
- **Type Rapport** – Concept de description des données résultant de l'exécution d'un processus de soins.
- **Rapport** – Instance incorporant tout ou partie des résultats obtenus en fin de processus de soins par le patient.

A l'heure actuelle, on remarquera que nous n'avons pas mis en lumière les différences entre les modes de fonctionnement online et offline. On pourra également signaler que cette approche à double niveau, avec d'un côté la définition de notre framework MA, qui sert de modèle de référence, et de l'autre celle de méta-classes génériques, qui servent d'archétypes, permet la conception de systèmes évolutifs disposant de classes interopérables qui modélisent l'information médicale (Kalra, 2006; Sachdeva & Bhalla, 2012). Ainsi, un tel système permet de stocker simultanément des informations produites par les activités médicales mais aussi les connaissances relatives à l'accomplissement de ces activités au travers des archétypes. Cependant, cela implique qu'il faudra procéder à la gestion des versions de ces archétypes (Isern & Moreno, 2008).

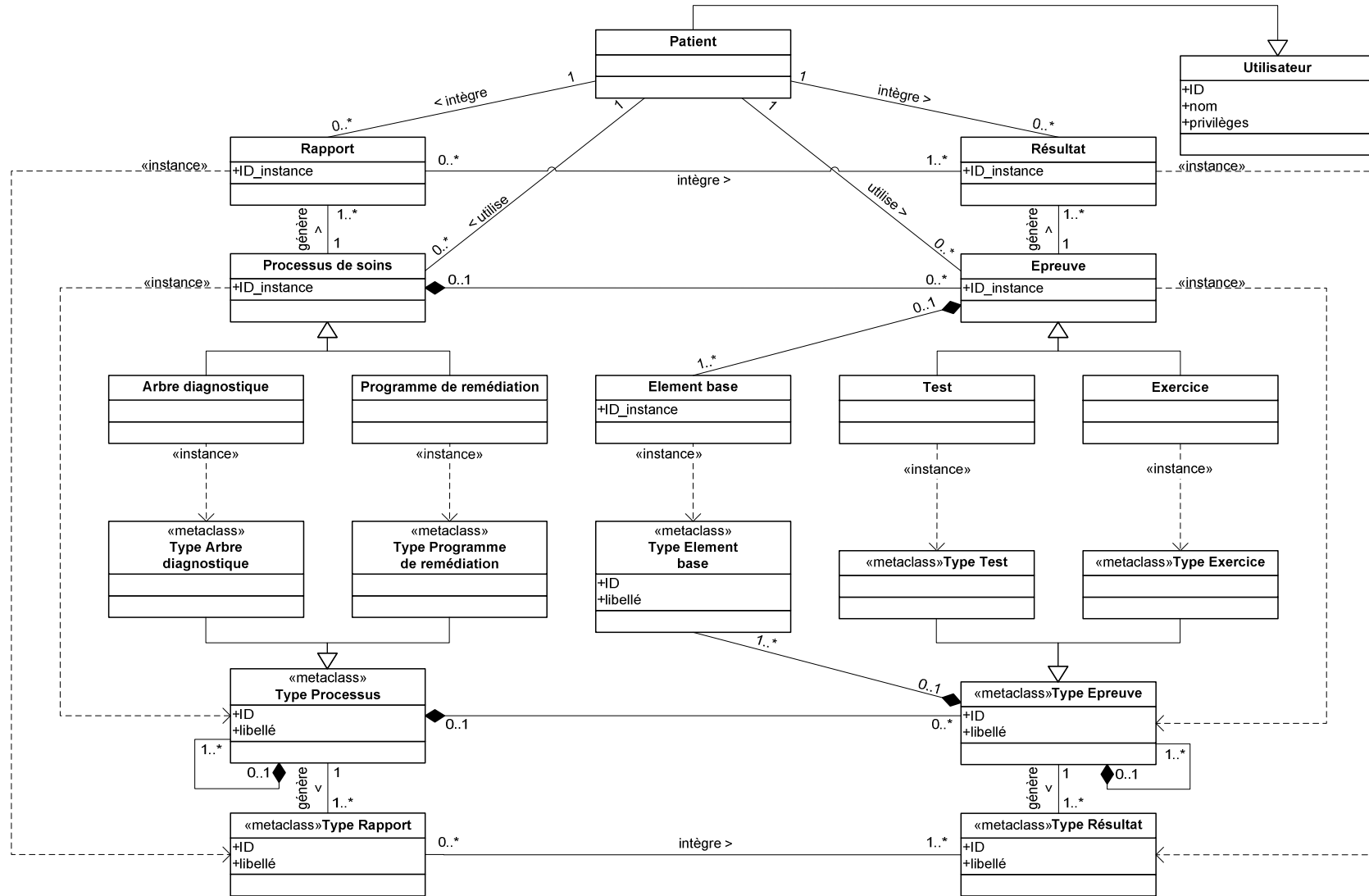


Figure 46 : Diagramme de classe métier (client)

B) Les exigences fonctionnelles

Cette activité a pour but de définir ce que doit faire le système d'un point de vue métier, c'est-à-dire d'établir clairement l'expression des besoins des utilisateurs. Au travers de cette partie de la démarche, les utilisateurs finaux doivent pouvoir valider, auprès des développeurs du système, l'analyse qui est faite de leur besoin (Rational Software, 2001). A cette occasion, cette analyse permet d'établir trois résultats (Gabay & Gabay, 2008) :

- La définition des cas d'utilisation métiers et leur description générale, à l'aide de diagrammes de cas d'utilisation systèmes.
- Les scénarii des cas d'utilisation métiers, grâce à des diagrammes de séquence systèmes.
- La navigation générale du système à étudier, c'est-à-dire le fonctionnement à haut niveau de l'IHM modélisé par des schémas de navigation générale.

Ces représentations doivent clairement montrer comment les acteurs du système interagissent avec lui et ce que le système fait pour eux. Nous expliciterons d'ailleurs les différents environnements d'utilisation du futur système (Obrenović & Starčević, 2006). Au terme de cette deuxième activité, l'expression des besoins est couverte.

1) L'élaboration du diagramme des cas d'utilisation système

FICHE GUIDE – FG4	
Activité 2 : Exigences fonctionnelles	
Sous-activité 2.1 : Elaboration du diagramme des cas d'utilisation système (DCU système)	
Objectif	Recueillir et décrire les besoins métiers des acteurs du système (boîte noire)
Point de départ	Concepts métiers identifiés et définis dans le DCL métier et le glossaire métier
Point d'arrivée	Concepts métiers identifiés définis dans le DCL métier et le glossaire métier
Démarche d'élaboration	
<p>1 - Identifier les acteurs métiers du système (acteurs internes et acteurs et/ou systèmes externes) en prenant comme base ceux définis dans le DAC.</p> <p>2 - Identifier les cas d'utilisation métiers.</p> <p>3 - Représenter les interactions entre les acteurs métiers et les cas d'utilisation métiers.</p> <p>4 - Définir les dépendances entre les cas d'utilisation métiers.</p> <p>5 - Décrire de manière générale les cas d'utilisation métiers.</p>	

Tableau 10 : Fiche Guide 4 pour UP7 (d'après (Gabay & Gabay, 2008))

A partir des diagrammes précédemment établis, nous proposons une vision générale des cas d'utilisation de chacun des sous-systèmes du dispositif futur. Ces cas d'utilisation sont des contraintes fonctionnelles du système (Alhir, 2002). Commençons maintenant avec une description générale des cas d'utilisation du sous-système permettant l'analyse et la validation de théorie (**Figure 47**) :

- « **Définir des processus de soins** » – Il s'agit de la création de processus de soins, à savoir de type d'arbres diagnostiques ou de type de programme de remédiation selon que ces processus sont destinés à être utilisés dans le sous-système d'aide diagnostique ou dans celui concernant la remédiation. Comme précédemment indiqué, un éditeur livré avec le produit final doit permettre le développement de ce type de contenu.
- « **Définir des épreuves** » – Nous cherchons ici à permettre à l'utilisateur averti d'enrichir le dispositif. Il va pouvoir définir de nouvelles épreuves, c'est-à-dire des types de tests ou des types d'exercices. Ce nouveau contenu pourra être utilisé dans de nouveaux processus de soins. Ce mode de création/édition s'appuiera sur la composition et le paramétrage d'éléments de base permettant la production de stimuli (sonores, visuels, ...) et le recueil de données (sonore via le micro, ou relatives au déplacement oculaire ou encore à l'activité électrique cérébrale, ...) respectivement destinés et produites par le patient. Ainsi, les données à analyser ainsi que les critères à évaluer (temps de réponse, qualité des réponses, ...) doivent

être définis par l'utilisateur de l'éditeur et seront récupérés à l'exécution, une fois que le patient aura été soumis à l'épreuve, sous la forme de résultats.

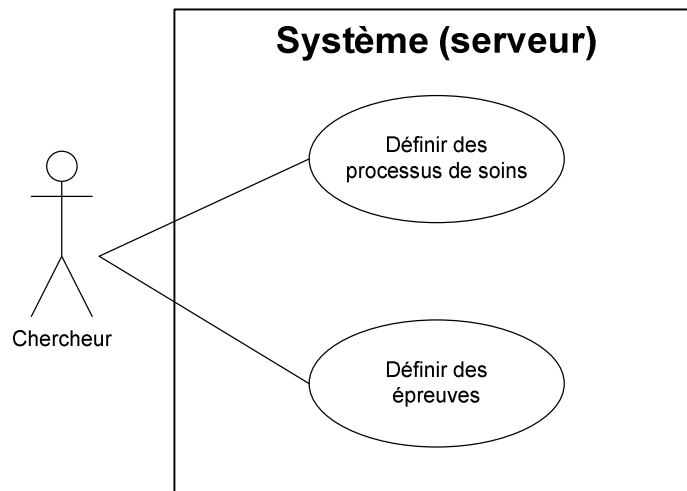


Figure 47 : Diagramme des cas d'utilisation système de la partie d'analyse et validation de théorie

Notre second diagramme des cas d'utilisation système va, quant à lui, porter sur le sous-système d'aide diagnostique. En voici une description générale (**Figure 48**) :

- « **Réaliser un bilan** » – La réalisation d'un bilan consiste, pour le professionnel de santé, à choisir un arbre diagnostique (Hadorn, 1995) qui va être utilisé pour piloter le client du patient. Cet arbre va permettre la réalisation d'une séquence de tests auxquels le patient va se soumettre.
- « **Passer les tests** » – Le patient se soumet aux tests successifs qui lui sont présentés. A la suite de chaque test, des résultats sont générés et analysés afin de déterminer le test suivant à faire passer.
- « **Analyse et conclusion** » – Une fois l'arbre diagnostique complètement exécuté, le serveur procède à la production de rapports à partir des résultats produits par chaque test. A partir de ces rapports, des propositions sont faites au professionnel de santé afin de déterminer, le cas échéant, de quelle pathologie souffre le patient. Il est important de signaler que le praticien dispose toujours d'un pouvoir de décision total, ainsi il peut décider de ne pas suivre les recommandations du dispositif pour poser lui-même le diagnostic qu'il juge pertinent (voir la sous-section **L'aide à la décision**). Il reste également toujours possible que le praticien ne détecte aucune anomalie au niveau des fonctions qu'il a souhaitées évaluer.
- « **Calibrage de la simulation** » – Le modèle de simulation, sensé reproduire le comportement d'un patient, est calibré en fonction des conclusions établies par le dispositif et le professionnel de santé. Ce modèle doit permettre de simuler, avec des performances comparables, la réalisation des tests que vient de subir le patient.

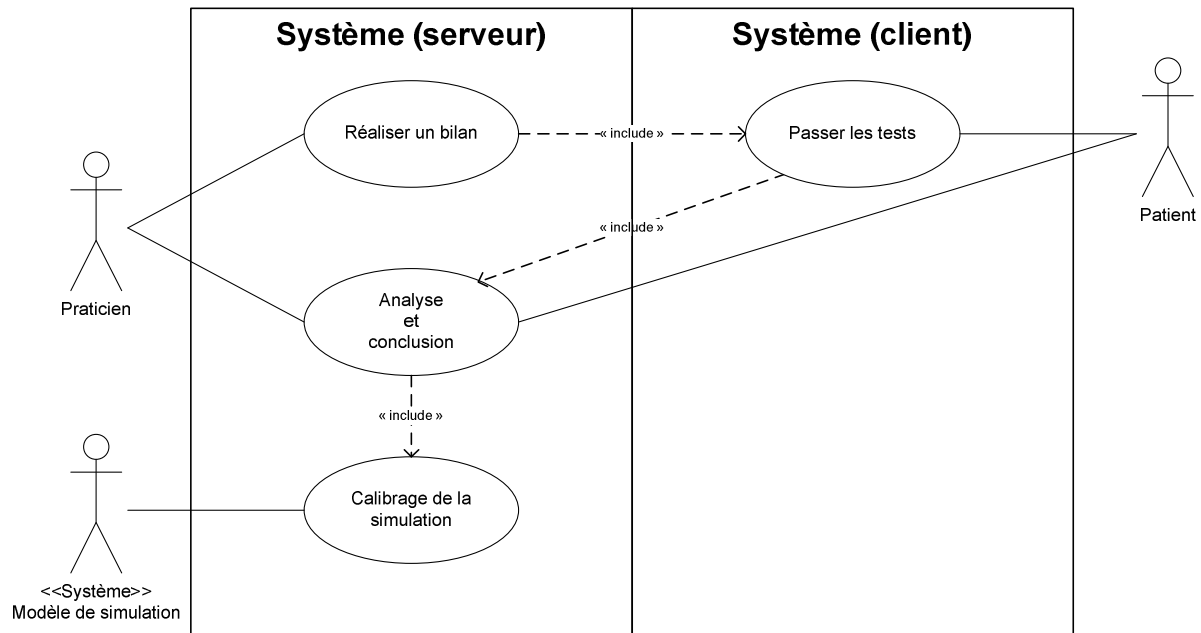


Figure 48 : Diagramme des cas d'utilisation système du sous-système d'aide diagnostique

Passons finalement à une description générale des cas d'utilisation système dans l'environnement dynamique de remédiation, c'est-à-dire à une description générale du dernier sous-système étudié (**Figure 49**):

- **« Planifier le programme de remédiation »** – En fonction du diagnostic qui a été posé pour le patient, le professionnel de santé, est amené à choisir parmi plusieurs programmes de remédiation, que le système a jugés pertinents. Ce programme devra être suivi par le patient, ainsi le client va être configuré afin qu'il puisse proposer au patient les exercices qui constituent le programme de remédiation.
- **« Suivre le programme »** – Le patient va se soumettre aux exercices. Selon les exercices, il se peut qu'il les réalise soit online soit offline. Dans tous les cas, des résultats seront produits pour chaque exercice fait par le patient.
- **« Simuler le suivi du programme »** – En parallèle de la réalisation des exercices par le patient, le système peut être amené à en simuler la réalisation par le modèle de simulation, qui a été calibré lors de la phase de diagnostic. Tout comme dans le cas du patient, chaque exercice complété donnera lieu à la production de résultats.
- **« Analyser les résultats »** – A partir des résultats générés par le patient le système va établir des rapports. Ces rapports vont servir à évaluer les performances globales obtenues lors du suivi du programme de remédiation. Il en va d'ailleurs de même pour le modèle de simulation dont nous évaluerons aussi les performances.
- **« Ajuster la remédiation »** – Par analyse des rapports précédemment produits, il va être possible de réévaluer la condition du patient. Pour cela, nous disposons effectivement à ce stade de deux types de données fortes : les performances réelles du patient, mesurées lors de sa réalisation des exercices, et ses performances attendues, obtenues à l'aide du modèle de simulation. Il est donc possible d'extraire des informations à la fois qualitatives, en ce qui concerne les critères évalués lors des exercices, mais aussi quantitatives, par comparaison entre résultats réellement produits et résultats attendus. Une modification du programme de remédiation, afin de mieux correspondre aux nouveaux besoins du patient, est alors rendue possible.

- « **Recalibrage de la simulation** » – Le patient étant amené à évoluer lors de sa prise en charge, il est judicieux de répercuter ses progrès au modèle de simulation le représentant. Ainsi, il est prévu de faire périodiquement, sur la demande du praticien, une mise à jour du modèle afin qu’il corresponde à la réalité, c’est-à-dire qu’il permette toujours de simuler les performances actuelles du patient. Une telle mise à jour revient à recalculer les valeurs des paramètres d’un modèle de simulation existant d’une manière similaire à celle initialement utilisée pendant le calibrage de ce même modèle.

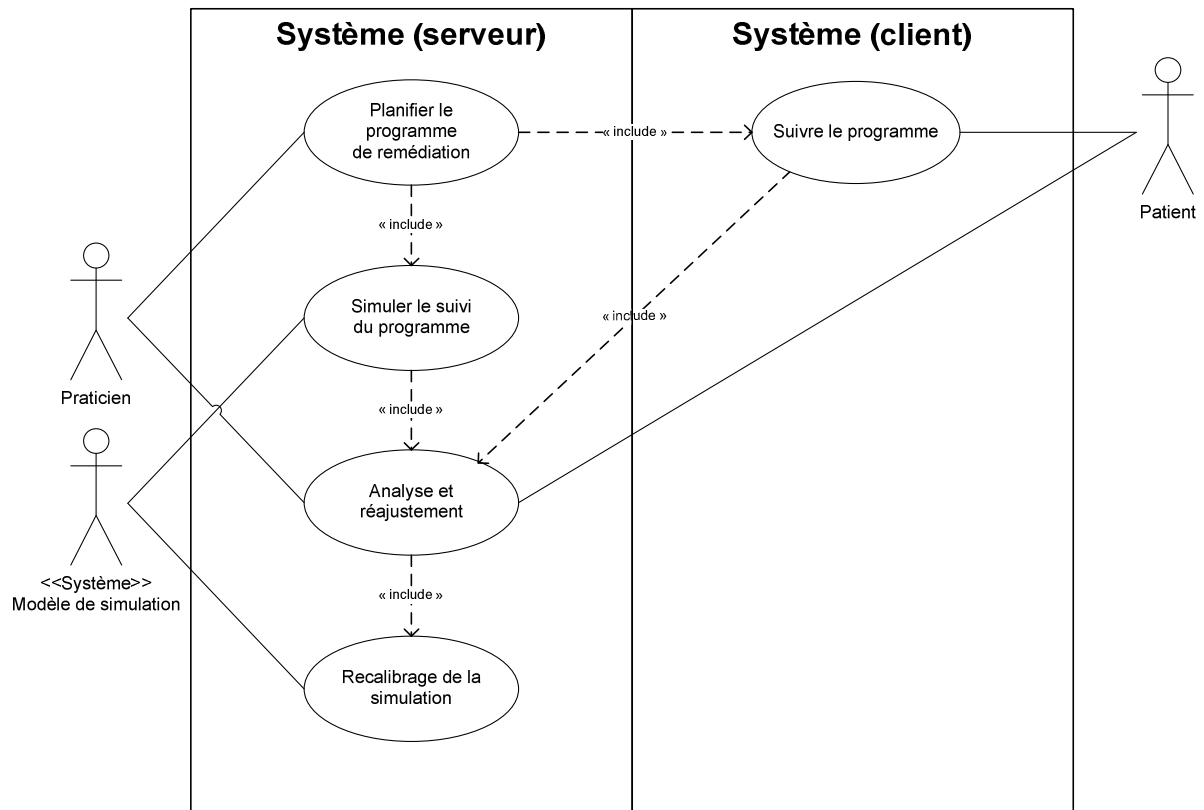


Figure 49 : Diagramme des cas d'utilisation système dans l'environnement dynamique de remédiation

Pour rappel, la **Figure 39** (voir sous-section **L'élaboration du schéma de contexte du domaine**) présente les environnements dans lesquels s'inscrivent ces cas d'utilisation (Obrenović & Starčević, 2006).

2) L'élaboration des diagrammes de séquence système

FICHE GUIDE – FG5	
Activité 2 : Exigences fonctionnelles	
Sous-activité 2.2 : Elaboration des diagrammes de séquence système (DSE)	
Objectif	Définir les interactions entre les acteurs métiers et le système (boîte noire) pour tous les cas d'utilisation métiers
Point de départ	Besoins métiers recueillis sous la forme de cas d'utilisation et décrits de manière générale
Point d'arrivée	Interactions entre acteurs métiers et système définis dans le DSE
Démarche d'élaboration	
Pour chaque cas d'utilisation (CU) :	
<ol style="list-style-type: none"> 1 - Reporter le ou les acteurs du CU sélectionné sur le diagramme de séquence système. 2 - Représenter le système sous forme d'objet. 3 - Déterminer les messages échangés entre les acteurs et le système (synchrones, asynchrones, résultats). 4 - Représenter les fragments d'interaction combinés si nécessaire (loop, alt, opt...). 	

Tableau 11 : Fiche Guide 5 pour UP7 (d'après (Gabay & Gabay, 2008))

Nous pouvons maintenant commencer à représenter les différentes interactions entre les acteurs et le système. Ces interactions vont s'effectuer conformément aux processus métiers que nous avons identifiés (voir la sous-section **L'élaboration du diagramme d'activité**). Ainsi, pour le premier des trois sous-systèmes (voir la sous-section **L'élaboration du schéma de contexte du domaine**), nous identifions les DSE système suivants (des cas d'utilisation de la **Figure 47**) :

- « Définir des processus de soins » (**Figure 50**)
- « Définir des épreuves » (**Figure 51**)

Il peut être intéressant de noter que ces deux diagrammes sont très similaires. En effet, chacun de ces deux cas d'utilisation consiste à spécifier puis à agréger des éléments de granularité moindre afin de définir des éléments de granularité supérieure.

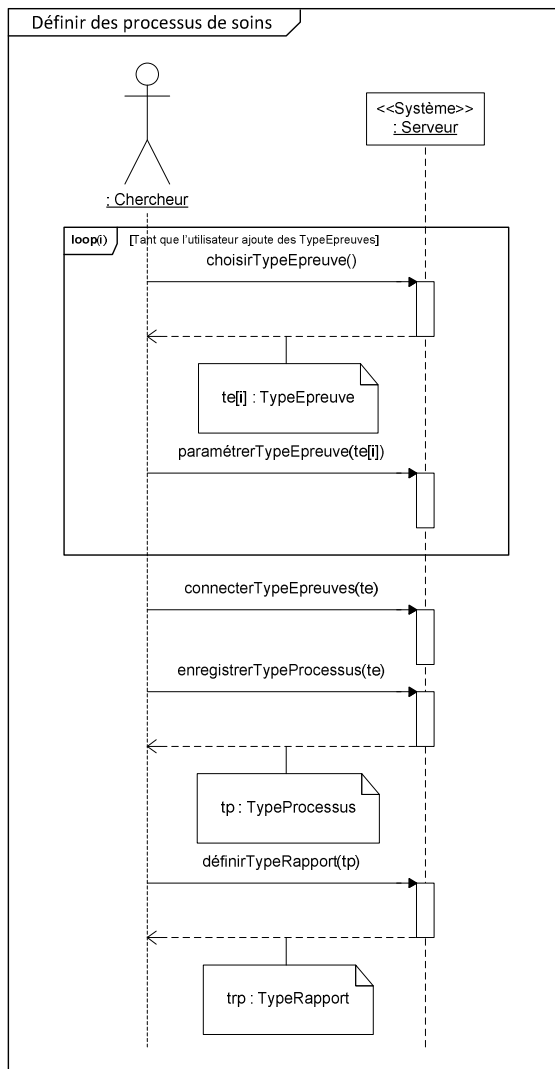


Figure 50 : Diagramme de séquence du cas d'utilisation « Définir des processus de soins »

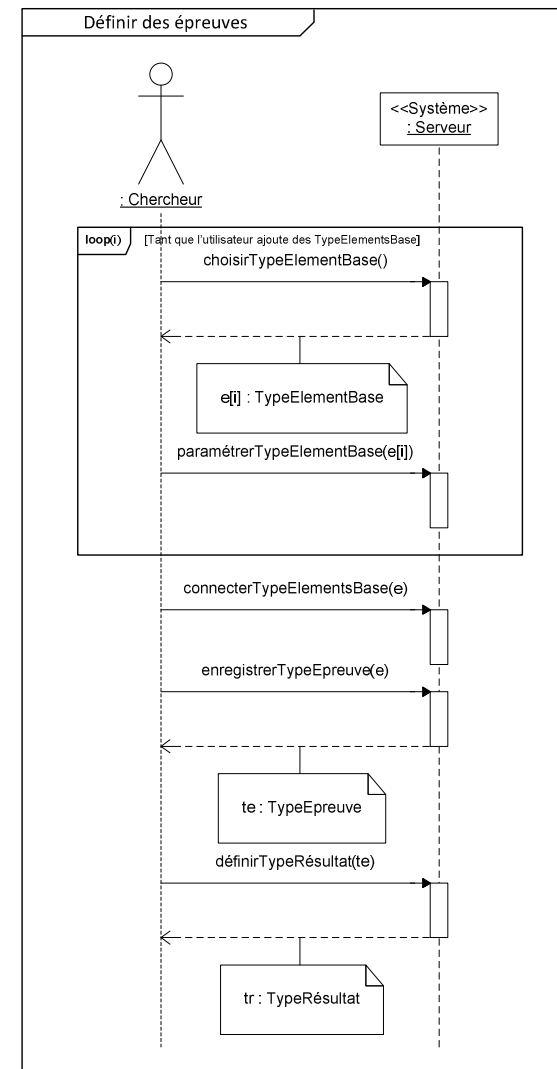


Figure 51 : Diagramme de séquence du cas d'utilisation « Définir des épreuves »

Dans le cas du sous-système d'aide diagnostique, nous représentons les DSE suivants (des cas d'utilisation de la **Figure 48**) :

- « Réaliser un bilan » (**Figure 52**)
- « Passer les tests » (**Figure 53**)
- « Analyse et conclusion » (**Figure 54**)
- « Calibrage de la simulation » (**Figure 55**)

On remarquera que ces cas d'utilisations vont se succéder les uns aux autres de façon séquentielle. L'ordre est celui dans lequel les différents cas ont été précédemment présentés.

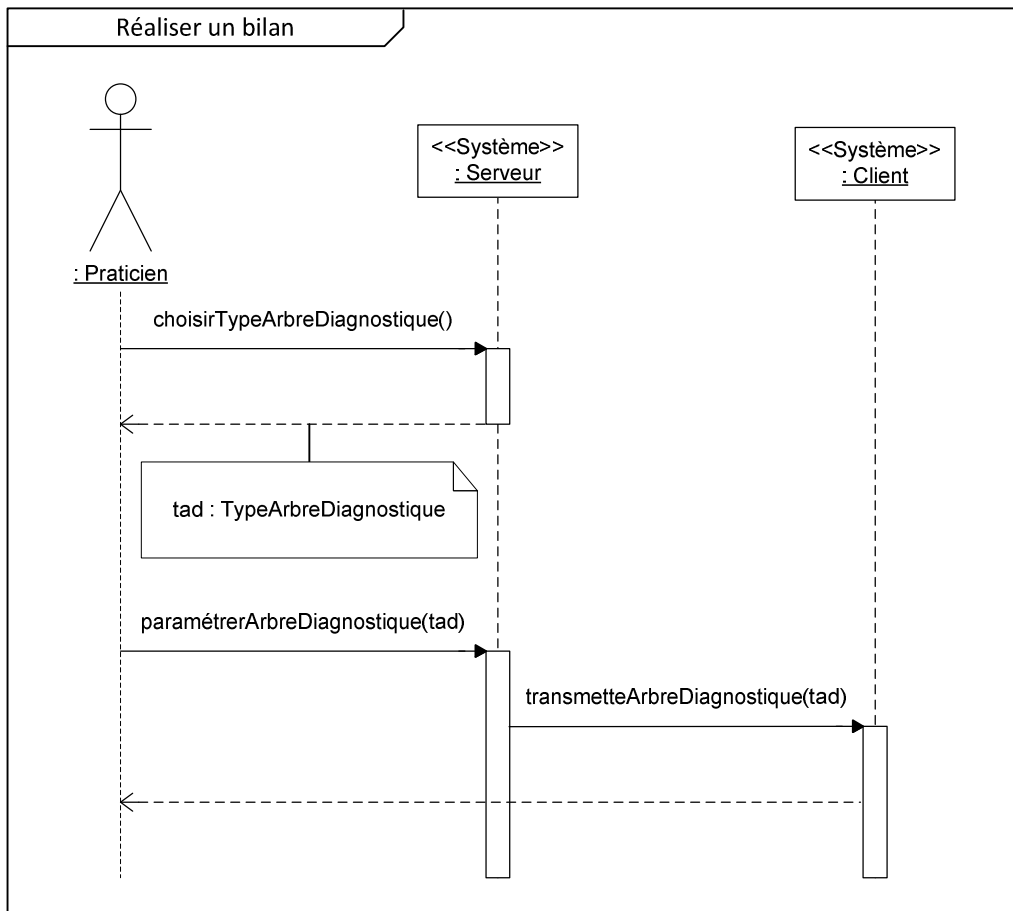


Figure 52 : Diagramme de séquence du cas d'utilisation « Réaliser un bilan »

La réalisation d'un bilan, le passage de tests et la phase d'analyse et conclusion se effectués en mode online. C'est pourquoi sont impliqués le client et le serveur ainsi que leurs utilisateurs respectifs.

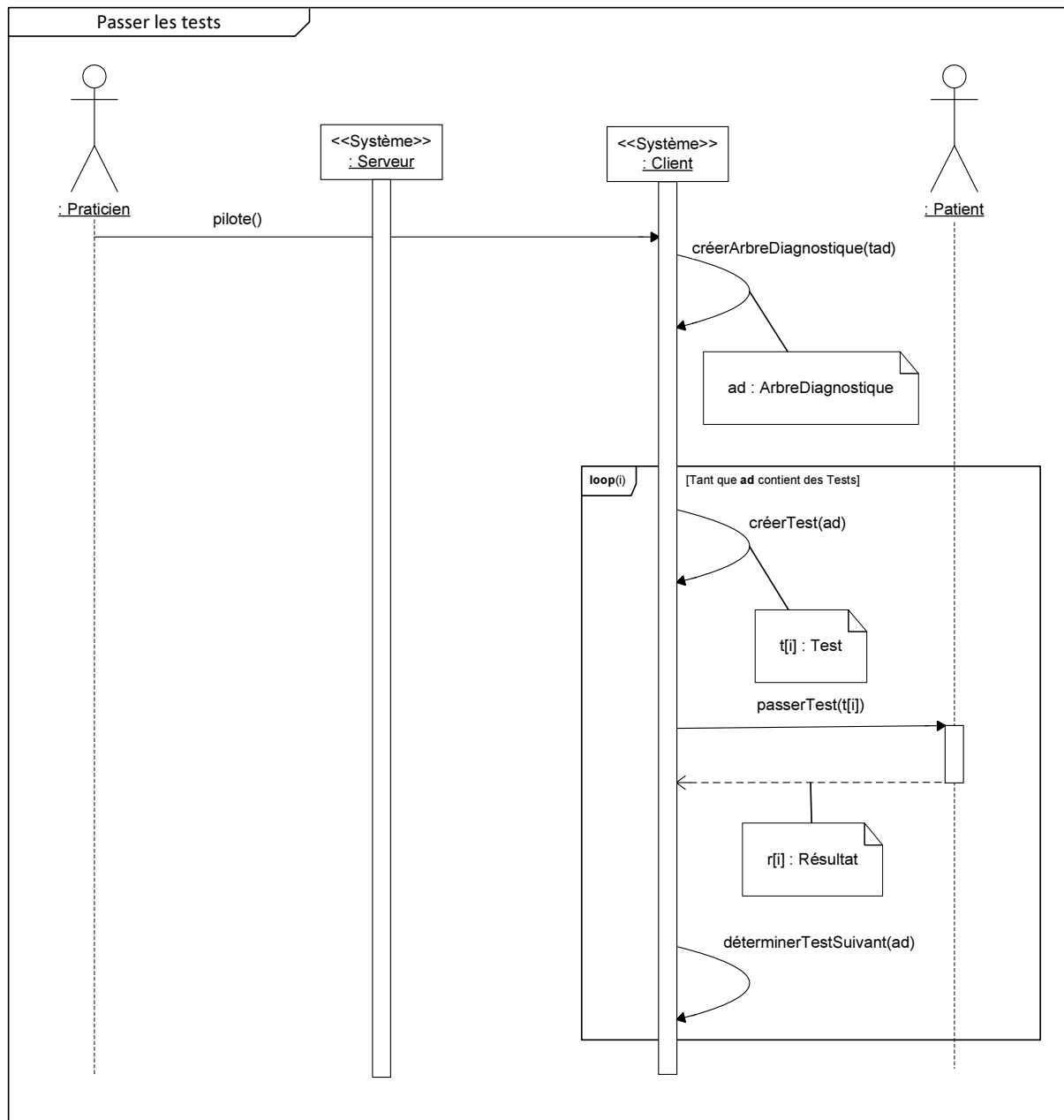


Figure 53 : Diagramme de séquence du cas d'utilisation « Passer les tests »

Le patient va se voir proposer un nombre de tests dépendant de l'arbre diagnostique sélectionné par le praticien. A l'issue de chaque test, un résultat est généré qui pourra être utilisé pour déterminer le test suivant, en accord avec l'arbre diagnostique utilisé. Les résultats ainsi générés seront transmis au praticien une fois l'arbre diagnostique entièrement exécuté.

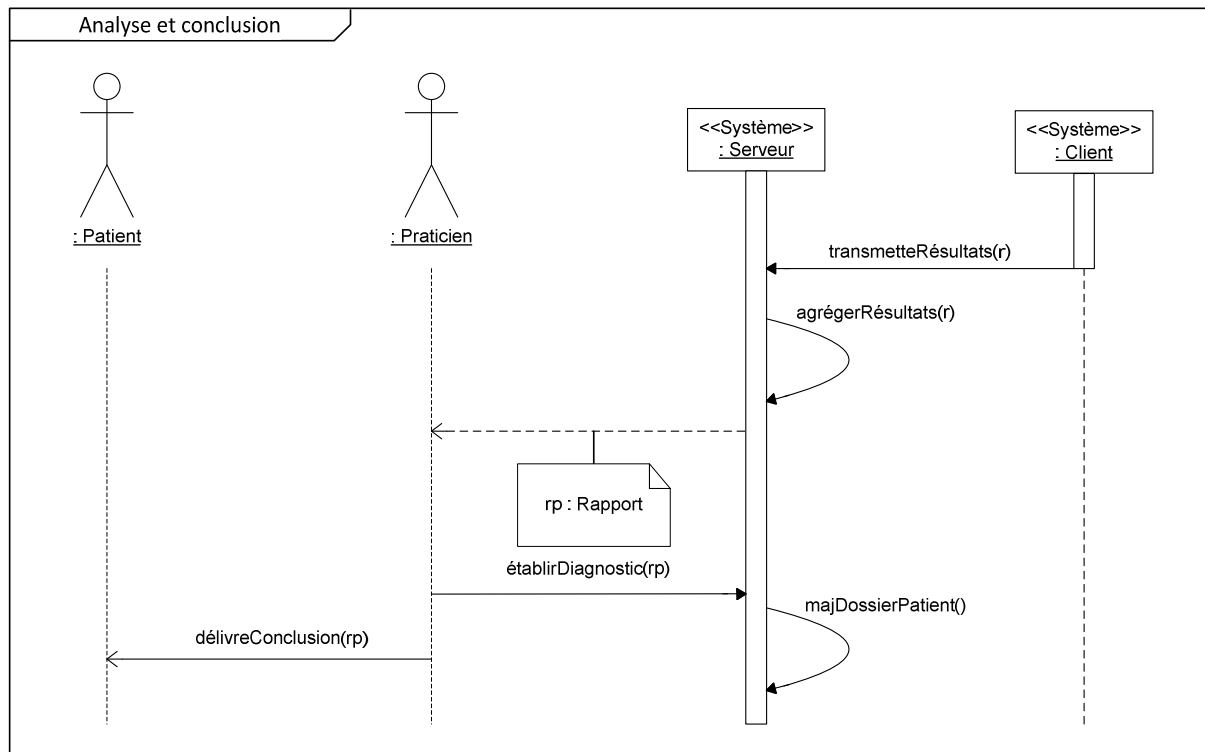


Figure 54 : Diagramme de séquence du cas d'utilisation « Analyse et conclusion »

Le rapport généré en fin de diagnostic, à partir des résultats reçus, va pouvoir être ajouté au dossier du patient. Consécutivement à cela, le modèle de simulation va pouvoir être instancié et calibré.

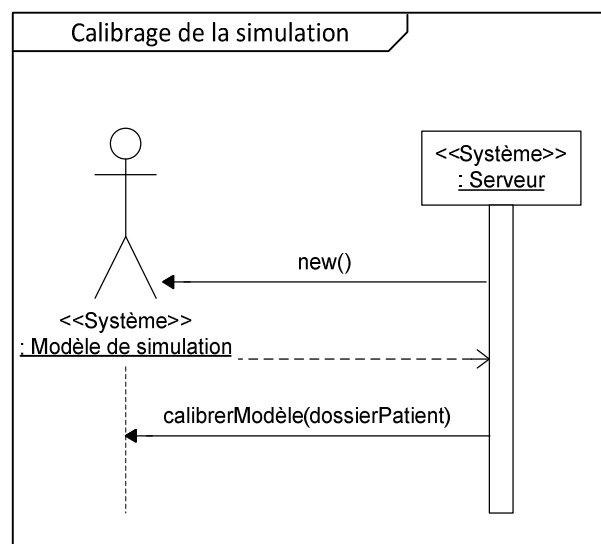


Figure 55 : Diagramme de séquence du cas d'utilisation « Calibrage de la simulation »

Cette partie du dispositif, visant à simuler l'effet de la remédiation sur le patient, représente la capacité du système à proposer une aide à la décision abordant le problème du pronostic de l'efficacité du traitement chez le patient (Greenes, 2014). Ce genre de considération est primordial avant d'engager le patient dans un traitement spécifique.

Pour finir, nous présentons les DSE du dernier des trois sous-systèmes, le sous-système de l'environnement dynamique de remédiation, suivants (des cas d'utilisation de la **Figure 49**) :

- « Planifier le programme de remédiation » (**Figure 56**)
- « Suivre le programme » (**Figure 57**)
- « Simuler le suivi du programme » (**Figure 58**)
- « Analyse et ajustement » (**Figure 59**)
- « Recalibrage de la simulation » (**Figure 60**)

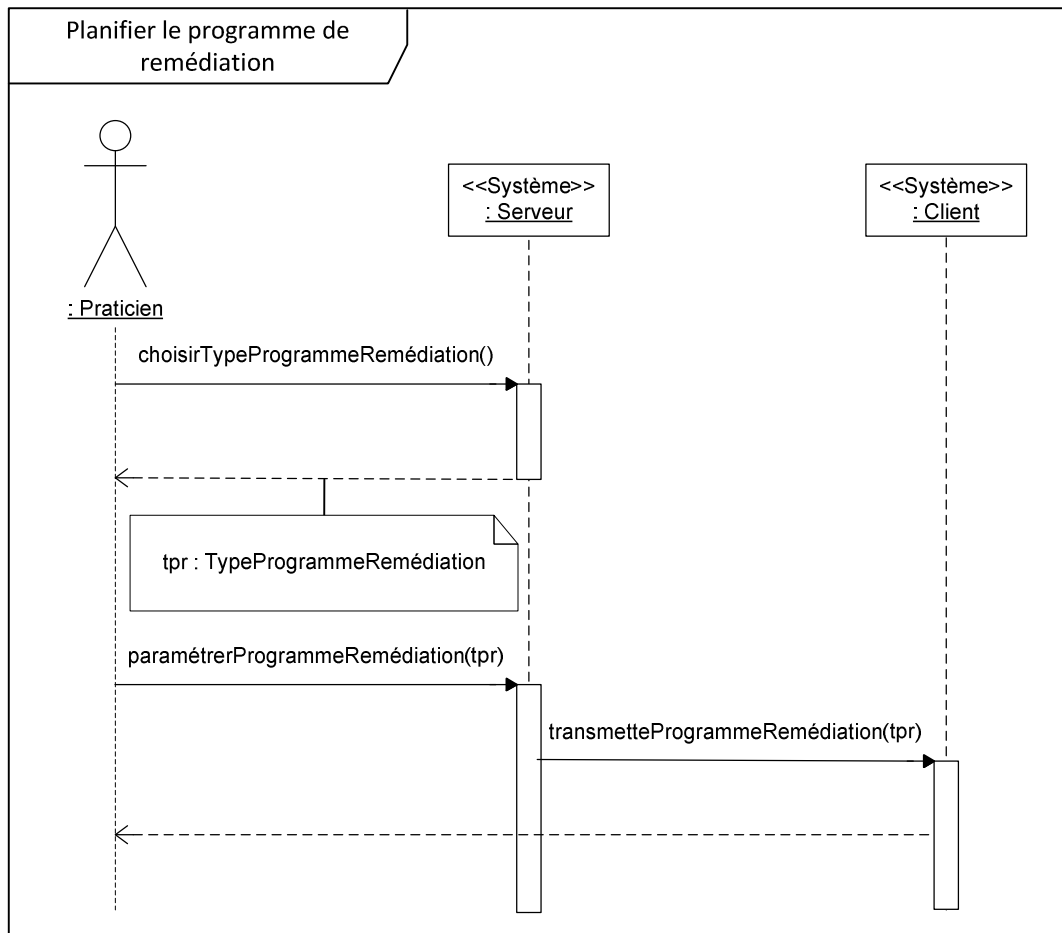


Figure 56 : Diagramme de séquence du cas d'utilisation « Planifier le programme de remédiation »

On peut également remarquer qu'il y a une grande similarité entre la réalisation des bilans et la mise en place de traitements de remédiation adaptés. Effectivement, dans le contexte de l'aide à la décision, on peut considérer qu'un traitement est une sorte de diagnostic dans la mesure où il fournit des informations sur la façon dont le patient y réagit (Greenes, 2014).

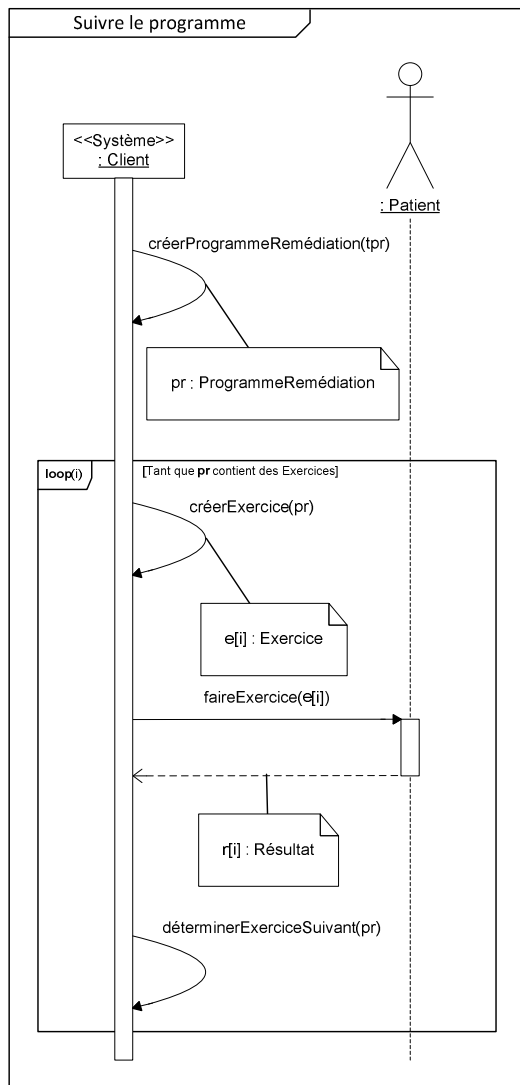


Figure 57 : Diagramme de séquence du cas d'utilisation « Suivre le programme »

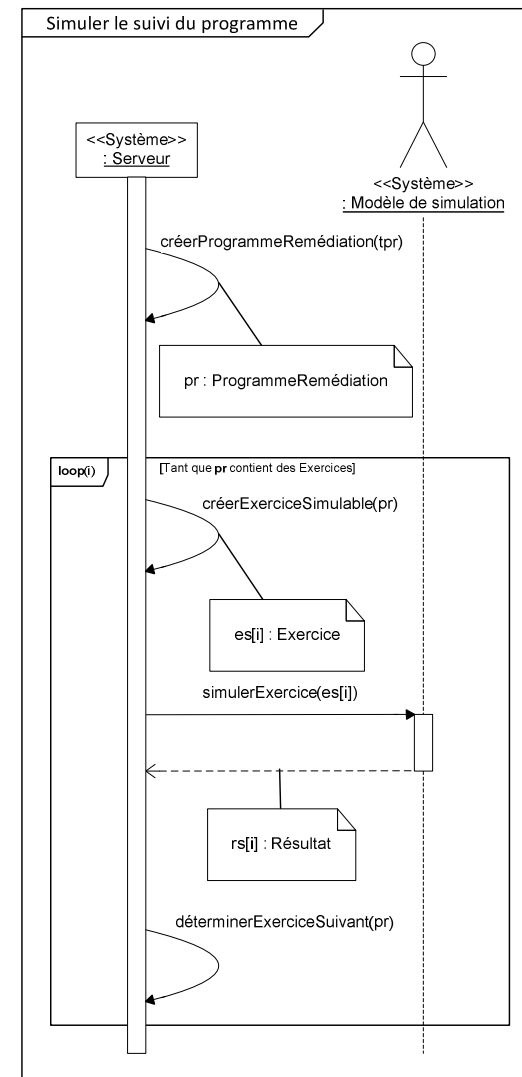


Figure 58 : Diagramme de séquence du cas d'utilisation « Simuler le suivi du programme »

On pourra constater que le modèle de simulation se comportera exactement comme le patient à l'exception près que son exécution se fera du côté seul du serveur. Ce modèle sera, par ailleurs, « attaché » au dossier de patient, et il pourra être géré par l'intermédiaire de consultations de celui-ci.

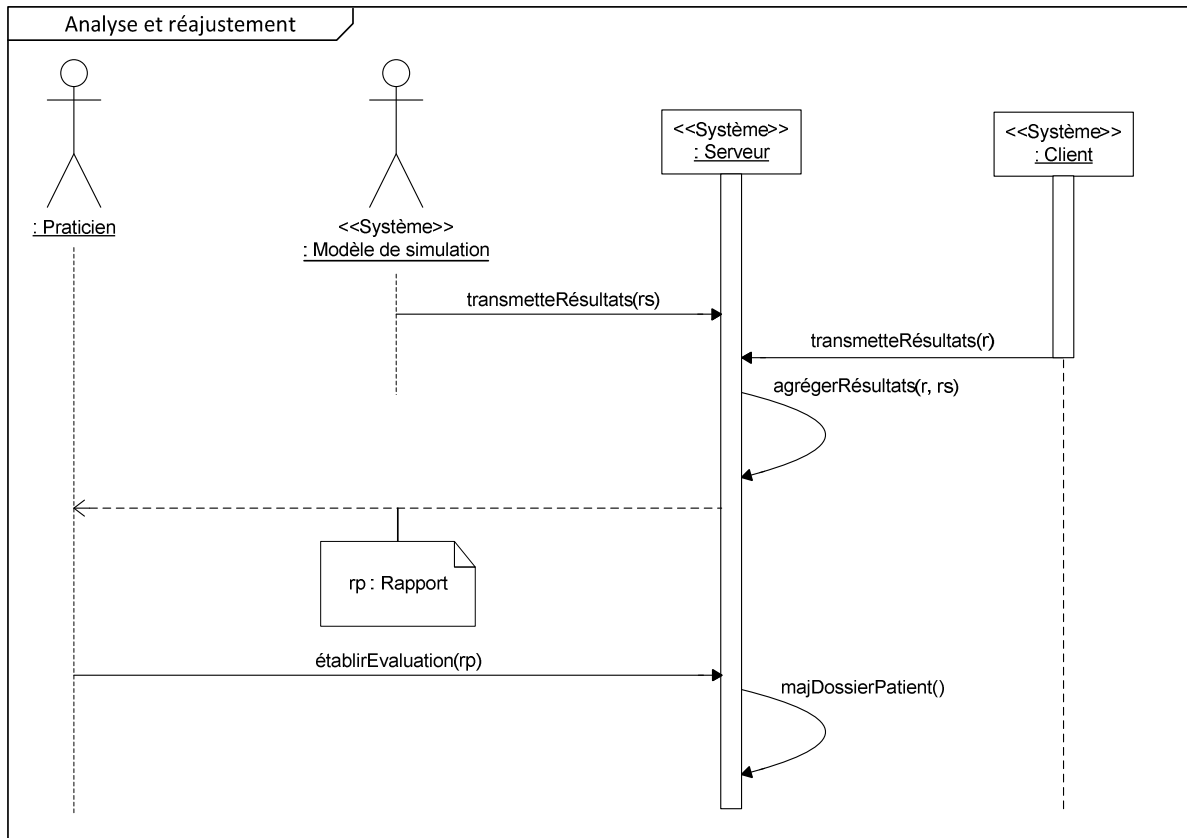


Figure 59 : Diagramme de séquence du cas d'utilisation « Analyse et ajustement »

La phase d'analyse et réajustement sert à intégrer au dossier d'un patient les résultats obtenus par ce dernier, ainsi que ceux éventuellement simulés, à l'issue du programme remédiation. Consécutivement à cela, le modèle de simulation peut nécessiter un recalibrage.

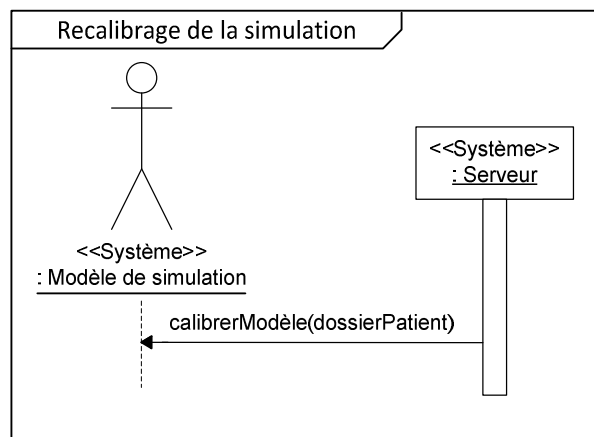


Figure 60 : Diagramme de séquence du cas d'utilisation « Recalibrage de la simulation »

3) L'élaboration du schéma de navigation générale

FICHE GUIDE – FG6	
Activité 2 : Exigences fonctionnelles	
Sous-activité 2.3 : Elaboration du schéma de navigation générale	
Objectif	Déterminer la cinématique des écrans du système
Point de départ	Interactions entre acteurs et système définies pour les cas d'utilisation métiers
Point d'arrivée	L'interface homme-machine générale définie
Démarche d'élaboration	
<p>1 - Identifier les principaux écrans provenant des cas d'utilisation métiers et des interactions décrites dans le DSE système. Les écrans correspondent aux états du schéma de navigation générale si l'on utilise un diagramme d'état-transition pour la représentation de la navigation.</p> <p>2 - Préciser les évènements/conditions associés à la transition entre les écrans.</p> <p>3 - Indiquer l'état début/fin si nécessaire.</p>	

Tableau 12 : Fiche Guide 6 pour UP7 (d'après (Gabay & Gabay, 2008))

Comme nous avons pu le voir précédemment, le système sera constitué de deux parties : un serveur et un client. A ce titre, chaque partie va disposer de ses propres écrans. Dans chacun de ces deux cas, le lancement de l'application destinée au traitement de la dyslexie propose une page d'accueil. Cette dernière centralise l'ensemble des choix offerts à l'utilisateur.

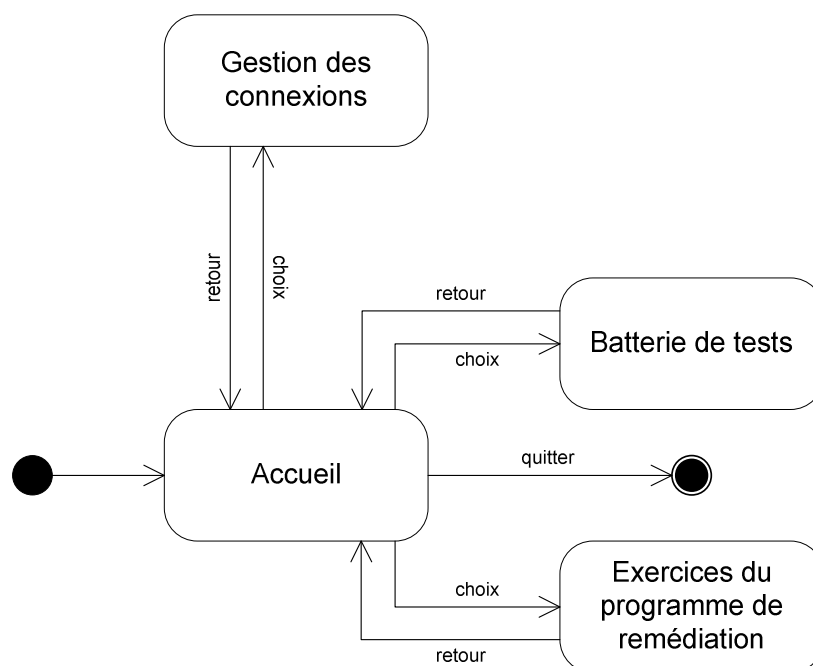


Figure 61 : Schéma de navigation générale du client

Dans le cas du client (**Figure 61**), les choix se limitent à la gestion de connexions avec des serveurs, à l'exécution de bilans constitués de tests pilotés depuis un serveur, et la réalisation d'exercices prescrits dans le cadre d'un programme de remédiation.

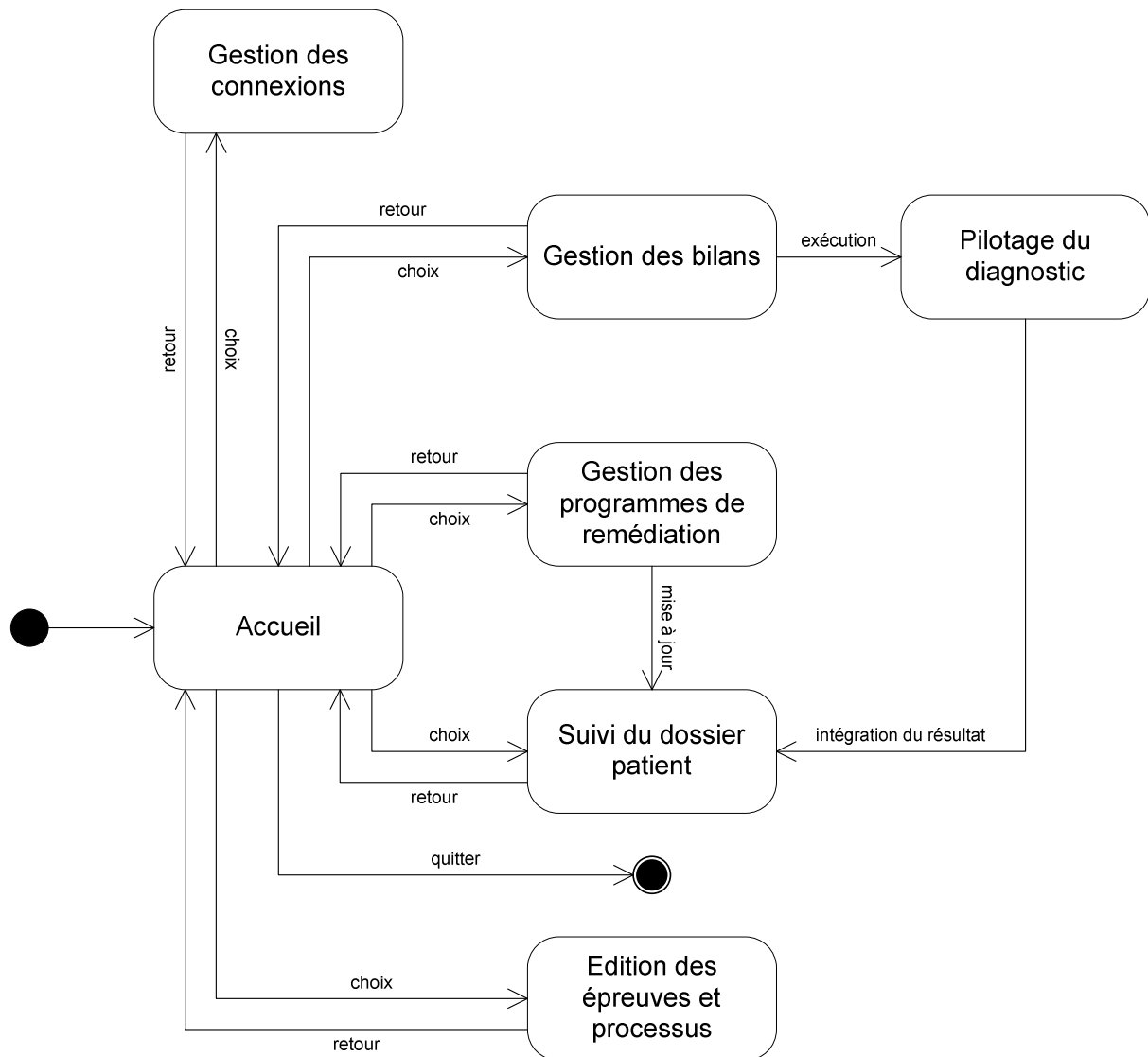


Figure 62 : Schéma de navigation générale du serveur

Du côté du serveur (**Figure 62**), il est spécifiquement possible de lancer l'exécution de bilans, de gérer les prescriptions de remédiation, de consulter et de modifier le dossier patient. L'édition d'épreuves et de processus est également une fonctionnalité exclusive du serveur.

On signalera que la réalisation de bilan aussi bien pour le serveur que pour le client, et a fortiori la possibilité de sélectionner ce choix depuis l'écran d'accueil, est tributaire d'une connexion préétablie entre le client et le serveur.

C) L'analyse des cas d'utilisation

Avec cette troisième activité, la démarche a pour but de cristalliser les besoins précédemment exprimés en une vue informatique du système. Elle permet ainsi de produire cinq résultats :

- La définition et la description détaillée de tous les cas d'utilisation, aussi bien métiers qu'informatiques.
- La description des scénarii pour chaque cas d'utilisation.
- L'identification des différents états des objets étudiés.
- Le maquettage des interfaces utilisateur pour chaque cas d'utilisation.
- Les structures des données impliquées dans chaque cas d'utilisation.

Ces résultats seront matérialisés par des diagrammes de cas d'utilisation, de séquence, d'état-transition et de classe, ainsi que des maquettes d'IHM. À l'issue de cette activité, l'analyse des cas d'utilisation est produite mais non encore consolidée ni validée définitivement et elle devrait couvrir environ 80% des cas. On pourra préciser que les cas d'utilisation les plus structurants pour la modélisation de notre architecture sont ceux (Jacobson, 2008a) :

- Permettant de réduire les risques les plus sérieux : pour cela, il faut s'assurer que l'on peut assurer fonctionnellement et techniquement les contraintes indispensables aux fonctionnalités du système.
- Etant les plus importants pour les utilisateurs finaux : c'est-à-dire qu'il faut identifier les fonctions principales du système et les contraintes fonctionnelles liées à leur utilisation.

1) L'élaboration du diagramme des cas d'utilisation

FICHE GUIDE – FG7	
Activité 3 : Analyse des cas d'utilisation	
Sous-activité 3.1 : Elaboration du diagramme des cas d'utilisation (DCU)	
Objectif	Définir la totalité des cas d'utilisation (métiers et informatiques). Les cas d'utilisation informatiques sont des fonctions complémentaires qui n'ont pas été identifiées lors de l'activité précédente (ex. un module d'administration)
Point de départ	Besoins métiers, interactions acteurs métiers/système, IHM définies
Point d'arrivée	Tous les cas d'utilisation définis dans le DCU
Démarche d'élaboration	
<p>1 - Identifier tous les acteurs du système en prenant comme base ceux définis dans le DCU système de l'activité précédente. Les acteurs informatiques apparaissent à ce niveau de l'analyse (ex. Administrateur d'une application).</p> <p>2 - Identifier tous les cas d'utilisation en prenant comme base ceux définis dans le DCU système de l'activité précédente. Ceux-ci sont souvent décomposés à un niveau plus détaillé. Les cas d'utilisation informatiques apparaissent à ce niveau de l'analyse.</p> <p>3 - Représenter les interactions entre les acteurs et les cas d'utilisation.</p> <p>4 - Définir les dépendances entre les cas d'utilisations.</p>	

Tableau 13 : Fiche Guide 7 pour UP7 (d'après (Gabay & Gabay, 2008))

On se retrouve ici à affiner les DCU système précédemment élaborés (voir la sous-section **L'élaboration du diagramme des cas d'utilisation système**). Les cas d'utilisation décrits précédemment nous semblent assez détaillés pour ne pas faire l'objet d'une décomposition. Nous allons cependant préciser la gestion et l'utilisation qui peuvent être faites des profils d'utilisateurs. On peut d'ailleurs noter que ces différentes fonctionnalités seront accessibles aussi bien du côté serveur que de celui client. Il est à noter que, du côté du serveur, un profil de patient sera vu comme un dossier patient. Pour des raisons de confidentialité, la totalité du profil d'un patient ne sera évidemment pas accessible à tous les praticiens. A ce titre, on pourra également signaler que certaines des opérations sur les profils d'utilisateurs pourront être soumises à l'approbation d'un administrateur du système. Cet utilisateur particulier, dont nous n'avons pas encore parlé, a pour rôle de s'assurer que la sécurité du système, ainsi que des données qu'il contient, n'est pas compromise par des utilisateurs qui ne devraient pas y avoir accès. Un tel administrateur peut appartenir à une PAMMA particulière ou au réseau de PAMMA.

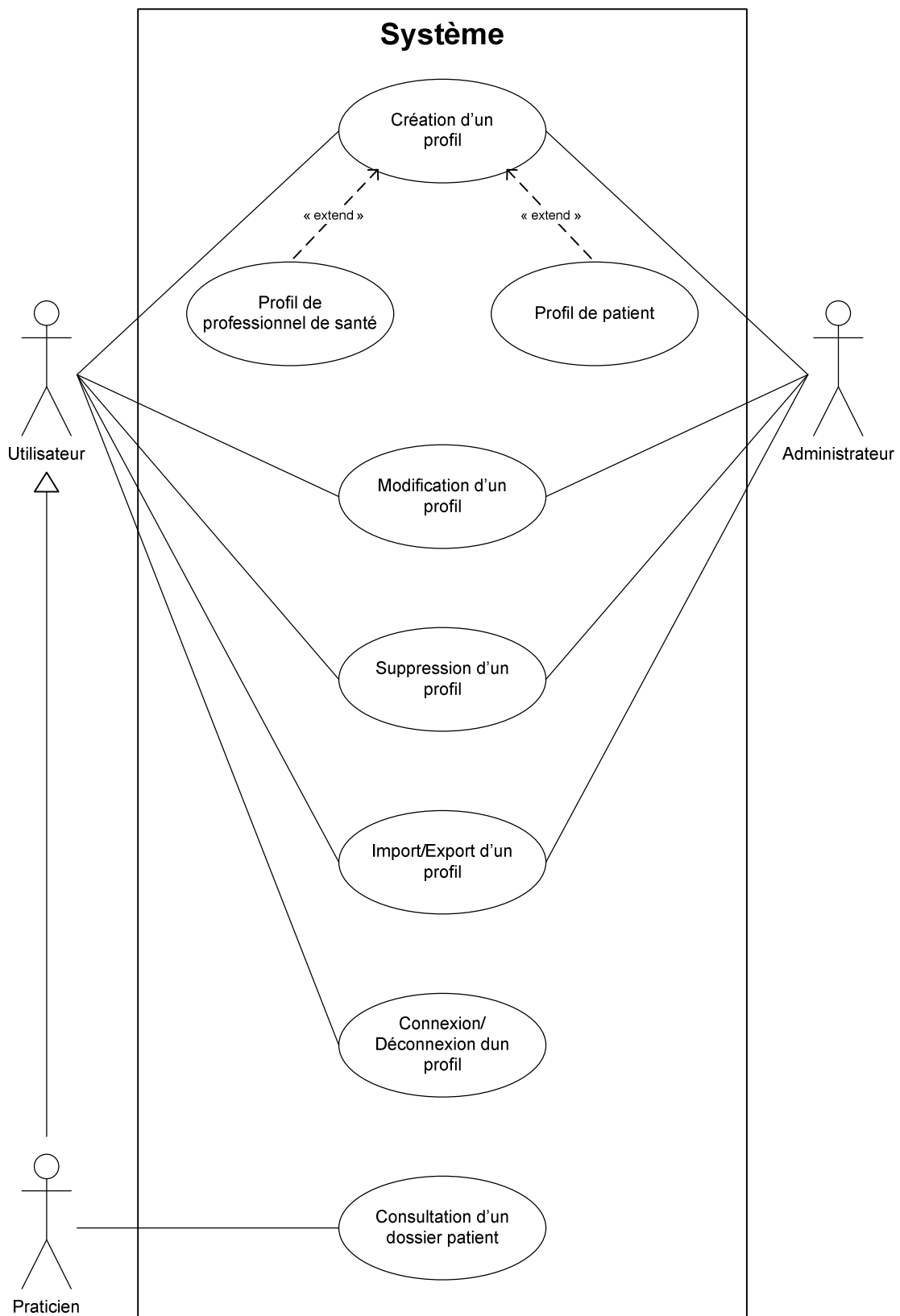


Figure 63 : Diagramme des cas d'utilisation de gestion des profils

Listons les différents cas d'utilisation informatiques de gestion des profils (**Figure 63**) :

- « **Création d'un profil** » – Il s'agit de la création d'un nouveau profil d'utilisateur du système. Un tel profil est d'ailleurs requis pour pouvoir accéder à ce dernier. On peut distinguer deux types de profils qui auront chacun leur mode de création propre : ceux des professionnels de santé et ceux des patients.
- « **Modification d'un profil** » – Après création d'un profil, il sera toujours possible de le modifier afin de prendre en compte des changements de paramètres ou certaines préférences de l'utilisateur.
- « **Suppression d'un profil** » – Si un profil ne sert plus, il est possible de le supprimer.
- « **Import/Export d'un profil** » – Il sera possible de sauvegarder un profil sous la forme d'un fichier ou par synchronisation avec un espace de stockage distant. De la même manière, une telle sauvegarde pourra être utilisée pour créer un nouveau profil ou mettre à jour un déjà existant.
- « **Connexion/Déconnexion d'un profil** » – Dans le cadre d'une utilisation normale du dispositif, l'utilisateur est amené à choisir le profil qu'il utilisera. Après utilisation de ce dernier, l'utilisateur devra s'en déconnecter pour, par exemple, décider d'utiliser un autre profil ou pour éteindre totalement ou partiellement le dispositif.
- « **Consultation d'un dossier patient** » – Le praticien doit pouvoir consulter les différents dossiers de ses patients (et seulement les dossiers de ses patients). Chaque dossier contiendra un certain nombre de données auxquelles le praticien aura accès en totalité ou partie. Les descriptions de ces privilèges de consultation des dossiers sont stockées dans le profil d'utilisateur du praticien.

Après s'être connecté à son profil, c'est-à-dire s'être authentifié, l'utilisateur va se trouver dans un mode offline. Il ne manque plus maintenant qu'à définir comment le système va lui permettre de passer en mode online. Ceci a pour but de permettre le travail entre client et serveur de manière synchrone ou asynchrone (voir la sous-section **L'élaboration du schéma de contexte du domaine**). Ainsi, il nous faut prévoir les cas d'utilisation suivants (**Figure 64**) :

- « **Connexion client-serveur** » – C'est la possibilité pour un utilisateur, qui utilise le système (via le client ou le serveur), de se connecter par le réseau à un autre utilisateur.
- « **Déconnexion client-serveur** » – Il s'agit pour des utilisateurs, entre lesquels une connexion est établie, de mettre fin à cette dernière.

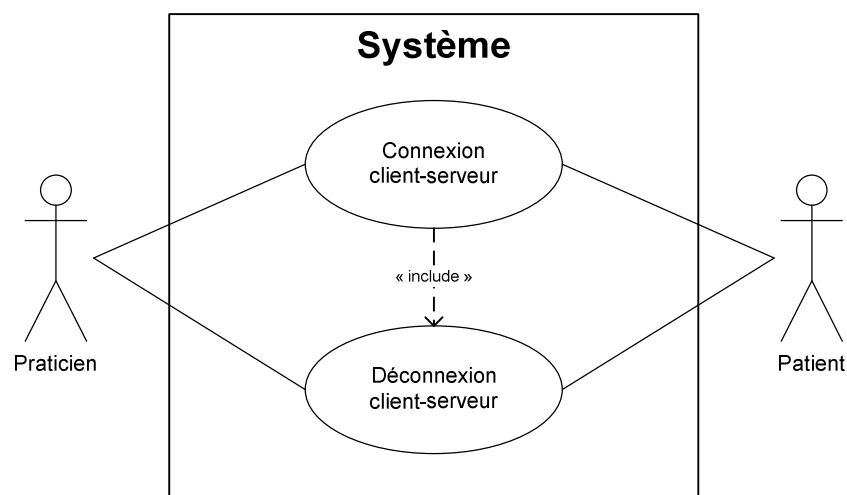


Figure 64 : Diagramme des cas d'utilisation des connexions client-serveur

2) La description des cas d'utilisation

FICHE GUIDE – FG8	
Activité 3 : Analyse des cas d'utilisation	
Sous-activité 3.2 : Description des cas d'utilisation (CU)	
Objectif	Décrire de manière textuelle et détaillée les cas d'utilisation
Point de départ	Tous les cas d'utilisation définis
Point d'arrivée	Cas d'utilisation décrits de manière textuelle
Démarche d'élaboration	
Pour chaque cas d'utilisation (CU) :	
<p>1 - Identifier l'objectif du CU.</p> <p>2 - Identifier les acteurs du CU à partir du DCU de la sous-activité précédente.</p> <p>3 - Définir les préconditions et éventuellement post conditions du CU.</p> <p>4 - Décrire le scénario nominal sous forme de liste d'actions avec une numérotation chronologique (1- Action 1, 2- Action 2...). L'objectif du CU doit être atteint au terme du scénario.</p> <p>5 - Décrire les scénarii alternatifs sous forme de liste d'actions avec une numérotation chronologique le reliant au scénario nominal (1a : titre 1, Cas alternatif « a » de l'étape 1 du scénario nominal). Les scénarii alternatifs doivent s'achever par la satisfaction ou l'abandon de l'objectif.</p>	

Tableau 14 : Fiche Guide 8 pour UP7 (d'après (Gabay & Gabay, 2008))

Nous allons maintenant analyser l'ensemble des vingt cas d'utilisation que nous avons identifiés jusqu'ici (voir les sous-sections **L'élaboration du diagramme des cas d'utilisation système** et **L'élaboration du diagramme des cas d'utilisation**) :

- Cas 1 - « Définir des processus de soins »
- Cas 2 - « Définir des épreuves »
- Cas 3 - « Réaliser un bilan »
- Cas 4 - « Passer les tests »
- Cas 5 - « Analyse et conclusion »
- Cas 6 - « Calibrage de la simulation »
- Cas 7 - « Planifier le programme de remédiation »
- Cas 8 - « Suivre le programme »
- Cas 9 - « Simuler le suivi du programme »
- Cas 10 - « Analyser les résultats »
- Cas 11 - « Ajuster la remédiation »
- Cas 12 - « Recalibrage de la simulation »
- Cas 13 - « Création d'un profil »
- Cas 14 - « Modification d'un profil »

- Cas 15 - « Suppression d'un profil »
- Cas 16 - « Import/Export d'un profil »
- Cas 17 - « Connexion/Déconnexion d'un profil »
- Cas 18 - « Consultation d'un dossier patient »
- Cas 19 - « Connexion client-serveur »
- Cas 20 - « Déconnexion client-serveur »

En premier lieu, nous allons détailler et décrire textuellement chacun de ces cas d'utilisation. A ce titre, nous en précisons pour chacun d'entre eux :

- L'**Objectif**, c'est-à-dire le contexte et les résultats attendus du cas d'utilisation.
- Les **Acteurs Concernés**, à savoir les utilisateurs concernés par ce cas.
- Les **Préconditions**, qui peuvent être assimilées à des contraintes de précédence indispensables à l'exécution de ce cas.
- Les **Post conditions**, qui rassemblent les contraintes devant être réunies après l'exécution du cas.
- Le **Scenarion nominal**, c'est-à-dire le scénario principal qui doit se dérouler sans incident et qui permet d'aboutir au résultat souhaité.
- Les **Scenarii alternatifs**, qui constituent les scénarii secondaires ou ceux qui correspondent à la résolution d'anomalies.

Cas 1 - « Définir des processus de soins » :

- **Objectif** : Permettre à l'utilisateur de paramétrer et agréger des épreuves, afin de définir des types d'arbres diagnostiques ou des programmes de remédiation.
- **Acteurs concernés** : Un professionnel de santé (praticien ou chercheur).
- **Préconditions** : L'utilisateur doit s'être authentifié et doit disposer de la partie "serveur" de la couche applicative du dispositif.
- **Post conditions** : Les processus créés peuvent être utilisés par les fonctionnalités d'exécution et analyse de résultats de processus de soins.
- **Scenarion nominal** :
 1. A partir de l'écran principal, l'utilisateur démarre l'éditeur.
 2. L'utilisateur opte pour la création d'un nouveau processus de soins.
 3. Il choisit des épreuves parmi celles mises à disposition.
 4. Il paramètre les propriétés de chaque future instance de ces épreuves.
 5. Il définit des relations entre ces différentes briques.
 6. Il fixe les paramètres d'entrée qu'acceptera ce processus de soins.
 7. Finalement, il définit la forme du rapport qui sera présenté en fin d'exécution d'un tel processus de soins.
 8. Le résultat est alors enregistré sous la forme d'un nouveau processus de soins qui sera disponible soit dans la partie d'aide diagnostique soit dans la partie liée à l'environnement de remédiation.
- **Scenarii alternatifs** :
 - 2a. Il est possible de partir d'un processus existant, que l'on peut éditer directement ou dupliquer avant édition.
 - 4a. Les paramétrages et personnalisations ne sont pas obligatoires dans la mesure où des valeurs par défaut seront systématiquement proposées.
 - 8a. L'enregistrement n'est pas obligatoire, mais dans ce cas, les changements effectués par l'utilisateur seront perdus.

Cas 2 - « Définir des épreuves » :

- **Objectif** : Permettre à l'utilisateur de paramétrer et agréger des éléments de base, enfin de définir des types de tests ou d'exercices.
- **Acteurs concernés** : Un professionnel de santé (praticien ou chercheur).
- **Préconditions** : L'utilisateur doit s'être authentifié et doit disposer de la partie "serveur" de la couche applicative du dispositif.
- **Post conditions** : Les épreuves créées peuvent être utilisées par les fonctionnalités d'exécution et d'analyse de résultats d'épreuves.
- **Scenari nominal** :
 1. A partir de l'écran principal, l'utilisateur démarre l'éditeur.
 2. L'utilisateur opte pour la création d'une nouvelle épreuve.
 3. Il choisit des éléments de base parmi ceux mis à disposition.
 4. Il paramètre les propriétés de chaque future instance de ces éléments.
 5. Il définit des relations entre ces différentes briques.
 6. Il fixe les paramètres d'entrée qu'acceptera cette épreuve.
 7. Finalement, il définit la forme des résultats qui seront présentés en fin d'exécution d'une telle épreuve.
 8. La nouvelle épreuve est enregistrée et sera disponible à l'intégration dans des processus de soins.
- **Scenarii alternatifs** :
 - 2a. Il est possible de partir d'une épreuve existante, que l'on peut éditer directement ou dupliquer avant édition.
 - 4a. Les paramétrages et personnalisations ne sont pas obligatoires dans la mesure où des valeurs par défaut seront systématiquement proposées.
 - 8a. L'enregistrement n'est pas obligatoire, mais dans ce cas, les changements effectués par l'utilisateur seront perdus.

Cas 3 - « Réaliser un bilan » :

- **Objectif** : Ce cas d'utilisation doit permettre au praticien de choisir quel arbre diagnostique exécuter.
- **Acteurs concernés** : Le praticien et le patient qui doivent se trouver dans un mode online.
- **Préconditions** : Tous les utilisateurs doivent être authentifiés et le patient, le client, doit s'être connecté au praticien, le serveur.
- **Post conditions** : Le système se retrouve dans le cas 4 - « Passer les tests ».
- **Scenari nominal** :
 1. A partir de l'écran principal, l'utilisateur clique sur la fonction relative aux bilans.
 2. En fonction de sa profession (et éventuellement d'autres préférences), le professionnel de santé doit choisir la batterie de tests, l'arbre diagnostique, à faire passer au patient.
 3. Un paramétrage manuel peut ensuite être précisé.
 4. Le professionnel de santé peut maintenant lancer l'exécution de l'arbre diagnostique sur le poste client.
- **Scenarii alternatifs** :
 - 3a. On peut choisir parmi des paramétrages rapides proposés par le système (paramétrages précédemment appliqués pour ce patient ou fréquemment utilisés par ce praticien, par exemple).

Cas 4 - « Passer les tests » :

- **Objectif** : Permettre au patient de réaliser la batterie de tests sous la surveillance du praticien.
- **Acteurs concernés** : Le praticien et le patient qui doivent se trouver dans un mode online.
- **Préconditions** : Le cas 3 - « Réaliser un bilan » doit venir d'être exécuté.
- **Post conditions** : Le système se retrouve dans le cas 5 - « Analyse et conclusion ».
- **Scenari nominal** : Pour chaque test à passer :
 1. Le praticien choisit quand le test commence.
 2. Le patient passe le test.
 3. Le système détermine quel est le test suivant à passer le cas échéant.
- **Scenarii alternatifs** :
 - Xa. A n'importe quel moment, le praticien peut choisir de mettre le test en pause, de le relancer, ou de réaliser toute autre action liée au pilotage de l'examen.
 - Xb. Pour assurer un meilleur pilotage, le client peut envoyer au serveur les résultats des tests au fur et à mesure qu'ils sont réalisés si cela est possible. On peut même envisager une sorte de streaming permettant une analyse quasiment en temps réel.

Cas 5 - « Analyse et conclusion » :

- **Objectif** : Offrir la possibilité au professionnel de santé d'analyser les résultats du bilan afin d'enrichir le dossier du patient de ses conclusions.
- **Acteurs concernés** : Le praticien et le patient qui doivent se trouver dans un mode online.
- **Préconditions** : Le cas 4 - « Passer les tests » doit venir d'être exécuté.
- **Post conditions** : Le système se retrouve éventuellement dans le cas 6 - « Calibrage de la simulation ».
- **Scenari nominal** :
 1. Le système agrège les données contenues dans les résultats des tests, les analyse et présente au praticien une aide diagnostique basée sur cette analyse sous la forme d'un rapport.
 2. Le praticien doit valider ou invalider les préconisations du système. Il n'est pas cantonné à une analyse macroscopique de ce que le système lui propose, mais il peut, par exemple, aller consulter des informations relatives à un des tests du bilan et visualiser les résultats que le patient y a obtenus.
 3. Le dossier du patient est ensuite enrichi des conclusions du professionnel de santé.
- **Scenarii alternatifs** :
 - 2a. Le praticien peut aussi décider de pratiquer des examens supplémentaires. Auquel cas, le patient devra repasser des tests ou carrément un nouveau bilan.
 - 3a. Le praticien peut sélectionner quelles informations issues des bilans seront attachées au dossier du patient et il peut aussi ajouter toute information complémentaire.

Cas 6 - « Calibrage de la simulation » :

- **Objectif** : Permettre de simuler, avec des performances comparables, la réalisation des tests que vient de subir le patient. L'intérêt n'est évidemment pas de reproduire les tests qui viennent d'être passés, mais de pouvoir anticiper les résultats

qu'obtiendrait le patient sur d'autres tests sollicitant des compétences qui ont déjà été évaluées.

- **Acteurs concernés** : N/A
- **Préconditions** : Le cas 5 - « Analyse et conclusion » doit avoir été exécuté (pas nécessairement à l'instant).
- **Post conditions** : L'exécution du modèle de simulation et l'analyse des résultats qu'il produit sont disponibles.
- **Scenario nominal** :
 1. Le système établit un paramétrage du modèle de simulation à partir du dossier complet du patient.
 2. Le modèle devient alors utilisable par les fonctionnalités d'analyse temporelle du suivi du patient.
- **Scenarii alternatifs** :
 - 1a. Le praticien peut choisir de ne pas procéder à cette création du modèle.

Cas 7 - « Planifier le programme de remédiation » :

- **Objectif** : Ce cas d'utilisation doit permettre au praticien de choisir quel programme de remédiation il va faire suivre au patient.
- **Acteurs concernés** : Le praticien (et le patient en mode online pour l'étape 4 du scenario nominal).
- **Préconditions** : Les utilisateurs doivent être authentifiés et le dossier patient doit être en cours d'utilisation par le praticien (et le client doit s'être connecté au serveur pour l'étape 4 du scenario nominal).
- **Post conditions** : Le système pourra ensuite se retrouver dans le cas 8 - « Suivre le programme » (ainsi qu'en parallèle et optionnellement dans le cas 9 - « Simuler le suivi du programme »).
- **Scenario nominal** :
 1. A partir de l'écran principal, l'utilisateur choisit la fonction relative à la mise en place d'un programme de remédiation.
 2. En fonction de sa profession (et éventuellement d'autres préférences) ainsi que des éléments déjà présents dans le dossier du patient, le professionnel de santé doit choisir le programme de remédiation qu'il va faire suivre au patient.
 3. Un paramétrage manuel peut ensuite être précisé.
 4. Le professionnel de santé peut maintenant lancer la mise en place du programme de remédiation sur le poste client.
- **Scenarii alternatifs** :
 - 3a. On peut également choisir parmi des paramétrages rapides proposés par le système (paramétrages précédemment appliqués pour ce patient ou fréquemment utilisés par ce praticien, par exemple).

Cas 8 - « Suivre le programme » :

- **Objectif** : Permettre au patient de réaliser les exercices qui lui ont été prescrits.
- **Acteurs concernés** : Le patient (et éventuellement le praticien pour des exercices qui doivent se faire en mode online).
- **Préconditions** : Le cas 7 - « Planifier le programme de remédiation » doit avoir été exécuté (pas nécessairement à l'instant).
- **Post conditions** : Le système pourra ensuite se retrouver dans le cas 10 - « Analyser les résultats ».

- **Scenari nominal** : Pour chaque exercice à faire :
 1. Le patient fait l'exercice.
 2. Le système détermine s'il y a encore un exercice à faire.
- **Scenarii alternatifs** :
 - Xa. Pour assurer le suivi, le client peut envoyer au serveur les résultats des exercices au fur et à mesure qu'ils sont réalisés si cela est possible. On peut même envisager une sorte de streaming permettant une analyse quasiment en temps réel.

Cas 9 - « Simuler le suivi du programme » :

- **Objectif** : Permettre d'obtenir et de simuler des résultats aux exercices comparables à ceux que le patient aurait obtenus avec le niveau qu'il avait lors du dernier calibrage du modèle.
- **Acteurs concernés** : N/A
- **Préconditions** : Le cas 7 - « Planifier le programme de remédiation » doit avoir été exécuté (pas nécessairement à l'instant) ainsi que le cas 6 - « Calibrage de la simulation ».
- **Post conditions** : Le système pourra ensuite se retrouver dans le cas 10 - « Analyser les résultats ».
- **Scenari nominal** : Pour chaque exercice à faire :
 1. Le système simule la réalisation de l'exercice.
 2. Le système détermine s'il y a encore un exercice à faire.
- **Scenarii alternatifs** :
 - Xa. Les exercices ne seront pas forcément réalisés qu'une seule fois, en effet la simulation pourra être répliquée plusieurs fois afin d'obtenir des résultats simulés statistiquement cohérents.

Cas 10 - « Analyser les résultats » :

- **Objectif** : Le professionnel de santé doit pouvoir analyser les résultats du programme de remédiation afin de faire le suivi du patient.
- **Acteurs concernés** : Le praticien.
- **Préconditions** : Le cas 8 - « Suivre le programme » doit venir d'être exécuté (et éventuellement le cas 9 - « Simuler le suivi du programme »).
- **Post conditions** : Le système se retrouve dans le cas 11 - « Ajuster la remédiation » (ainsi qu'en parallèle et optionnellement dans le cas 12 - « Recalibrage de la simulation »).
- **Scenari nominal** :
 1. Le système agrège les données contenues dans les résultats des exercices, les analyse et présente au praticien un récapitulatif des performances obtenues lors de la réalisation par le patient des exercices (temps passé, taux de bonnes réponses, etc.).
 2. Le praticien doit évaluer l'efficacité du programme de remédiation. Il n'est pas limité à une analyse macroscopique de ce que le système lui propose, mais il peut, par exemple, aller consulter des informations relatives à un exercice et visualiser les résultats que le patient y a obtenus. Le praticien peut, en outre, comparer les résultats obtenus par le patient à ceux potentiellement générés par simulation. Cette comparaison permet une quantification des progrès du patient

3. Le dossier du patient est ensuite enrichi des conclusions du professionnel de santé.

- **Scenarii alternatifs :**

3a. Le praticien peut sélectionner quelles informations issues des résultats du programme de remédiation seront attachées au dossier du patient et il peut aussi ajouter toute information complémentaire.

Cas 11 - « Ajuster la remédiation » :

- **Objectif :** Permettre au praticien d'adapter la remédiation aux résultats déjà obtenus par le patient.

- **Acteurs concernés :** Le praticien.

- **Préconditions :** Le cas 10 - « Analyser les résultats » doit avoir été exécuté (pas nécessairement à l'instant).

- **Post conditions :** Le système se retrouve éventuellement dans le cas 12 - « Recalibrage de la simulation ».

- **Scenario nominal :**

1. Le praticien procède aux ajustements nécessaires (changement de programme ou de posologie des exercices, voire la réévaluation de l'état pathologique du patient).

2. Le dossier du patient est ensuite enrichi des nouveaux aménagements mis en place par le professionnel de santé.

- **Scenarii alternatifs :**

3a. Le praticien peut sélectionner quelles informations seront attachées au dossier du patient et il peut aussi ajouter toute information complémentaire.

Cas 12 - « Recalibrage de la simulation » :

- **Objectif :** Mettre à jour le modèle de simulation afin qu'il adhère au mieux au profil actuel du patient.

- **Acteurs concernés :** N/A

- **Préconditions :** Le cas 11 - « Analyser les résultats » doit avoir été exécuté (pas nécessairement à l'instant) ainsi que le cas 6 - « Calibrage de la simulation ».

- **Post conditions :** L'exécution d'un nouveau paramétrage du modèle de simulation et l'analyse des résultats qu'il produit sont disponibles.

- **Scenario nominal :**

1. Le système établit un paramétrage du modèle de simulation à partir du dossier complet du patient, ou intègre seulement les nouveaux ajouts depuis le dernier (re)calibrage.

2. Le modèle nouvellement paramétré devient alors utilisable par les fonctionnalités d'analyse temporelle du suivi du patient.

- **Scenarii alternatifs :**

1a. Le praticien peut choisir de ne pas procéder à cette mise à jour ou recréation du modèle.

Cas 13 - « Création d'un profil » :

- **Objectif :** Permettre à l'utilisateur de créer un profil permettant de l'authentifier et de conserver ses préférences et historiques d'utilisation du dispositif.

- **Acteurs concernés :** Tous les utilisateurs.

- **Préconditions :** Le logiciel doit être installé et activé.

- **Post conditions** : Il est ensuite possible d'utiliser le dispositif en s'authentifiant.
- **Scenarii nominal** :
 1. Dès le lancement du logiciel (avant l'écran principal), l'utilisateur opte pour la création d'un nouveau profil sur l'écran de gestion des profils.
 2. L'utilisateur saisit les informations de son profil (nom du profil, rôle vis-à-vis du système, méthode d'authentification, sexe, âge, etc.).
 3. Une fois les informations validées, le profil est créé.
- **Scenarii alternatifs** :
 - 2a. Toutes les informations ne sont pas forcément obligatoires, d'autre part, selon qu'il s'agit d'un profil destiné au serveur ou au client, les mêmes informations ne seront pas demandées.
 - 3a. La création peut être soumise à l'approbation d'un administrateur.

Cas 14 - « Modification d'un profil » :

- **Objectif** : Permettre à l'utilisateur de modifier les informations d'un profil.
- **Acteurs concernés** : Tous les utilisateurs.
- **Préconditions** : Le logiciel doit être installé et activé.
- **Post conditions** : Le profil mis à jour peut ensuite être utilisé.
- **Scenarii nominal** :
 1. L'utilisateur sélectionne un profil sur l'écran de gestion des profils.
 2. Il opte ensuite pour la modification de profil.
 3. L'utilisateur doit s'authentifier (avec les identifiants du profil cible).
 4. L'utilisateur saisit les nouvelles informations de son profil.
 5. Une fois les informations validées, le profil est mis à jour.
- **Scenarii alternatifs** :
 - 4a. Toutes les informations ne sont pas forcément obligatoires, d'autre part, selon qu'il s'agit d'un profil serveur ou client, les mêmes types d'informations ne seront pas présents.
 - 5a. La modification peut être soumise à l'approbation d'un administrateur.

Cas 15 - « Suppression d'un profil » :

- **Objectif** : Permettre à l'utilisateur de supprimer un profil.
- **Acteurs concernés** : Tous les utilisateurs.
- **Préconditions** : Le logiciel doit être installé et activé.
- **Post conditions** : Le profil disparaît de la liste des profils utilisables.
- **Scenarii nominal** :
 1. L'utilisateur sélectionne un profil sur l'écran de gestion des profils.
 2. Il opte ensuite pour la suppression de profil.
 3. L'utilisateur doit s'authentifier (avec les identifiants du profil cible).
 4. Une fois la suppression validée, l'ensemble des données du profil disparaît.
- **Scenarii alternatifs** : N/A
 - 4a. La suppression peut être soumise à l'approbation d'un administrateur.

Cas 16 - « Import/Export d'un profil » :

- **Objectif** : Permettre d'enregistrer ou de charger un profil dans son intégralité, respectivement, dans ou à partir d'un fichier.
- **Acteurs concernés** : Tous les utilisateurs.
- **Préconditions** : Le logiciel doit être installé et activé.

- **Post conditions** : Dans le cas d'une importation, le profil apparaît maintenant dans la liste des profils utilisables. Dans celui d'une exportation, l'utilisateur disposera d'une copie statique (locale ou distante) du profil (permettant une éventuelle restauration ou une utilisation multiposte du profil).
- **Scenario nominal** :
 1. L'utilisateur sélectionne un profil sur l'écran de gestion des profils.
 2. Il opte ensuite pour l'exportation de ce profil.
 3. L'utilisateur doit s'authentifier (avec les identifiants du profil cible).
 4. Une fois l'exportation effectuée, le profil est sauvegardé dans un fichier (pour une exportation locale).
- **Scenarii alternatifs** :
 - 1a. Dans le cas d'une importation, il n'y a pas besoin de sélectionner un profil, mais un fichier (pour une exportation locale)
 - 4a. Dans le cas d'une exportation distante, la sauvegarde sera directement envoyée sur un serveur accessible via internet.
 - 4b. Dans le cas de l'importation, le profil est maintenant disponible dans la liste des profils.
 - 4c. L'importation d'un profil peut être soumise à l'approbation d'un administrateur.

Cas 17 - « Connexion/Déconnexion d'un profil » :

- **Objectif** : Permettre à l'utilisateur d'utiliser un profil.
- **Acteurs concernés** : Tous les utilisateurs.
- **Préconditions** : Le logiciel doit être installé et activé.
- **Post conditions** : On peut ensuite utiliser toutes les fonctionnalités et privilèges accessibles à ce profil (selon le type du logiciel, client ou serveur, les informations saisies par l'utilisateur à la création du profil, et les autorisations accordées par l'administrateur du système).
- **Scenario nominal** :
 1. L'utilisateur sélectionne un profil sur l'écran de gestion des profils.
 2. Il opte ensuite pour démarrer le logiciel en utilisant ce profil.
 3. L'utilisateur doit s'authentifier (avec les identifiants du profil sélectionné).
- **Scenarii alternatifs** :
 4. Par la suite, pour se déconnecter, l'utilisateur pourra sélectionner une option de déconnexion ou simplement fermer l'application (et confirmer son choix).

Cas 18 - « Consultation d'un dossier patient » :

- **Objectif** : Permettre au praticien de consulter les dossiers de ces différents patients conformément à ses privilèges.
- **Acteurs concernés** : Le praticien.
- **Préconditions** : L'utilisateur doit être authentifié.
- **Post conditions** : N/A
- **Scenario nominal** :
 1. A partir de l'écran principal, l'utilisateur choisit la fonction relative au suivi des dossiers de ses patients.
 2. Le praticien spécifie ensuite le patient dont il veut accéder au dossier.
 3. Le professionnel de santé peut maintenant consulter le dossier du patient (qui est enrichi par tous les professionnels de santé qui le suivent), le modifier, réaliser des simulations (si le modèle est déjà initialisé), etc..

- **Scenarii alternatifs :**

- 3a. Certaines données peuvent être à accès restreint : seul un ou plusieurs praticiens peuvent les visualiser, ou les modifier (conformément au système de droits mis en place par l'administrateur).

Cas 19 - « Connexion client-serveur » :

- **Objectif** : Permettre à deux utilisateurs/systèmes d'établir une connexion entre eux.
- **Acteurs concernés** : Le praticien et le patient.
- **Préconditions** : Les utilisateurs doivent être authentifiés.
- **Post conditions** : A l'initiative d'un des acteurs, la connexion pourra être clôturée tel que présenté par le cas 20 - « Déconnexion client-serveur ».
- **Scenario nominal** :
 1. A partir de l'écran principal, l'utilisateur choisi de gérer les connexions et de tenter de se connecter à une autre PAMMA. Une demande de connexion est ainsi envoyée via le réseau d'une PAMMA à l'autre.
 2. La PAMMA qui reçoit la demande présente cette dernière à son utilisateur.
 3. L'utilisateur accepte la demande et sa PAMMA met à jour ses informations concernant cette nouvelle connexion.
 4. La PAMMA à l'initiative de la connexion est notifiée et met, le cas échéant, à jour ses informations au sujet de cette connexion.
 5. Ayant demandé la connexion, l'utilisateur est notifié du résultat de sa requête.
- **Scenarii alternatifs** :
 - 3a. L'utilisateur refuse la demande.

Cas 20 - « Déconnexion client-serveur » :

- **Objectif** : Permettre à un utilisateur de mettre fin à une connexion établie avec un autre utilisateur.
- **Acteurs concernés** : Le praticien et le patient.
- **Préconditions** : Les utilisateurs doivent être authentifiés, et une connexion doit être établie entre eux.
- **Post conditions** : La connexion, dont l'état est enregistré par le serveur et le client, est clôturée par ces derniers.
- **Scenario nominal** :
 1. A partir de l'écran principal, un des utilisateurs choisit de mettre fin à une connexion en cours dans le gestionnaire des connexions. Sa PAMMA met ainsi à jour ses informations concernant cette déconnexion.
 2. L'autre PAMMA concernée par la déconnexion est informée de la déconnexion et met également à jour ses informations de connexion.
 3. L'utilisateur de cette PAMMA est alors notifié de la déconnexion.
- **Scenarii alternatifs** : N/A

3) L'élaboration des diagrammes de séquence

FICHE GUIDE – FG9	
Activité 3 : Analyse des cas d'utilisation	
Sous-activité 3.3 : Elaboration des diagrammes de séquence (DSE)	
Objectif	Représenter les interactions entre les acteurs et les objets du système pour tous les cas d'utilisation en considérant les différents scénarii décrits
Point de départ	Cas d'utilisation décrits de manière textuelle et concepts métiers identifiés et définis
Point d'arrivée	Interactions entre acteurs et objets du système décrits dans le DSE pour tous les cas d'utilisation
Démarche d'élaboration	
Pour chaque cas d'utilisation (CU) et chaque scénario identifié :	
<ol style="list-style-type: none"> 1 - Reporter le ou les acteurs du DCU impliqués dans le scénario. 2 - Représenter les objets du système impliqués dans le scénario en prenant comme base ceux identifiés lors de l'activité de modélisation métier. 3 - Identifier les messages (synchrones/asynchrones) entre les acteurs et les objets et entre les objets eux-mêmes. La chronologie des échanges doit être respectée. 4 - Modéliser les fragments d'interactions si nécessaire. 	

Tableau 15 : Fiche Guide 9 pour UP7 (d'après (Gabay & Gabay, 2008))

Avant de détailler les vingt cas utilisation que nous venons de présenter, il paraît judicieux de s'intéresser aux préconditions que certains requièrent explicitement. En effet, il existe une sorte de cas d'utilisation factuels, ne représentant pas nécessairement un intérêt d'un point de vue métier, qui sont préalablement nécessaires à l'accomplissement d'autres cas d'utilisation. Ainsi, nous avons identifié les préconditions suivantes :

- Installation et activation du système (**Figure 65**)
- Authentification de l'utilisateur (**Figure 66**)
- Accéder à un dossier patient (**Figure 67**)

D'autre part, il est également pertinent de préciser dès maintenant l'ensemble des stéréotypes que nous allons utiliser pour nos DSE :

- <<Matériel>>, qui représente les éléments électroniques ou mécaniques utilisés.
- <<Système>>, qui désigne la PAMMA indépendamment du fait qu'il s'agisse du client ou du serveur.
- <<Système de l'éditeur>>, qui est la partie du système entièrement sous la responsabilité de l'éditeur (c'est-à-dire nous-mêmes) permettant la vérification des licences des produits ou encore distribuant leur mise à jour.
- <<Données>>, qui désigne des données statiques utilisées par le système.

La première des préconditions, l'installation et l'activation du système, va nécessiter les actions suivantes (**Figure 65**) :

- Une intervention locale sur l'ordinateur destiné à héberger l'instance de la PAMMA.
- Une intervention distante permettant l'activation de l'instance de la PAMMA par l'éditeur (c'est-à-dire nous-mêmes).

Ces interventions visent à s'assurer que la PAMMA est correctement installée/configurée et qu'il ne s'agit pas d'une version piratée ou modifiée. Consécutivement à cela, la PAMMA est prête à rejoindre un réseau de confiance de PAMMA.

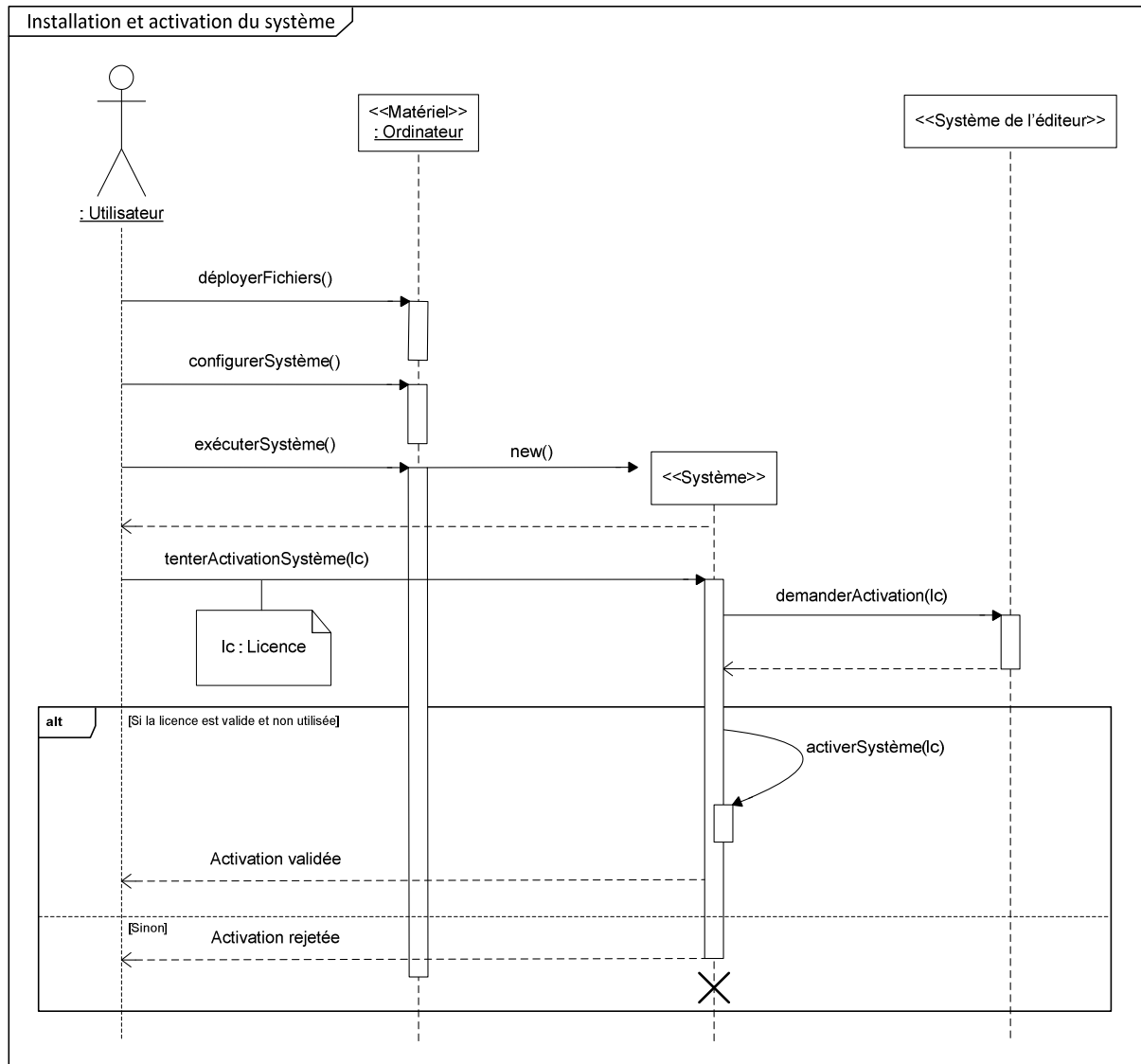


Figure 65 : Diagramme de séquence de la précondition aux cas d'utilisation « Installation et activation du système »

L'authentification correspond à fournir au système les identifiants, c'est-à-dire au moins un nom d'utilisateur et un mot de passe, qui correspondent à un profil. Le système peut alors vérifier si ces identifiants sont ceux d'un profil, et le cas échéant accorder à l'utilisateur les privilèges et fonctionnalités disponibles pour ce profil (**Figure 66**).

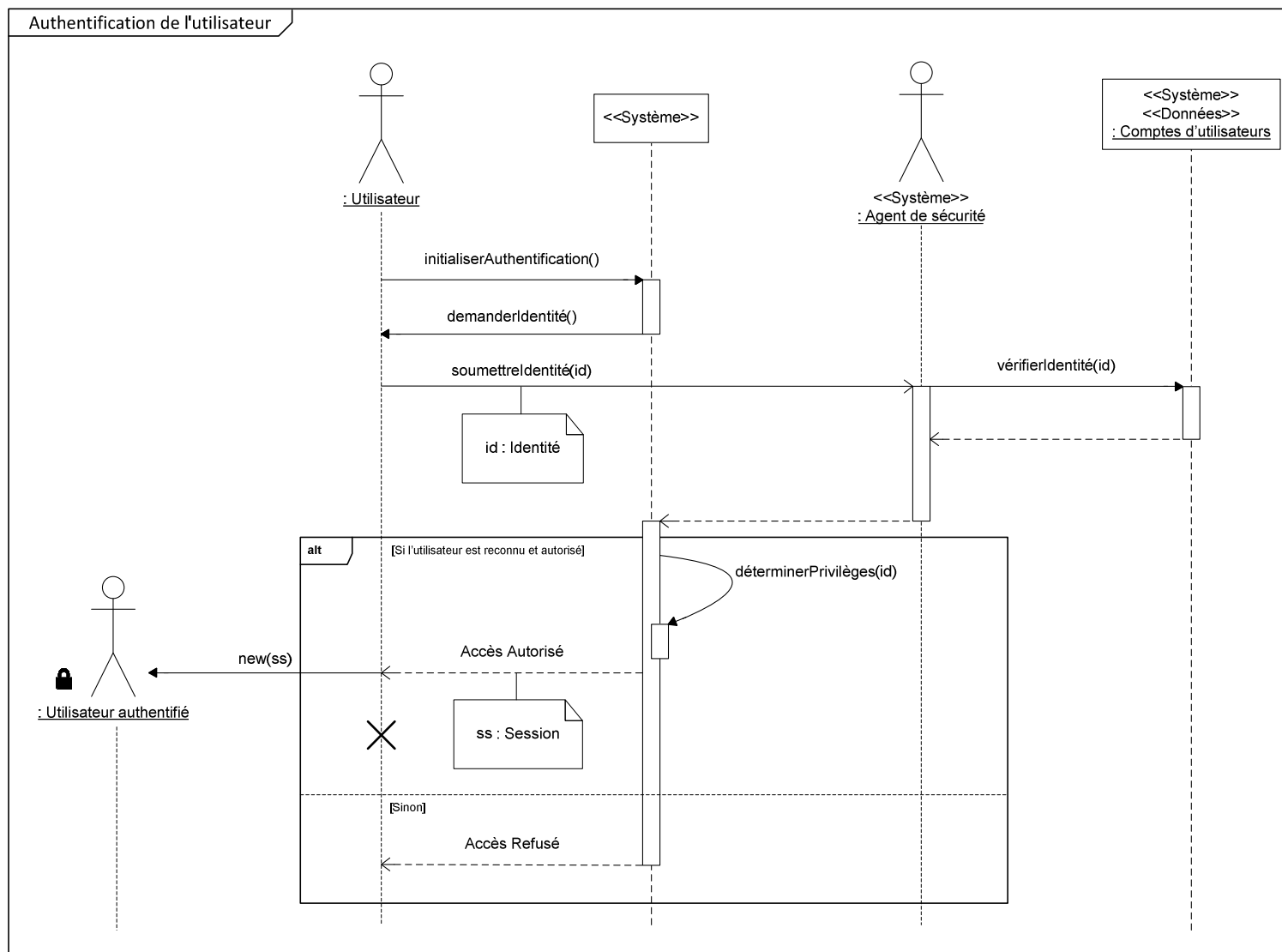


Figure 66 : Diagramme de séquence de la précondition aux cas d'utilisation « Authentification de l'utilisateur »

Ainsi, après s'être assuré que la PAMMA remplit nos impératifs de sécurité, nous pouvons vérifier que l'utilisateur est lui aussi de confiance. Comme représenté sur la **Figure 66**, les utilisateurs authentifiés sont représentés accompagnés d'une icône de cadenas (**Figure 43**). Ainsi, dans la suite de ce document, les utilisateurs exhibant une telle iconographie seront considérés comme bénéficiant d'une session, qui est valide et en cours d'exécution, sur la PAMMA qu'ils utilisent. Cette session est garante de leur identité et permet un échange sécurisé entre l'utilisateur et le système. Le praticien peut ainsi accéder aux dossiers de ses patients de manière sécurisée (**Figure 67**).

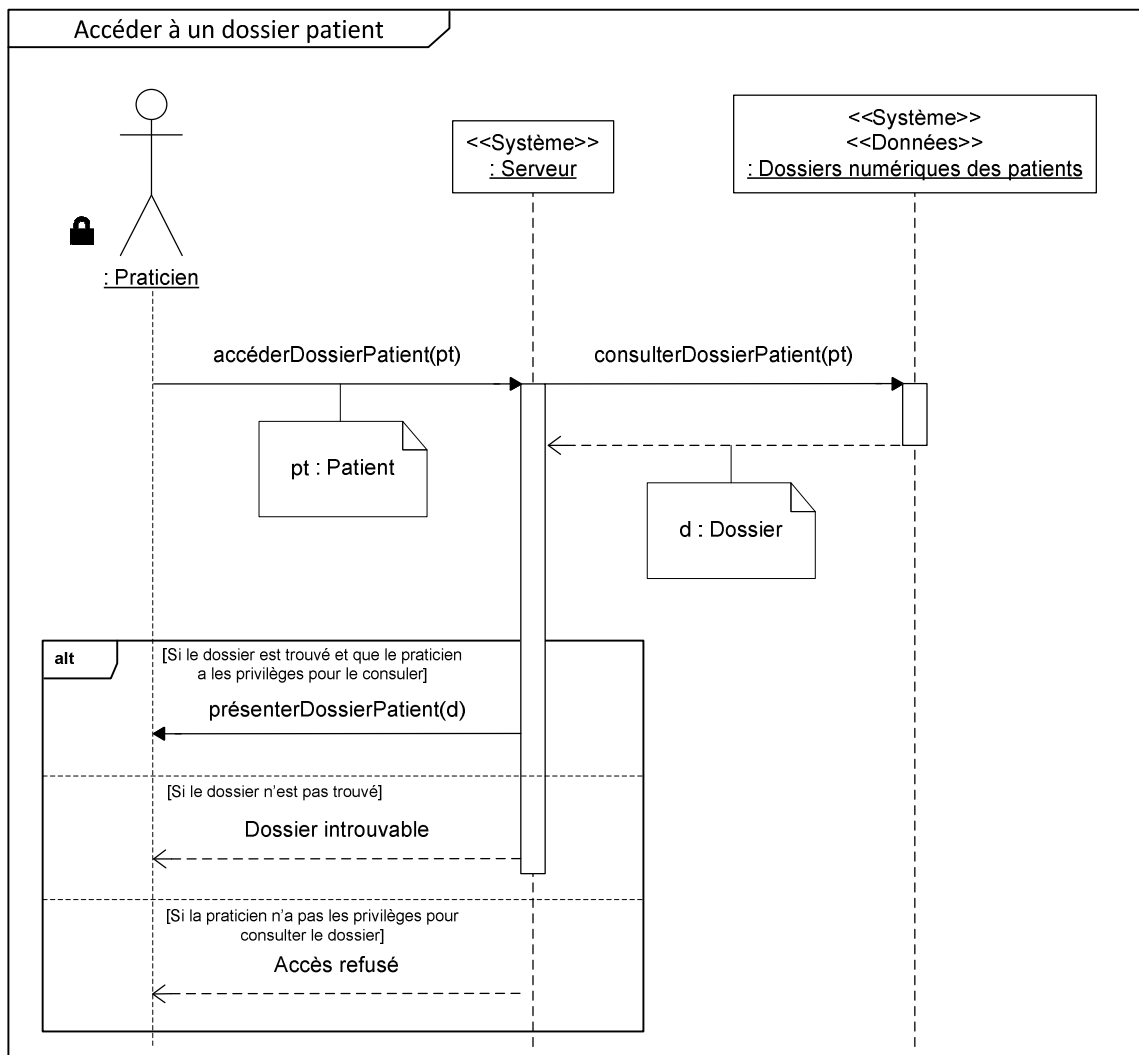


Figure 67 : Diagramme de séquence de la précondition aux cas d'utilisation « Accéder à un dossier patient »

Nous allons maintenant pouvoir présenter sous la forme de diagrammes de séquence l'ensemble des cas d'utilisation que nous avons précédemment identifié. Par ailleurs, nous signalons que chacun de ces cas d'utilisation peut être vu comme une véritable application pour la PAMMA (sauf ceux liés à l'utilisation de l'éditeur). A ce titre, chaque diagramme qui va suivre décrit le fonctionnement à haut niveau d'une application qu'il faudra implémenter, dans le cadre du traitement de la dyslexie, à l'aide d'agents sur la PAMMA.

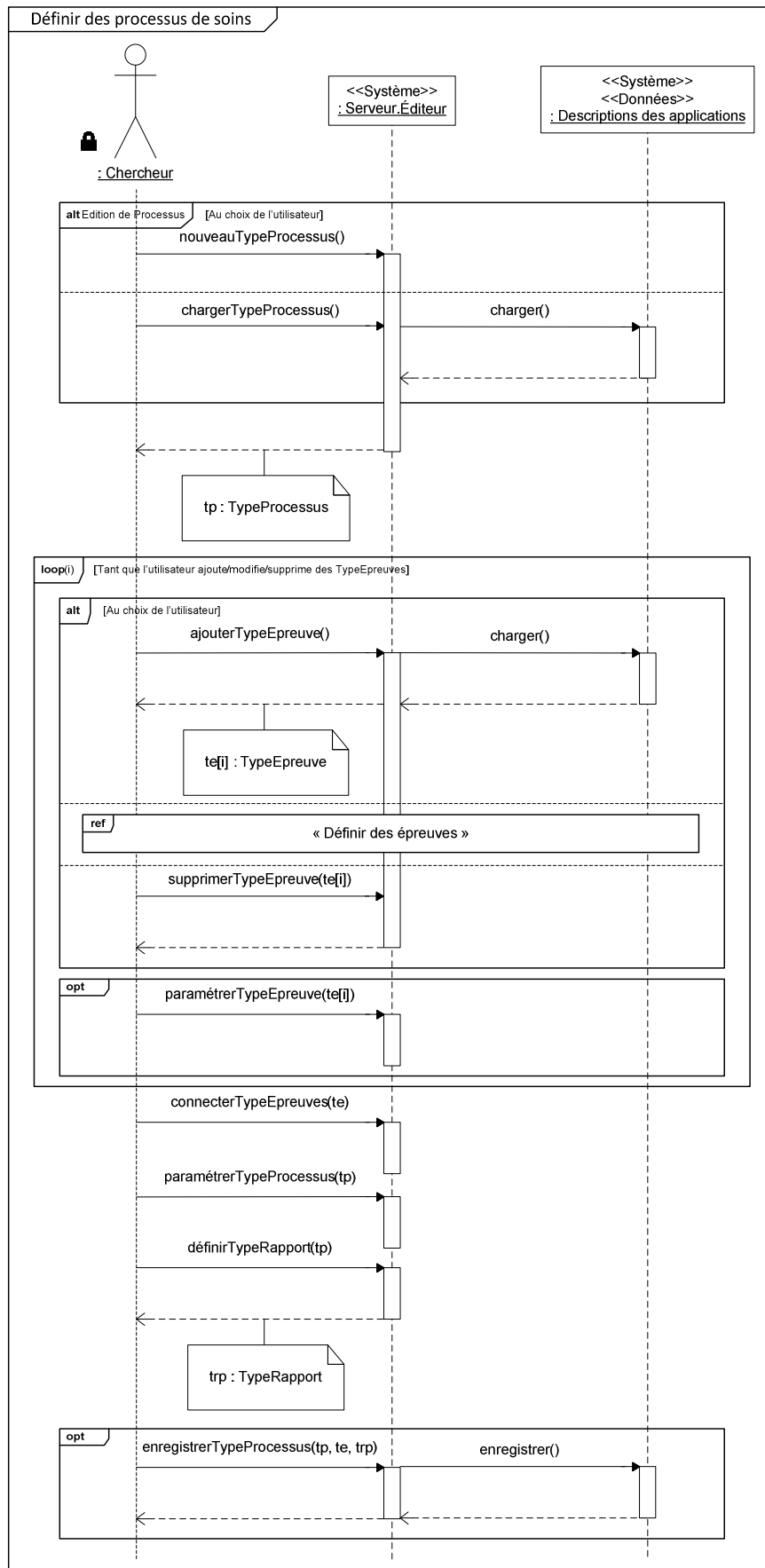


Figure 68 : Diagramme de séquence du cas d'utilisation 1 « Définir des processus de soins »

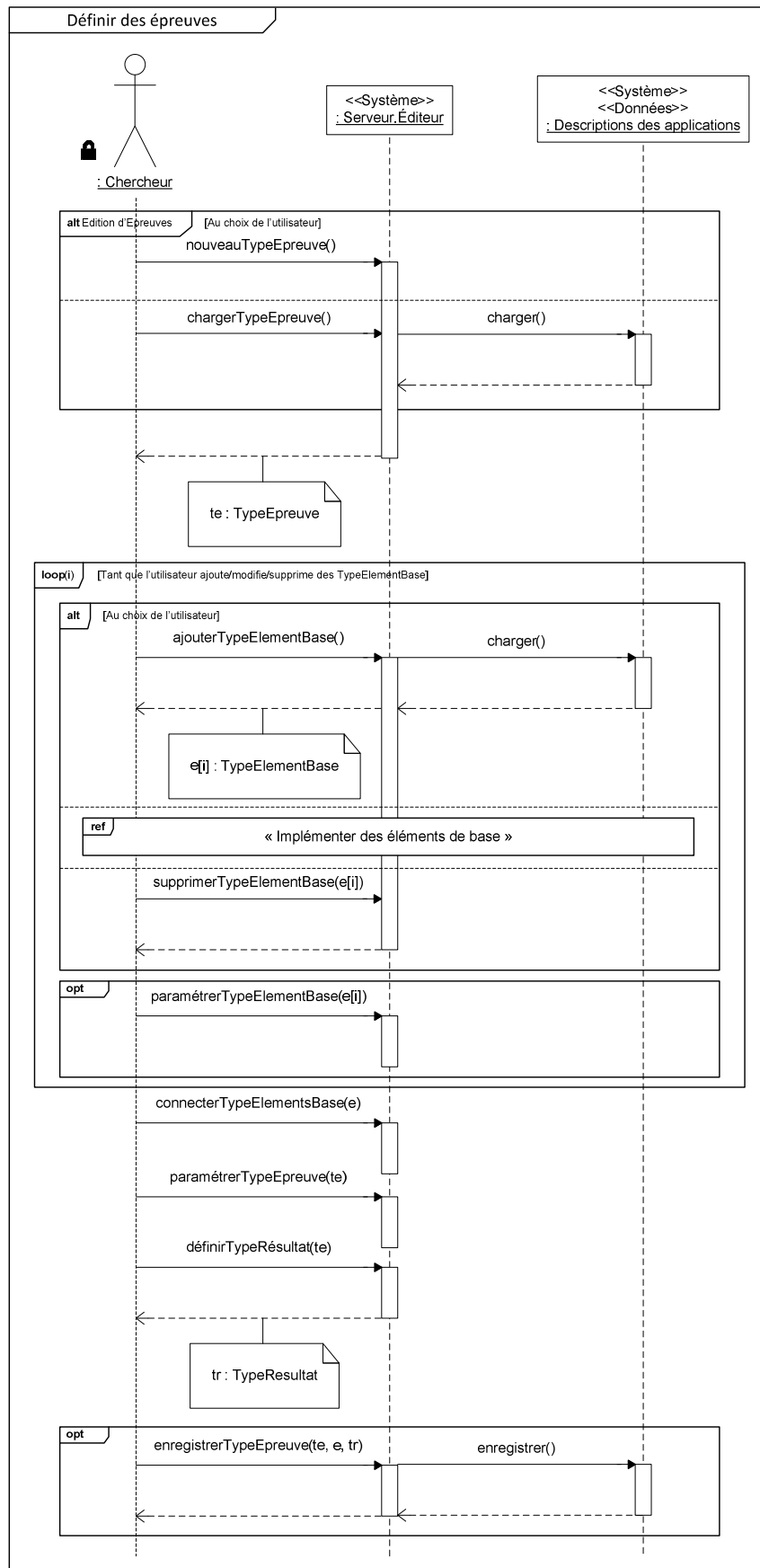


Figure 69 : Diagramme de séquence du cas d'utilisation 2 « Définir des épreuves »

La définition des processus de soins (**Figure 68**) ainsi que celle des épreuves (**Figure 69**) consistent à intégrer des processus métiers (voir la section **L'intégration des processus métiers**). En effet, cette étape de spécification revient à modéliser des processus cliniques impliqués dans la prise en charge des dyslexiques afin de les informatiser. Evidemment, l'éditeur peut également servir à modéliser d'autres applications que celles s'adressant uniquement au traitement de la dyslexie. Chaque processus ainsi modélisé est une application à part entière pour la PAMMA. Nous rappelons que l'ensemble des applications pour la dyslexie que nous présentons ici est générique. En effet, notre but est de décrire un cadre générique de prise en charge dans lequel des applications spécifiques, pour le diagnostic ou la remédiation, pourront par la suite être réalisées.

Ici, nous avons choisi de ne pas détailler l'implémentation des éléments de base, car cela revient à coder de tels composants. Cette phase de programmation ne peut être réalisée que par un informaticien et, dans cette partie, nous nous concentrons uniquement sur les parties du système s'adressant aux utilisateurs que sont les professionnels de santé et les patients. D'autre part, nous signalons, bien que nous ne l'ayons pas fait apparaître au travers d'un diagramme, que l'utilisateur de l'éditeur a également la possibilité de supprimer un processus de soins ou une épreuve. Une telle suppression revient, par exemple, à simplement retirer l'élément ciblé de la base de données de description des applications.

Dans le cadre de la réalisation de bilans (**Figure 70**), le système commence par présenter une liste des applications que l'utilisateur peut exécuter. Cette liste contient des métadonnées de ces différentes applications. Chacune de ces applications est un type d'arbre diagnostique, c'est-à-dire un squelette d'application spécifique à une technique de diagnostic. Le praticien va pouvoir choisir quelle application doit être instanciée et en préciser les paramètres.

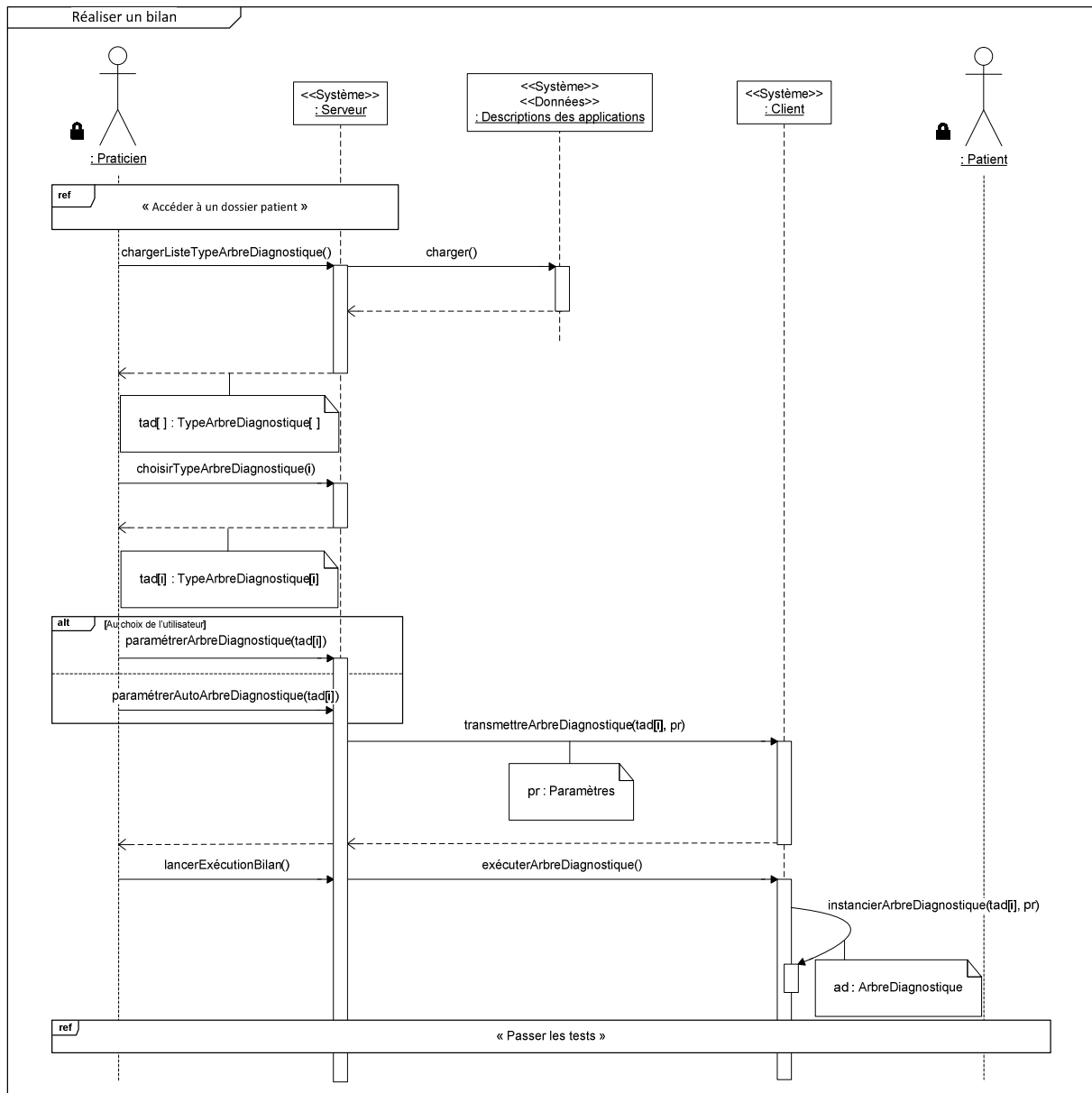


Figure 70 : Diagramme de séquence du cas d'utilisation 3 « Réaliser un bilan »

Une fois l'application instanciée sur le client, le patient peut commencer à passer les tests définis par l'arbre diagnostique (Figure 71). Les tests sont ainsi successivement instanciés et ils produisent chacun un résultat une fois qu'ils sont achevés.

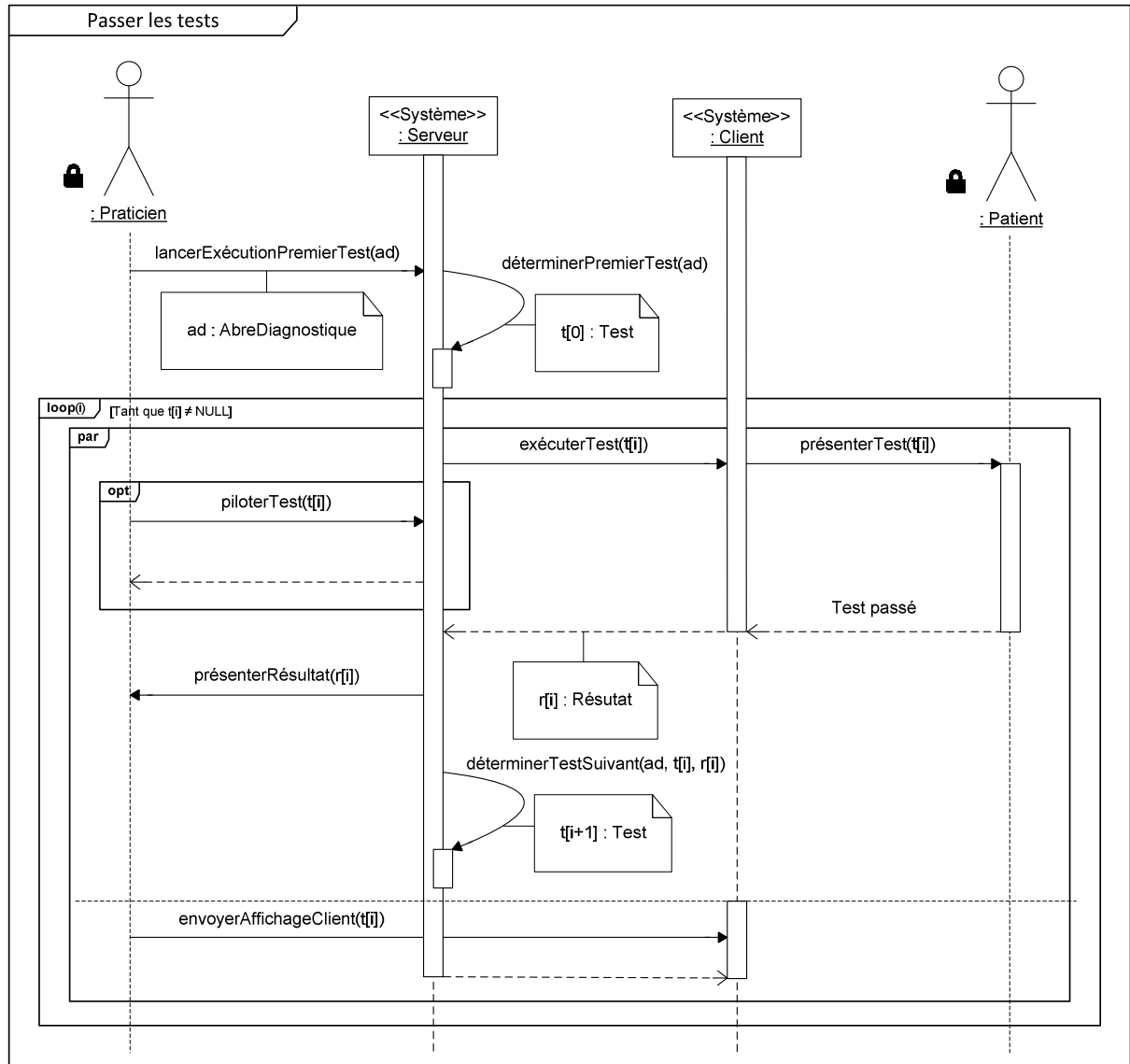


Figure 71 : Diagramme de séquence du cas d'utilisation 4 « Passer les tests »

Une fois tous les résultats générés, un rapport peut être agrégé/constitué. Ce rapport va être une véritable aide à la décision pour le praticien afin de faciliter son analyse du bilan (**Figure 72**). Lorsque le diagnostic est finalement établi par le praticien, il est intégré au rapport puis ajouté au DMP du patient.

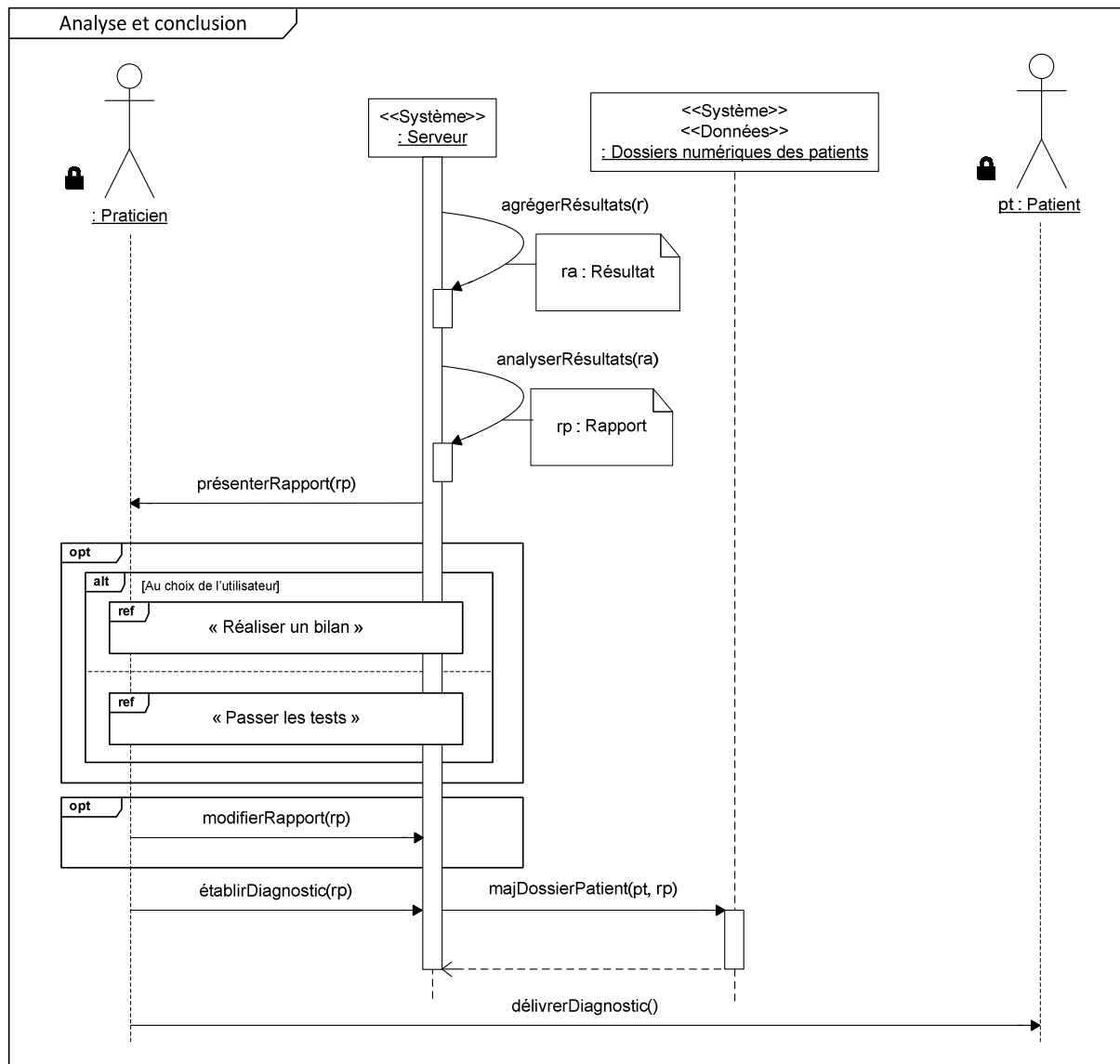


Figure 72 : Diagramme de séquence du cas d'utilisation 5 « Analyse et conclusion »

Si le DMP d'un patient contient au moins un rapport de bilan, un modèle de simulation peut être calibré (**Figure 73**). Ce modèle va avoir pour but la reproduction des performances déjà observées chez le patient. Ainsi, il sera possible de simuler le comportement de ce dernier pour la réalisation d'exercices de remédiation et de comparer les résultats obtenus par simulation à ceux réellement obtenus par le patient sur ces mêmes exercices.

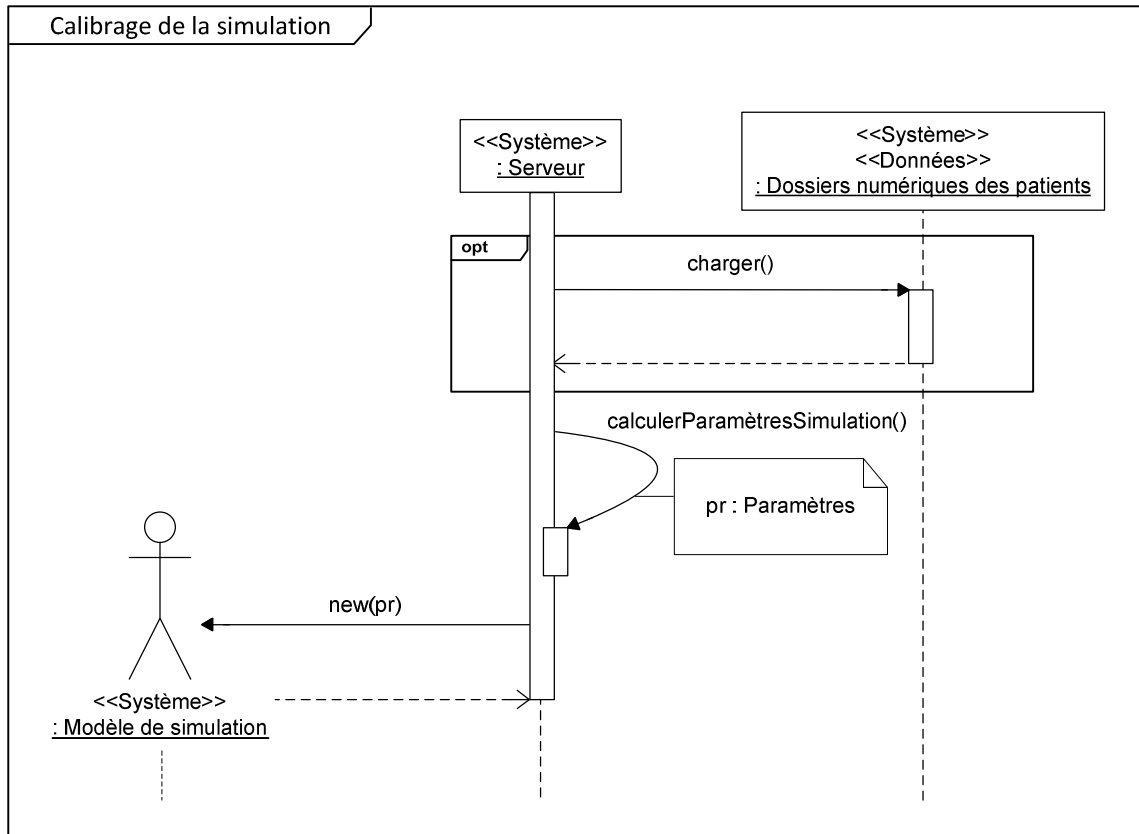


Figure 73 : Diagramme de séquence du cas d'utilisation 6 « Calibrage de la simulation »

Dans le cadre de la planification du programme de remédiation (**Figure 74**), le système commence par présenter une liste des applications que l'utilisateur peut exécuter. Comme pour la réalisation de bilans (**Figure 70**), cette liste contient des métadonnées de ces différentes applications. Chacune de ces applications est un type de programme de remédiation, c'est-à-dire un squelette d'application spécifique à une technique de remédiation. Le praticien va pouvoir choisir quelle application doit être instanciée et en préciser les paramètres.

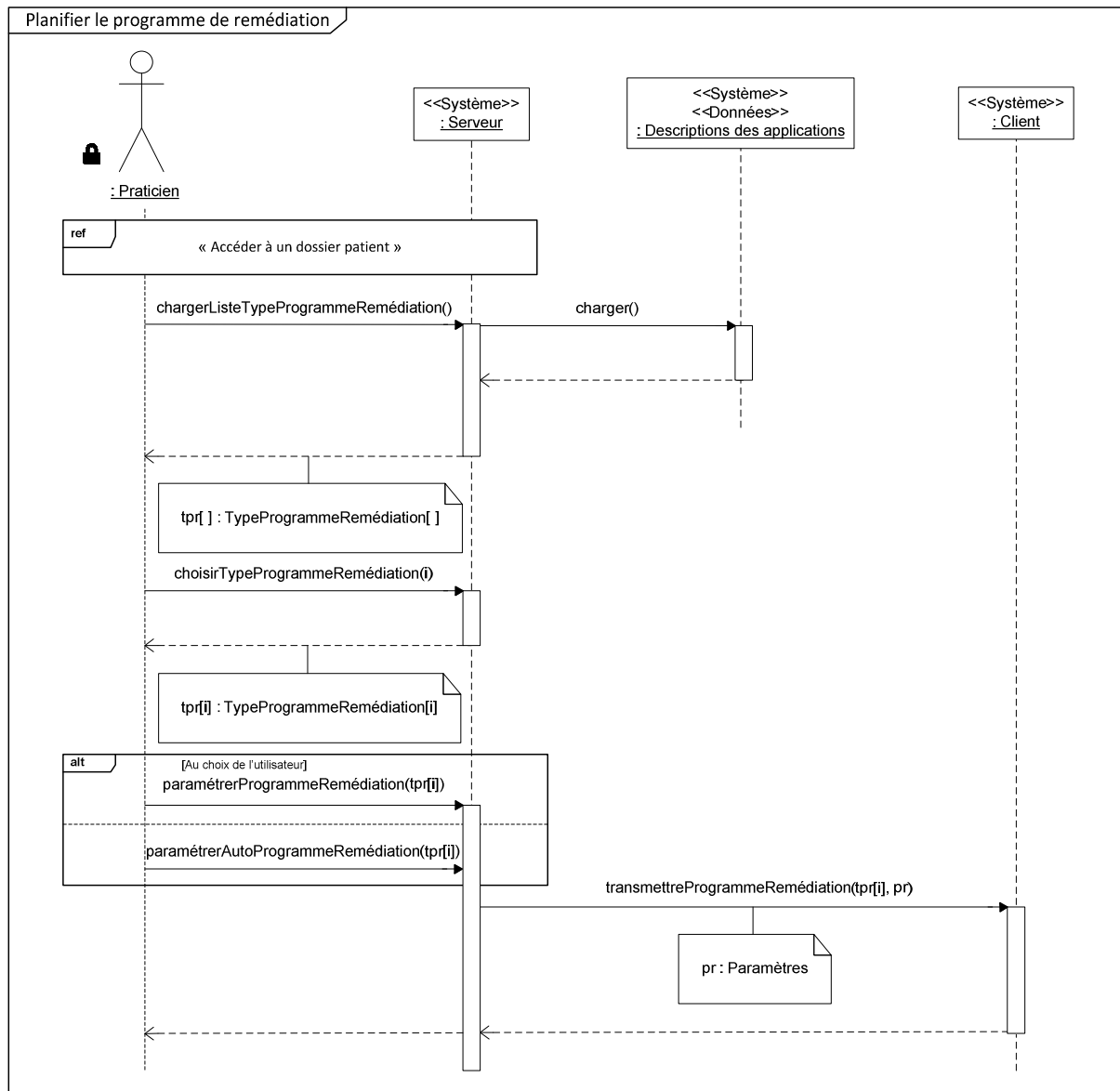


Figure 74 : Diagramme de séquence du cas d'utilisation 7 « Planifier le programme de remédiation »

Une fois l'application de remédiation instanciée sur le client, le patient peut commencer à faire les exercices définis dans le programme (**Figure 75**). Les exercices sont ainsi successivement instanciés et ils produisent chacun un résultat une fois qu'ils sont achevés.

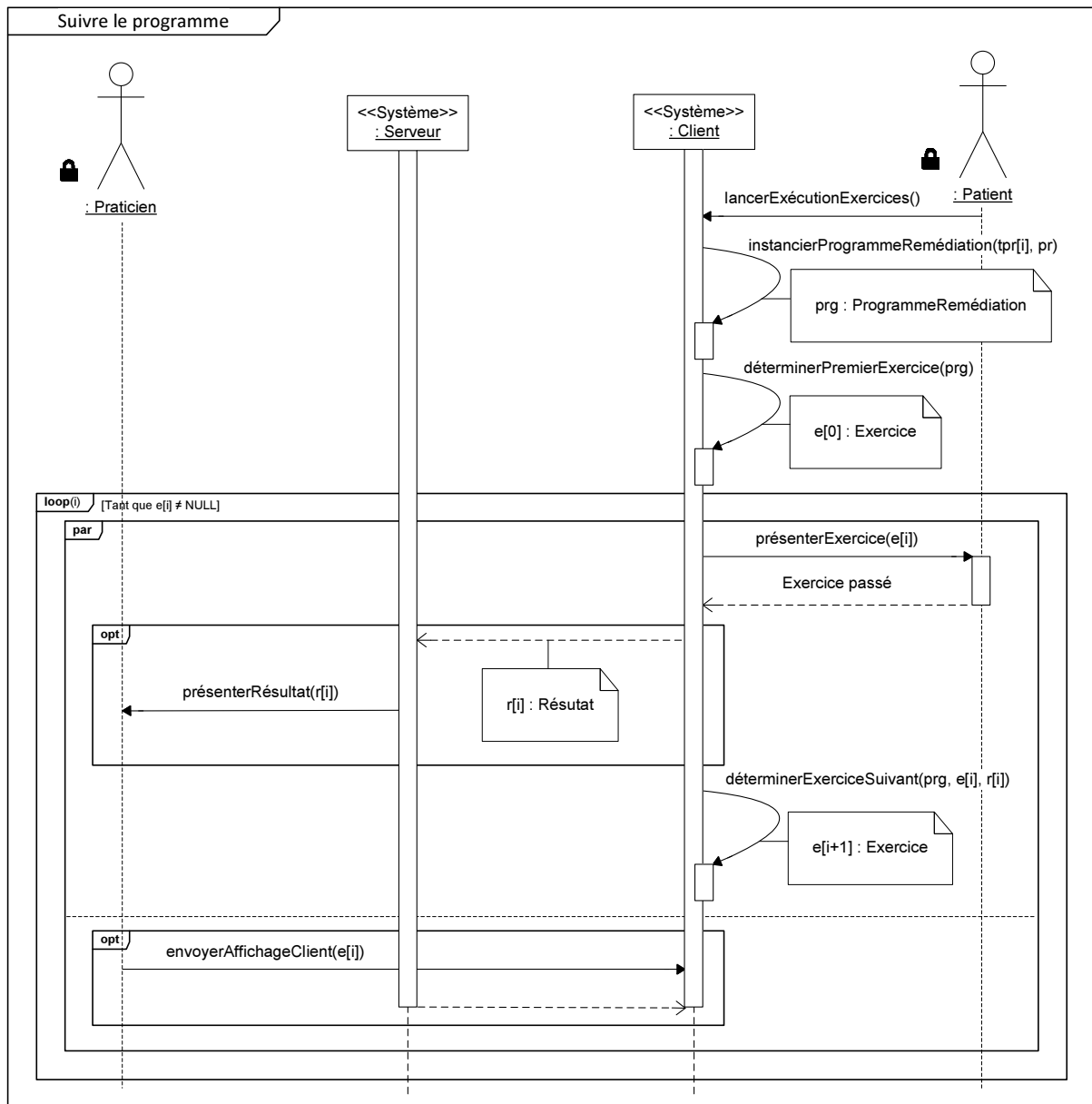


Figure 75 : Diagramme de séquence du cas d'utilisation 8 « Suivre le programme »

Comme annoncé précédemment, les exercices proposés au patient peuvent également être simulés (**Figure 76**). Contrairement au patient qui fait les exercices sur le client, la simulation s'effectue sur le serveur.

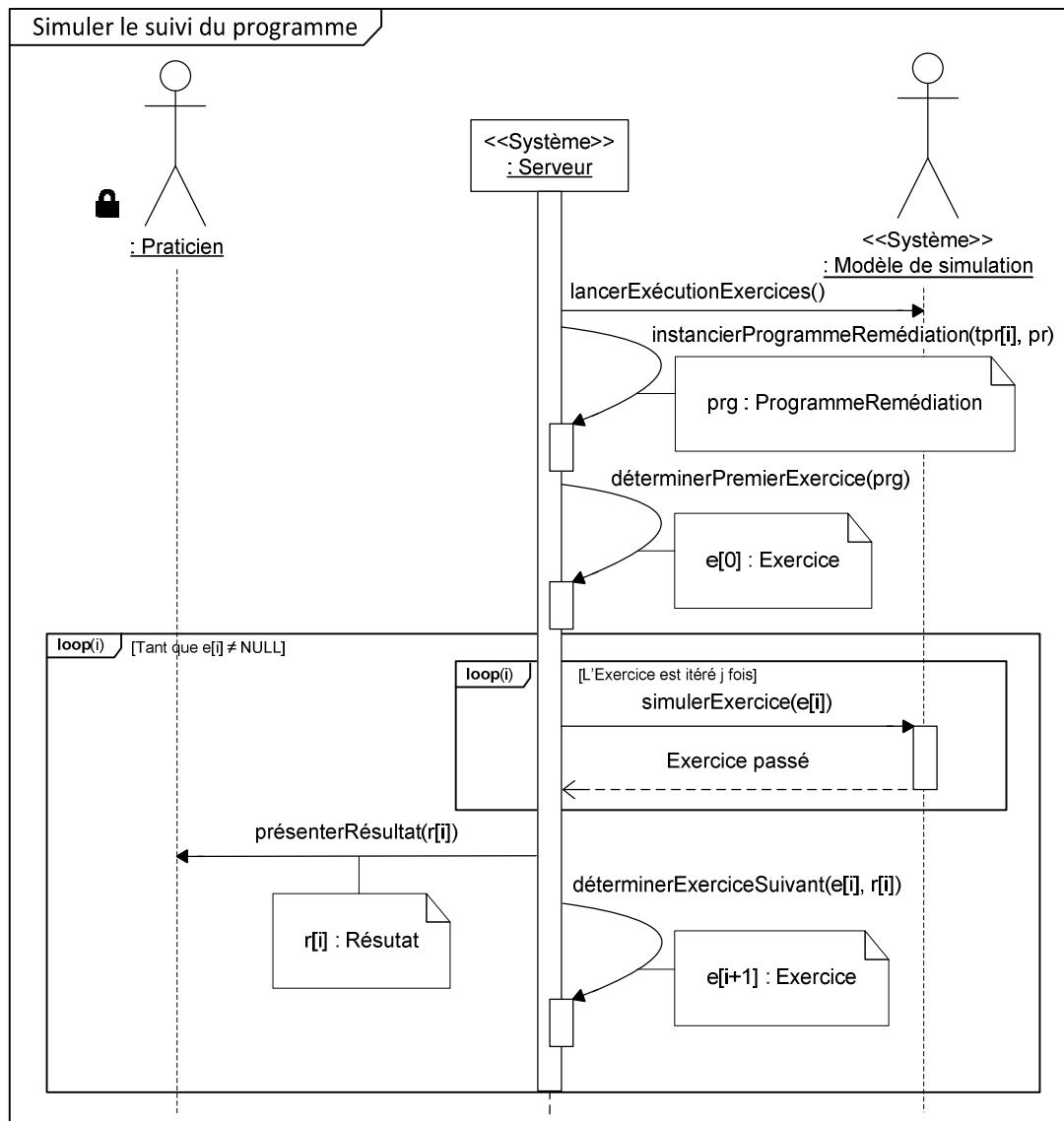


Figure 76 : Diagramme de séquence du cas d'utilisation 9 « Simuler le suivi du programme »

Une fois tous les résultats des exercices générés, un rapport peut être agrégé/constitué. Ce rapport va être une véritable aide à la décision pour le praticien afin de lui permettre d'évaluer l'efficacité de la remédiation (**Figure 77**). Cette évaluation peut être intégrée au DMP du patient.

Consécutivement à cette analyse des résultats des exercices du programme de remédiation prescrit par le praticien, ce dernier peut décider d'adapter la remédiation à la nouvelle condition du patient (**Figure 78**). Cette évolution de l'état du patient peut également donner lieu à un recalibrage du modèle de simulation qui lui est associé (**Figure 79**).

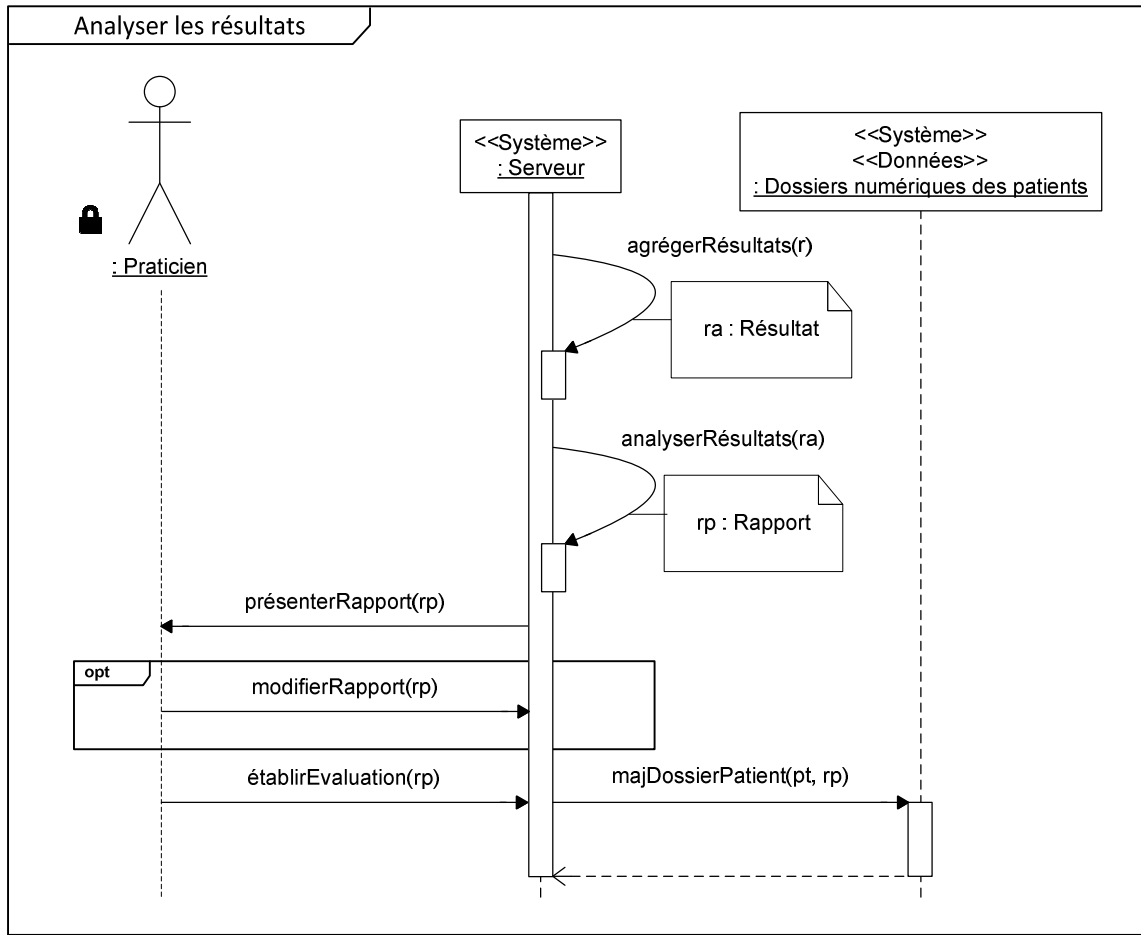


Figure 77 : Diagramme de séquence du cas d'utilisation 10 « Analyser les résultats »

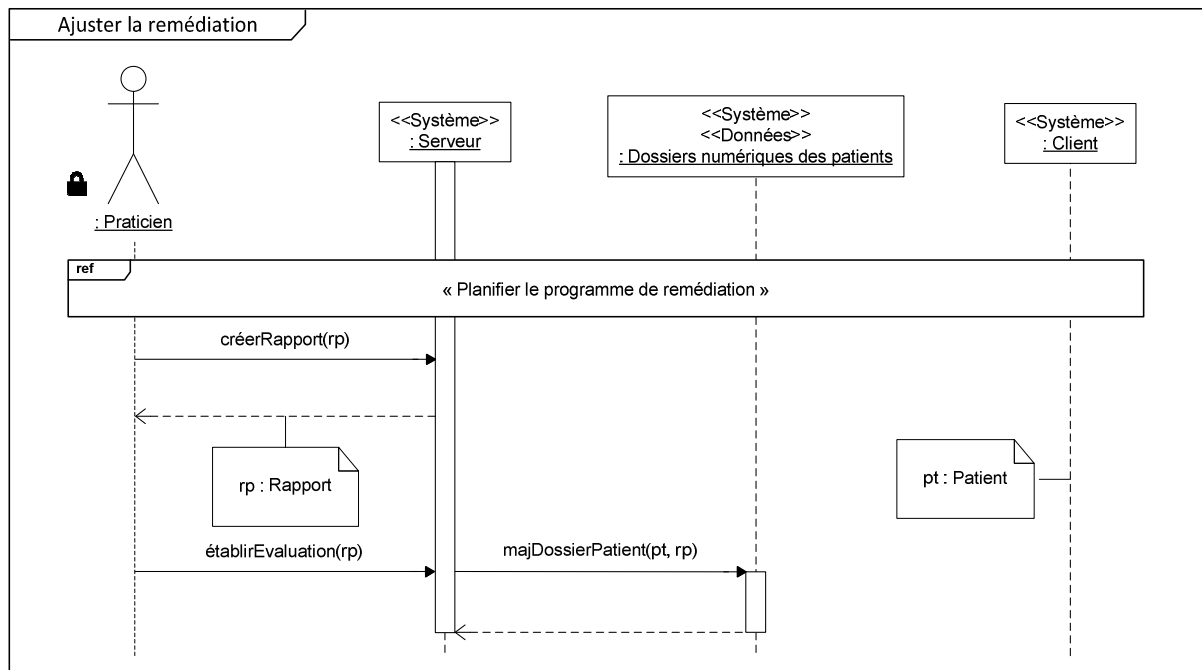


Figure 78 : Diagramme de séquence du cas d'utilisation 11 « Ajuster la remédiation »

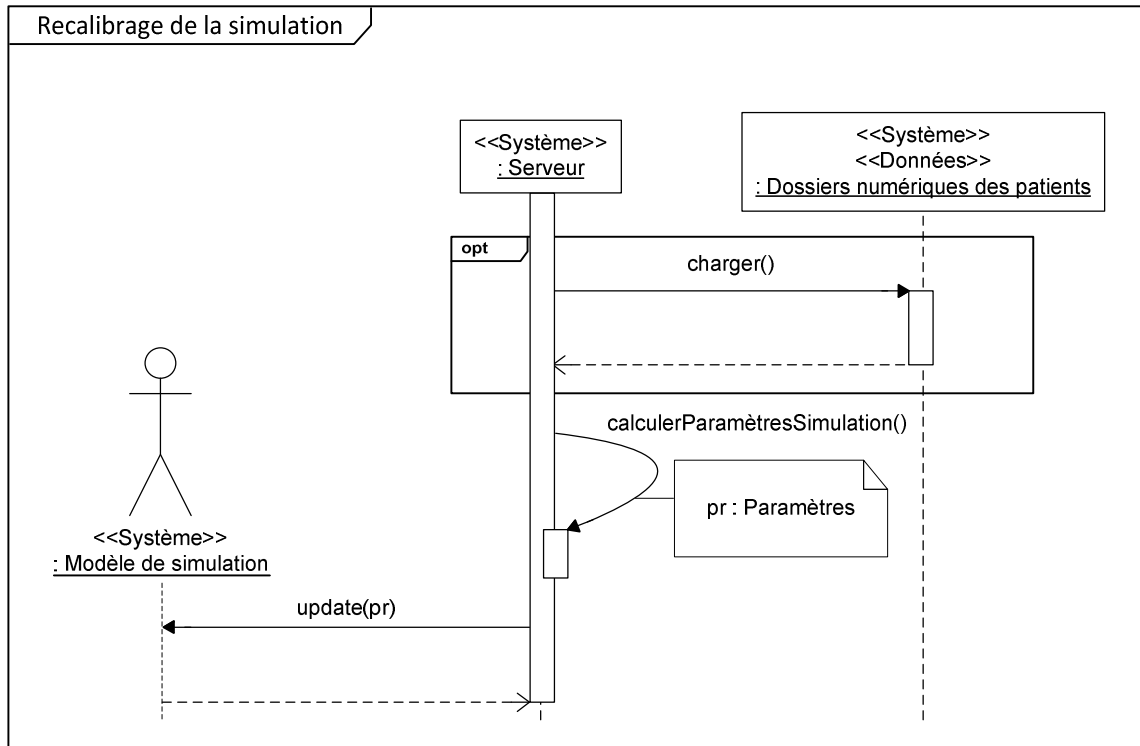


Figure 79 : Diagramme de séquence du cas d'utilisation 12 « Recalibrage de la simulation »

Nous allons maintenant nous intéresser aux cas d'utilisation informatique. Pour rappel, nous avons identifié les cas d'utilisation suivants (voir la sous-section **L'élaboration du diagramme des cas d'utilisation**) :

- Création d'un profil (**Figure 81**)
- Modification d'un profil (**Figure 82**)
- Suppression d'un profil (**Figure 83**)
- Import/Export d'un profil (**Figure 84**)
- Connexion/Déconnexion d'un profil (**Figure 85**)
- Consultation d'un dossier patient (**Figure 86**)
- Connexion client-serveur (**Figure 87**)
- Déconnexion client-serveur (**Figure 88**)

Les cinq premiers de ces cas concernent la gestion et l'utilisation des profils. En particulier, les quatre premiers décrivent la gestion des profils et le cinquième leur utilisation. Nous rappelons que cette gestion peut être soumise à la validation des modifications par un administrateur. A ce titre, nous avons identifié une fonction, mutualisée par ces cas d'utilisation de gestion des profils, qui permet la validation d'une action par l'administrateur (**Figure 80**). D'autre part, il est notable que tous ces cas d'utilisation sont indépendants du fait que l'utilisateur utilise le serveur ou le client.

La création d'un profil (**Figure 81**) va simplement consister à l'ajout, par exemple, d'un nouveau compte d'utilisateur dans la base de données prévue à cet effet. Bien que nous n'ayons pas détaillé l'ensemble des informations qui sont nécessaires à ce processus de création, on peut imaginer que l'utilisateur va devoir choisir un nom d'utilisateur, un mot de passe et fournir des renseignements le concernant tel que son rôle en temps qu'utilisateur.

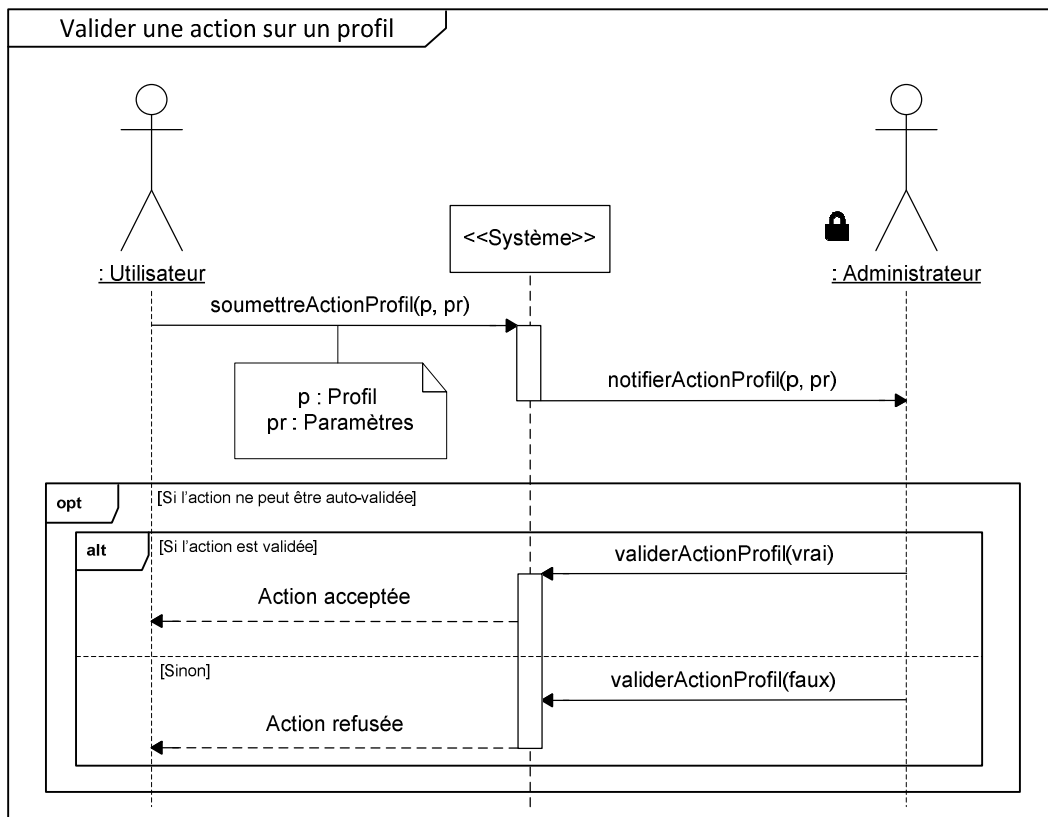


Figure 80 : Diagramme de séquence de la fonction « Valider une action sur un profil »

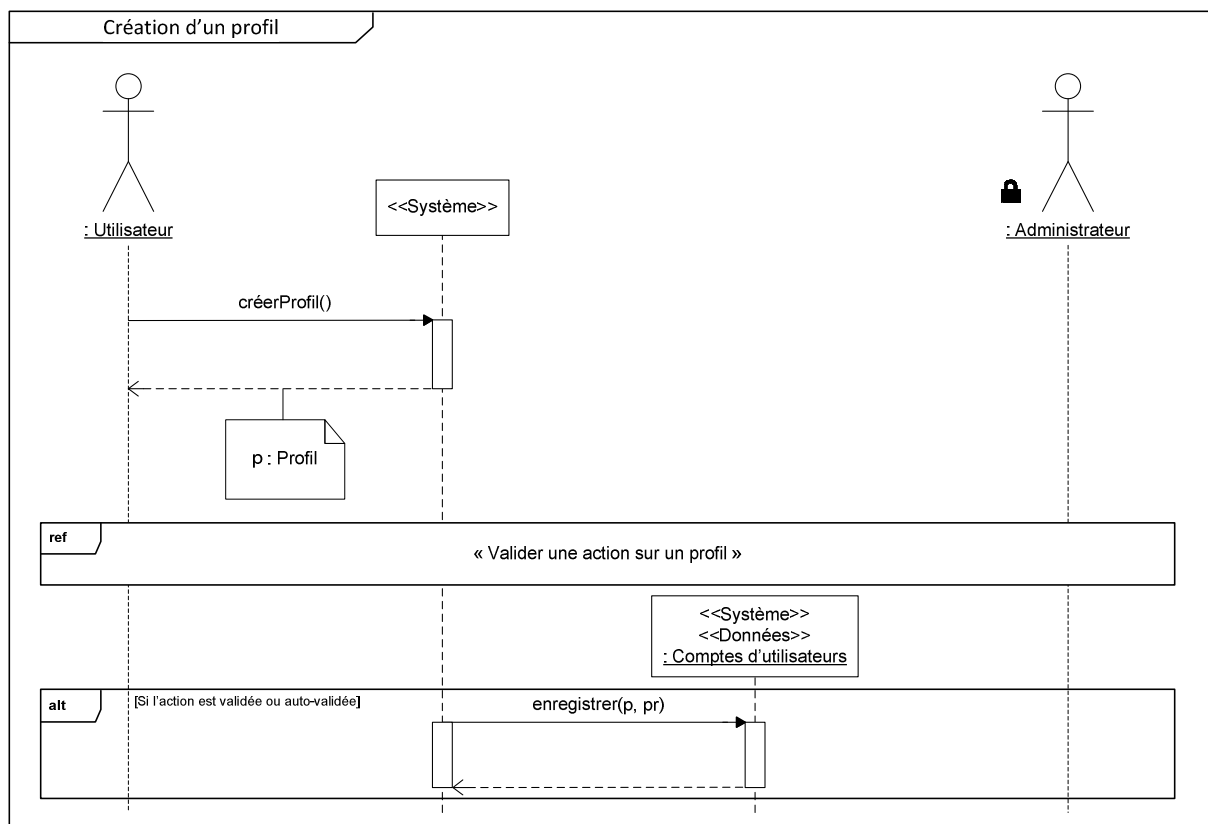


Figure 81 : Diagramme de séquence du cas d'utilisation 13 « Création d'un profil »

La modification d'un profil existant (**Figure 82**) va correspondre à la mise à jour d'un compte d'utilisateur par un utilisateur. Cette action permet à l'utilisateur de revenir sur certaines des informations qui avaient été fournies au système à la création du profil. Une telle opération peut également être effectuée par un praticien qui souhaite créer ou modifier le dossier d'un patient : dans ce cas, l'authentification s'effectue avec les identifiants du praticien. En effet, le DMP d'un patient fait partie du profil de ce dernier : il faut donc qu'un praticien puisse avec ses propres identifiants y effectuer des modifications si ses privilèges d'utilisateur le permettent.

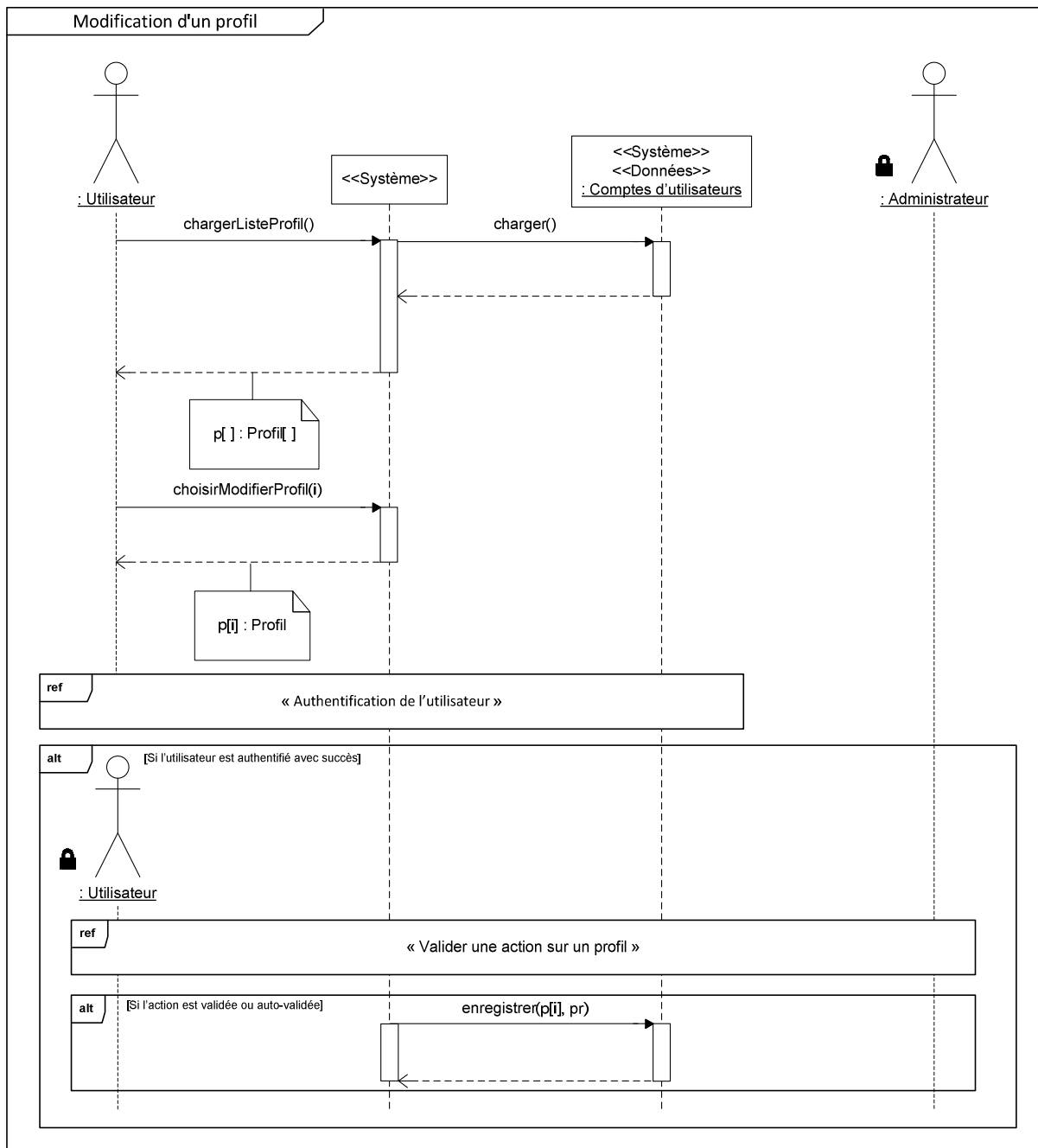


Figure 82 : Diagramme de séquence du cas d'utilisation 14 « Modification d'un profil »

Un profil peut naturellement faire l'objet d'une suppression (**Figure 83**). Ceci correspond, par exemple, à retirer ce profil de la base de données des comptes d'utilisateurs.

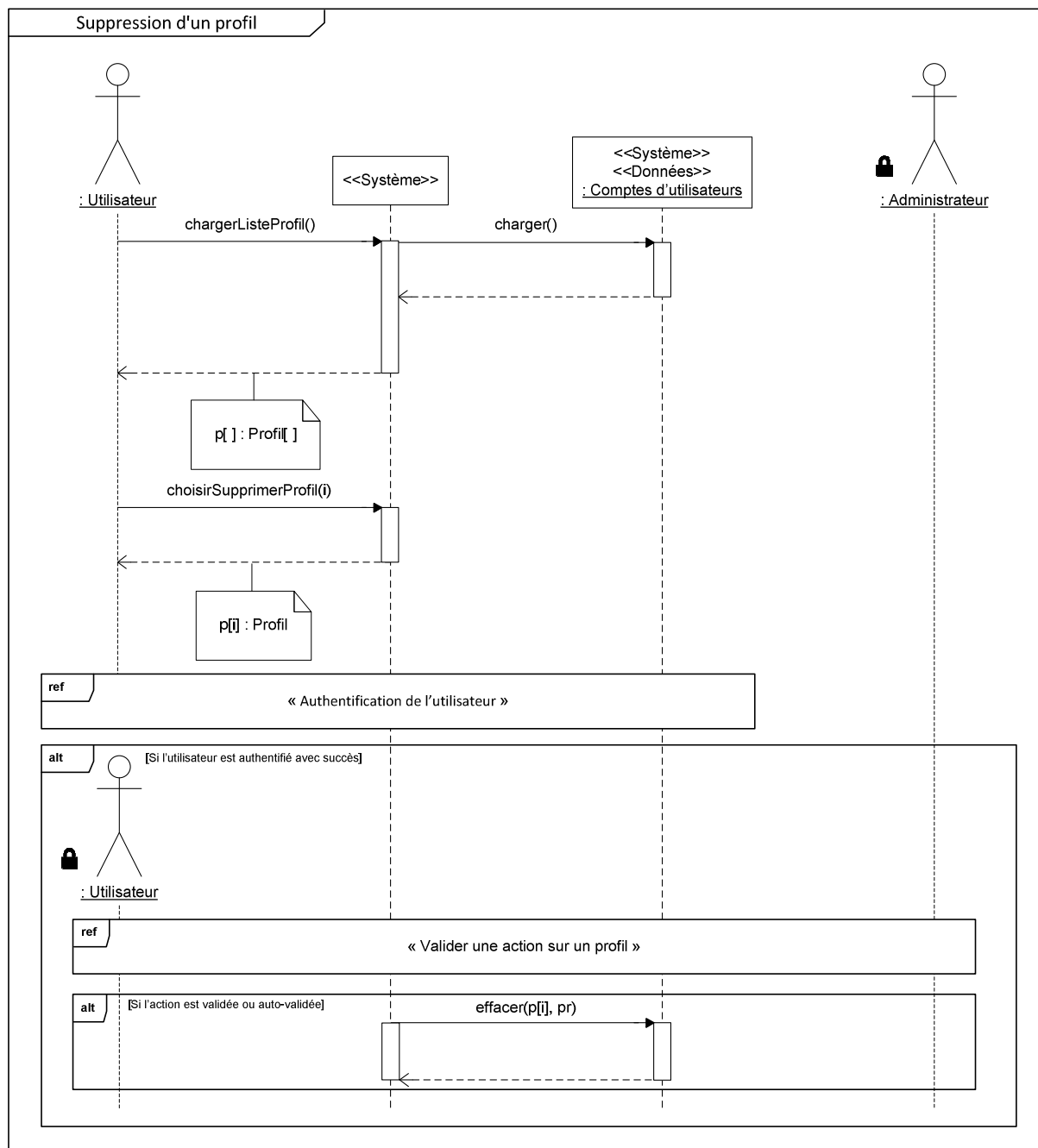


Figure 83 : Diagramme de séquence du cas d'utilisation 15 « Suppression d'un profil »

La dernière opération de gestion des profils que nous abordons est celle de l'import/export (**Figure 84**). Dans le cas de l'export, un profil va être enregistré dans un fichier. Cet enregistrement peut être local à une PAMMA ou peut faire intervenir un système de stockage distant de celle-ci. Dans le cas de l'import, un fichier, qui a été précédemment généré par la fonctionnalité d'export, va être chargé par le système. De ce chargement découle la création d'un nouveau profil cohérent avec les informations contenues dans le fichier.

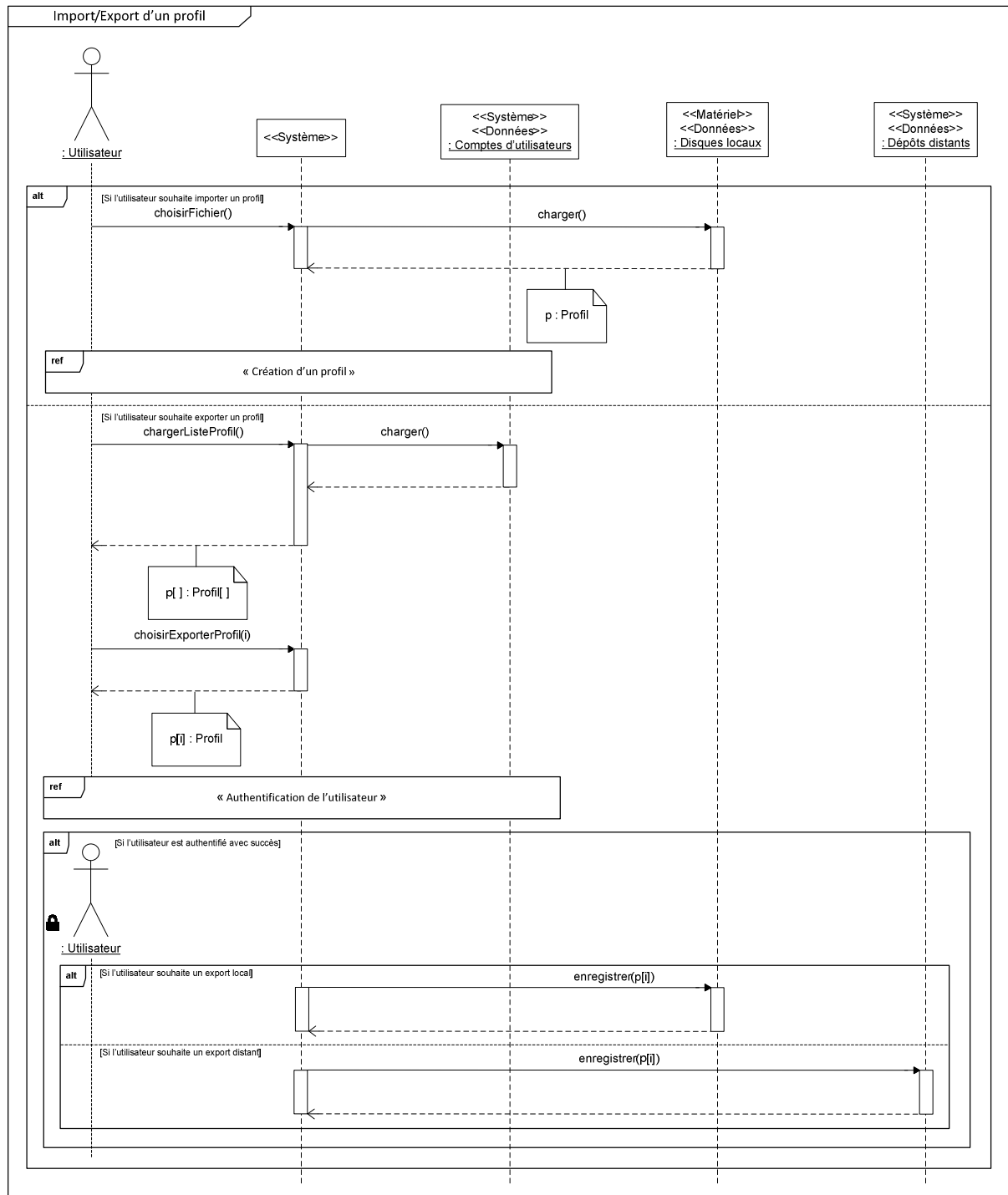


Figure 84 : Diagramme de séquence du cas d'utilisation 16 « Import/Export d'un profil »

Le cas d'utilisation suivant traite de l'utilisation qui est faite d'un profil (**Figure 85**). Dans un premier temps, l'utilisateur va être amené à choisir un profil puis s'y connecter, c'est-à-dire qu'il va s'authentifier (**Figure 66**). Une fois que cet utilisateur aura fini d'utiliser le système, il peut simplement mettre fin à l'utilisation du profil, avec lequel il s'est authentifié, en s'en déconnectant.

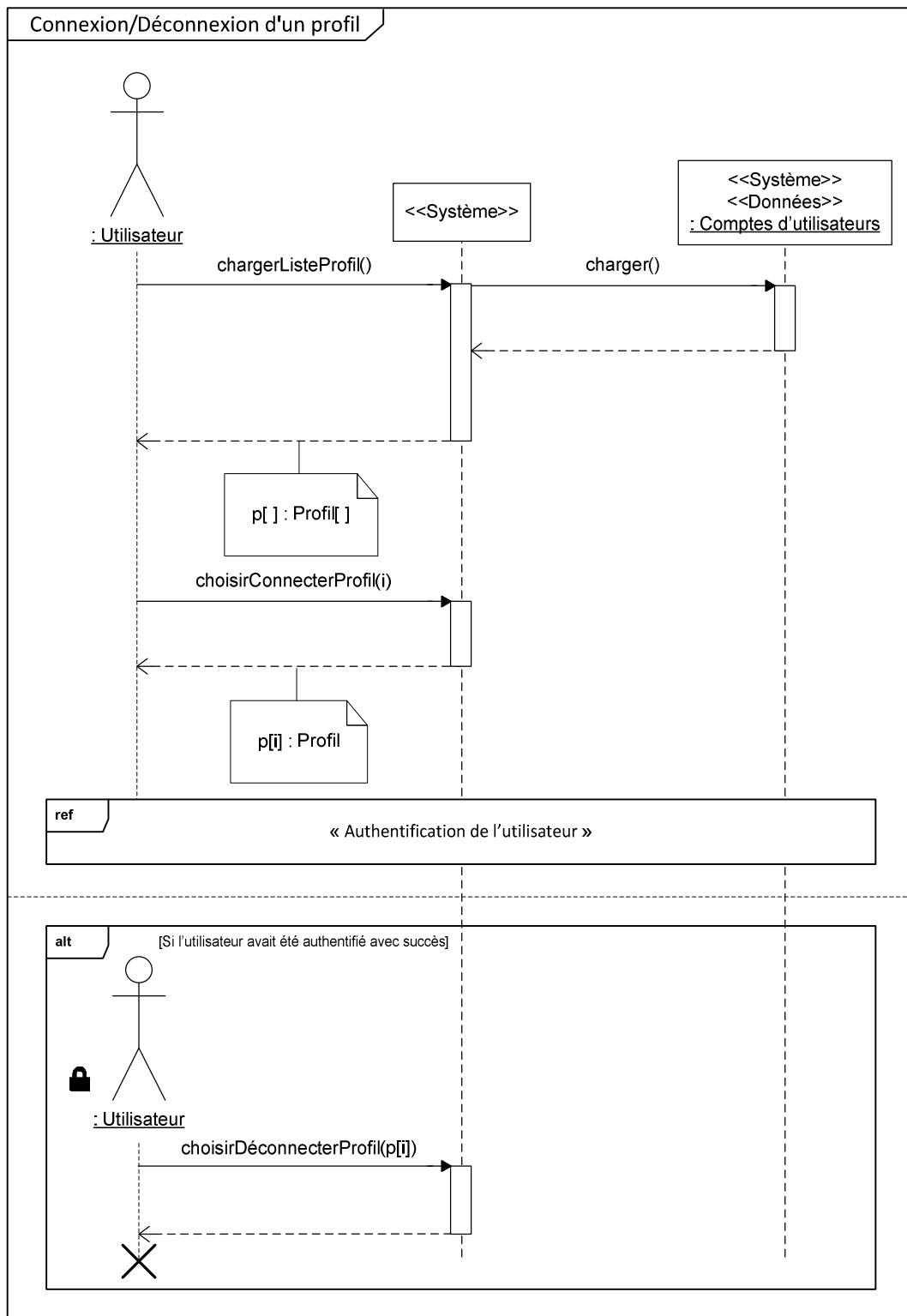


Figure 85 : Diagramme de séquence du cas d'utilisation 17 « Connexion/Déconnexion d'un profil »

Les opérations liées à la prise en charge de la dyslexie, que nous avons détaillées au travers de plusieurs cas d'utilisation, impliquent la réalisation de différents types d'opérations sur le dossier patient. Cependant, ces opérations peuvent également être effectuées par le praticien sur demande, et non uniquement dans le cadre d'un processus de soins préétabli

tel que nous l'avons déjà présenté. A ce titre, le praticien doit simplement pouvoir accéder au dossier d'un patient afin d'y apporter directement les modifications et les ajustements qu'il juge nécessaires (**Figure 67**).

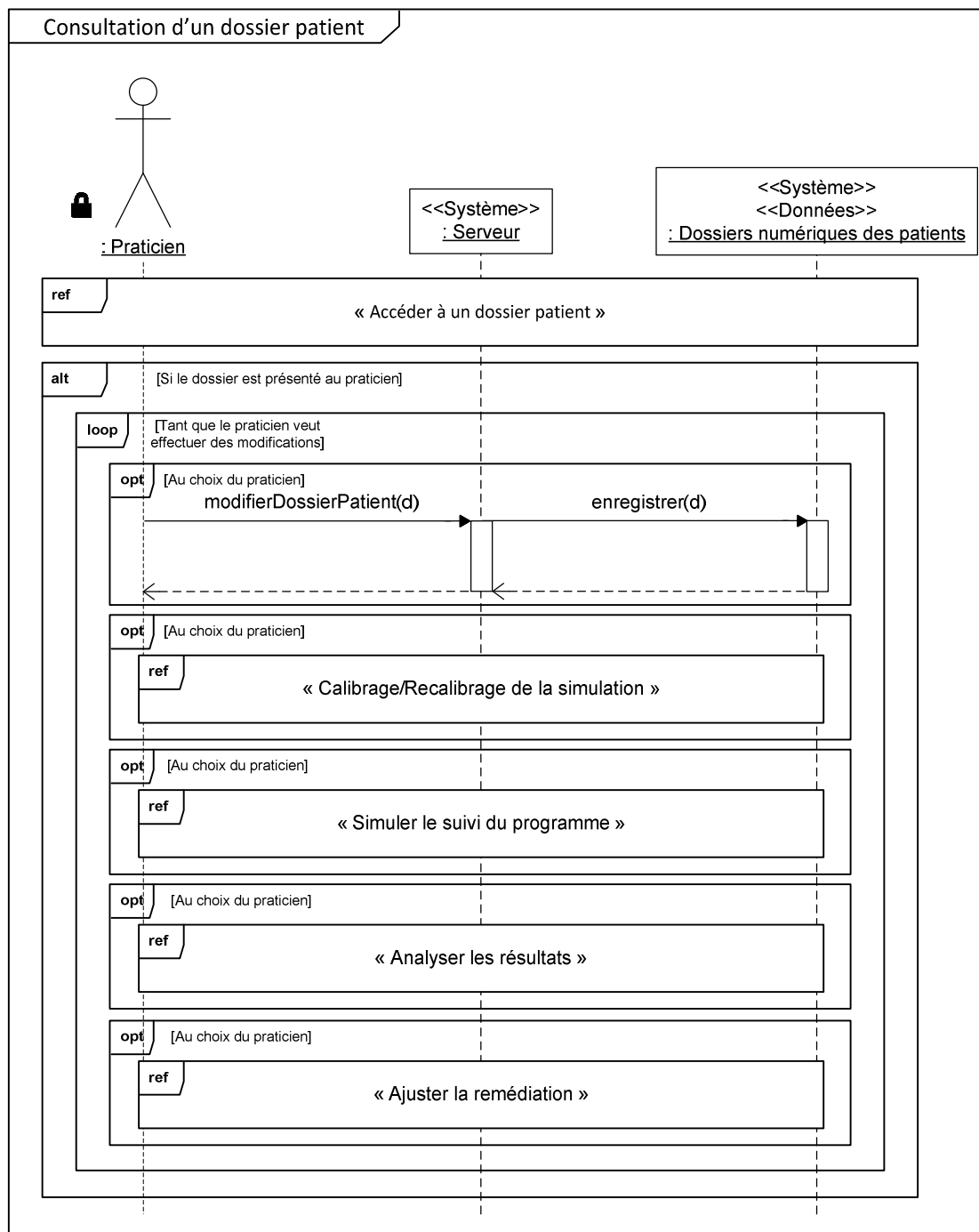


Figure 86 : Diagramme de séquence du cas d'utilisation 18 « Consultation d'un dossier patient »

En ce qui concerne les connexions client-serveur, on signalera que ces dernières peuvent aussi bien être à l'initiative du client que du serveur (**Figure 87**). De la même manière, l'un comme l'autre peuvent mettre fin à la déconnexion (**Figure 88**). Dans ce cas, l'autre plateforme ne reçoit qu'un avis de déconnexion et ne peut pas y mettre un quelconque veto.

Hormis ce type de déconnexions volontaires, les PAMMA devront régulièrement vérifier que leurs connexions enregistrées sont toujours actives, et si cela n'est plus le cas elles devront y mettre un terme.

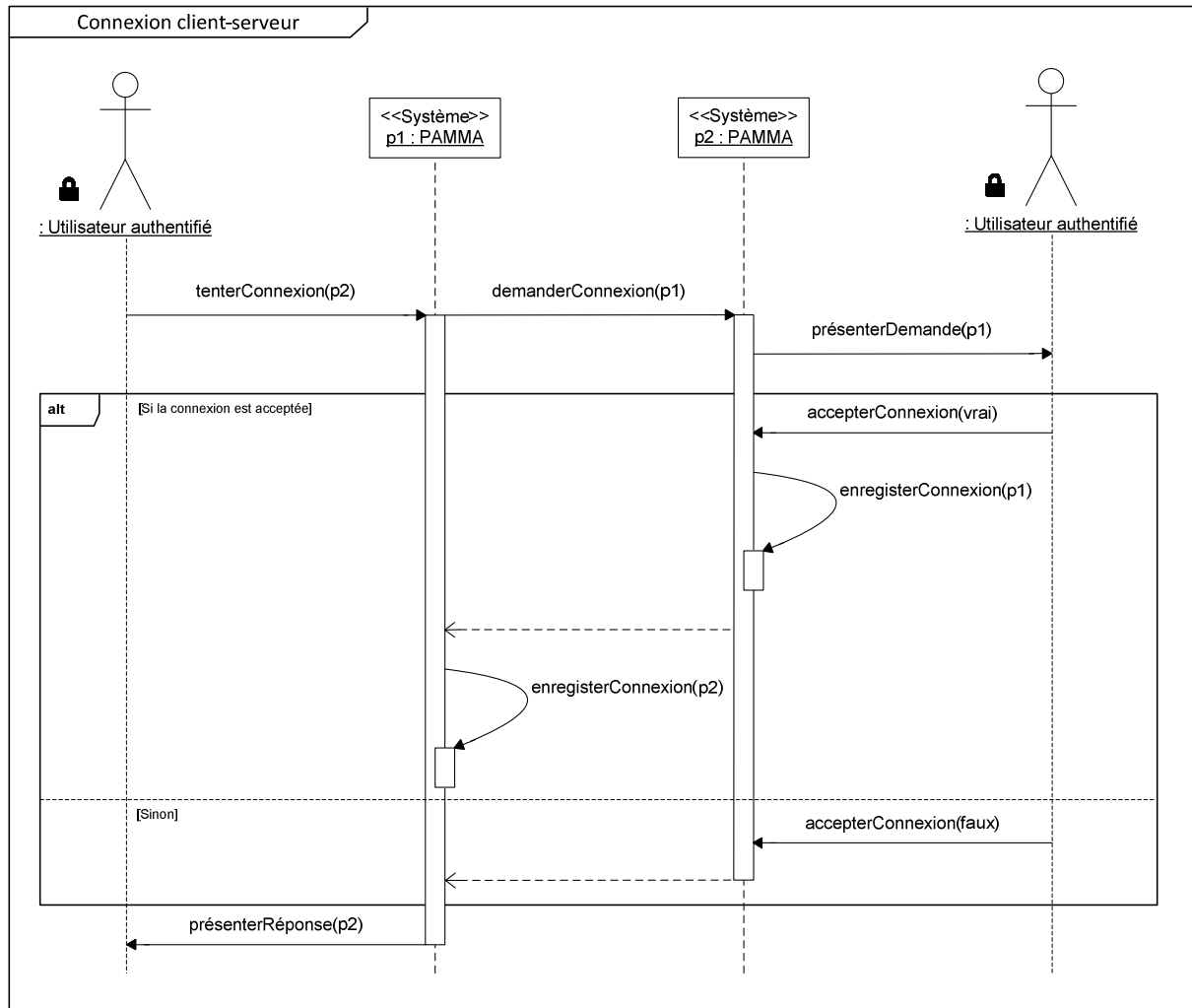


Figure 87 : Diagramme de séquence du cas d'utilisation 19 « Connexion client-serveur »

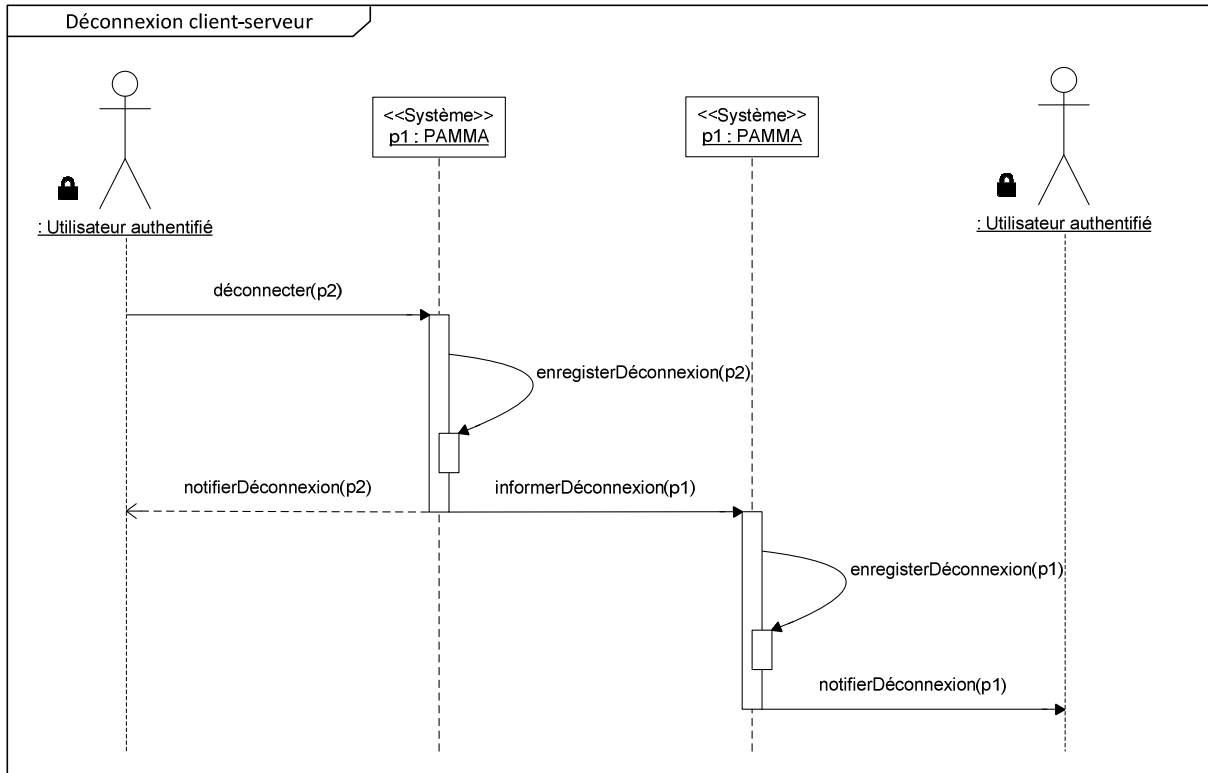


Figure 88 : Diagramme de séquence du cas d'utilisation 20 « Déconnexion client-serveur »

4) L'élaboration des diagrammes d'état-transition

FICHE GUIDE – FG10	
Activité 3 : Analyse des cas d'utilisation	
Sous-activité 3.4 : Elaboration des diagrammes d'état-transition (DET)	
Objectif	Définir d'une part les états des objets significatifs du système étudié et d'autre part les actions associées aux transitions
Point de départ	Cas d'utilisation décrits de manière textuelle et concepts métiers identifiés et définis
Point d'arrivée	Les différents états des objets et les transitions entre objets représentés dans le DET
Démarche d'élaboration	
Pour chaque objet significatif du système étudié :	
<p>1 - Identifier les états pertinents de l'objet à représenter. Considérer les états élémentaires mais aussi composites si nécessaire.</p> <p>2 - Définir les transitions entre les états des objets avec éventuellement les gardes associées. Utiliser, le cas échéant, tous les opérateurs disponibles (points de jonction, points de choix...).</p> <p>3 - Définir les actions pour les transitions et les activités pour les objets.</p> <p>4 - Représenter l'état initial et le ou les états finaux.</p>	

Tableau 16 : Fiche Guide 10 pour UP7 (d'après (Gabay & Gabay, 2008))

Parmi les objets que nous avons mis en évidence avec notre analyse des cas d'utilisation, il nous semble important de préciser les différents états et transitions des objets suivants :

- Une PAMMA (**Figure 89**)
- Un utilisateur (**Figure 90**)
- Un profil (**Figure 91**)
- Une connexion entre PAMMA (**Figure 92**)
- Un dossier patient (**Figure 93**)
- Un modèle de simulation (**Figure 94**)
- Une application de la PAMMA (**Figure 95**)

Lorsque nous considérons la PAMMA comme un objet (**Figure 89**), nous voulons pouvoir représenter les différents états qu'elle peut prendre sur une machine qui l'hébergerait. Ces états sont ainsi perceptibles indépendamment de l'exécution de la PAMMA ou d'une application qu'elle permettrait de définir et d'exécuter. On notera que seul le passage par l'état d'activation de la PAMMA a réellement besoin d'être stocké sur cette dernière, les autres états pouvant être déduits facilement d'une observation du système et/ou de son système de fichiers sur la machine qui l'héberge. Ce passage à l'état d'activation pourra être

signalé par la présence de données sur la forme d'un fichier ou d'une entrée en base de données.

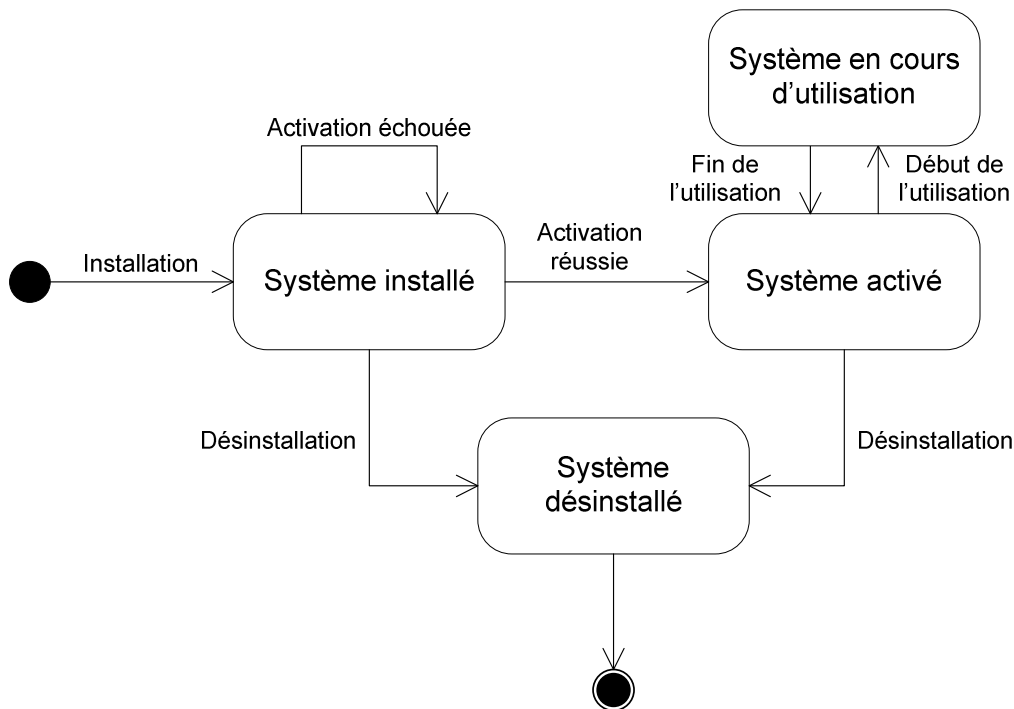


Figure 89 : Diagramme d'état-transition d'une PAMMA

Le second objet qui nous intéresse est l'utilisateur (Figure 90). En le représentant, nous voulons décrire à un haut niveau d'abstraction le rôle qu'adopte un individu quand il souhaite utiliser la PAMMA. Cet objet et ses états n'ont pas réellement besoin d'être enregistrés par le système mais ils peuvent être déduits de l'état de la PAMMA (Figure 89) et de celui des profils (Figure 91).

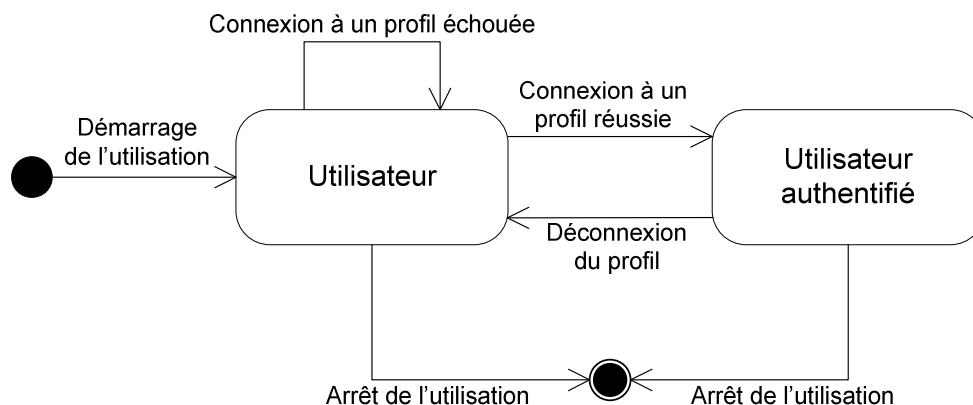


Figure 90 : Diagramme d'état-transition d'un utilisateur

L'objet suivant correspond à un profil tel qu'utilisé par un utilisateur authentifié (Figure 91). Cette représentation étend d'ailleurs les informations fournies par l'analyse des cas d'utilisation dans la mesure où elle précise qu'un tel profil peut faire l'objet d'une suppression. Cet objet va pouvoir être stocké au sein d'une PAMMA sous la forme de fichiers

ou d'entrées en base de données. Lorsqu'il est en cours d'utilisation, cet objet se retrouve évidemment en mémoire d'exécution.

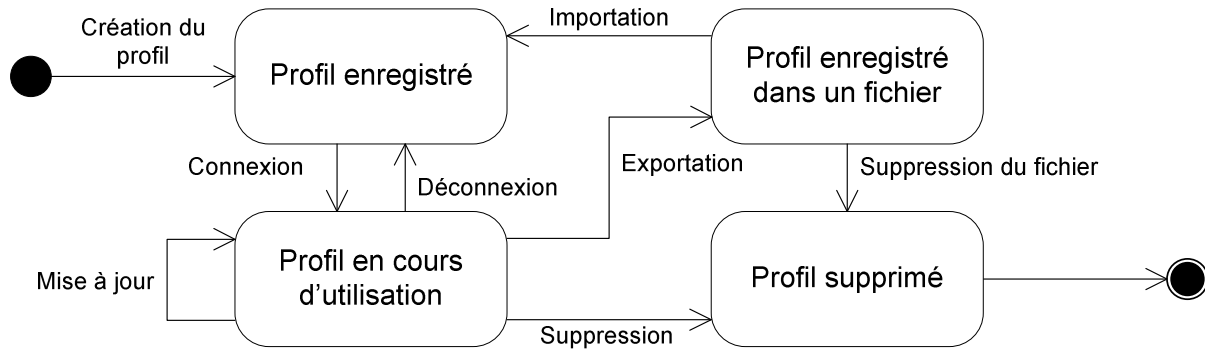


Figure 91 : Diagramme d'état-transition d'un profil

Les connexions entre PAMMA font également l'objet d'un diagramme d'état-transition (**Figure 92**). On signalera que tels objets sont maintenus sous la forme de paires, pour lesquelles chacune des deux PAMMA impliquées dans la connexion conserve une des deux copies de l'objet. Ces objets pourront être stockés en mémoire d'exécution mais également sous la forme de données statiques dans le cas de connexions persistantes au sein du réseau de PAMMA.

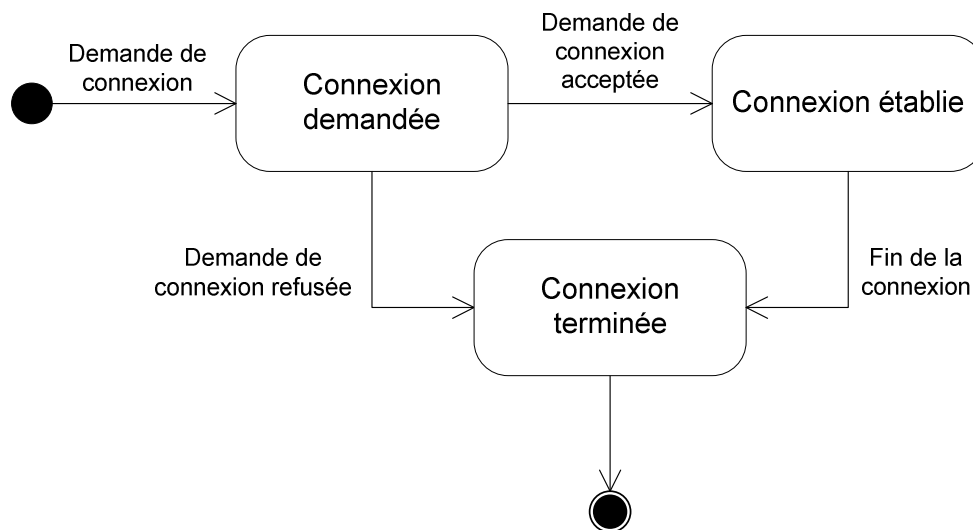


Figure 92 : Diagramme d'état-transition d'une connexion entre PAMMA

Les profils des patients peuvent être étendus à l'aide d'un dossier patient consultable par les praticiens (**Figure 93**). Il s'agit de permettre aux profils des patients d'être accédés et agrémentés de nouvelles informations par les praticiens. En effet, hormis l'import/export on retrouve en tous points un comportement similaire à celui d'un profil (**Figure 91**). Ainsi, comme dans le cas d'un profil, cette représentation étend les informations fournies par l'analyse des cas d'utilisation car elle stipule que cet objet peut être supprimé. Une telle suppression ne devrait cependant pouvoir être réalisée que par un administrateur du système. Pour les mêmes raisons, cet objet va pouvoir être stocké au sein d'une PAMMA

sous la forme de fichiers ou d'entrées en base de données, et il se retrouve en mémoire d'exécution quand il est en cours d'utilisation.

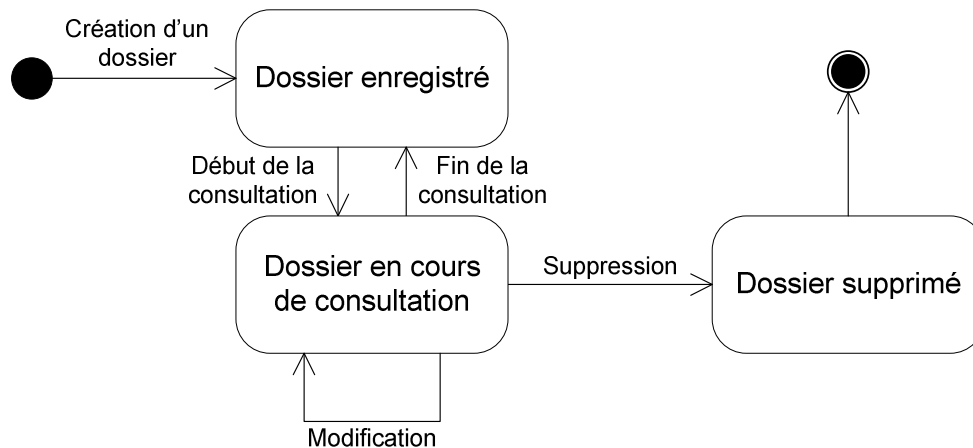


Figure 93 : Diagramme d'état-transition d'un dossier patient

L'objet suivant est le modèle de simulation qui devrait permettre de reproduire des comportements de patients (**Figure 94**). Ce modèle peut venir agrémenter le dossier d'un patient. A ce titre, il doit être possible de le supprimer comme l'est le dossier patient (**Figure 93**). Il peut aussi être stocké sous la forme de fichiers ou d'entrées en base de données, et il est mis en mémoire d'exécution quand il est en cours d'utilisation.

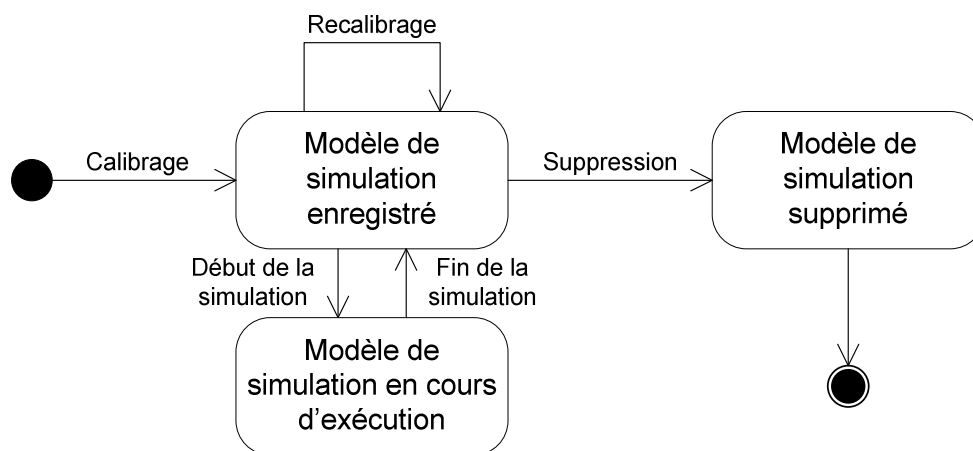


Figure 94 : Diagramme d'état-transition d'un modèle de simulation

Le dernier des objets que nous allons étudier est une application fonctionnant sur une PAMMA (**Figure 95**). On pourra signaler que ce diagramme se compose de deux types d'états différents : les états relatifs à la modélisation/spécification de l'application et ceux relatifs à son instanciation/exécution. Dans le cas de la spécification, l'objet peut être stocké sous la forme de fichiers ou d'entrées en base de données au sein d'une PAMMA. Dans celui de l'exécution, il passe naturellement en mémoire d'exécution. Il est notable qu'une application est constituée d'objets de granularité moindre tels que nous l'avons précédemment indiqué sur nos diagrammes de classes (**Figure 45** et **Figure 46**).

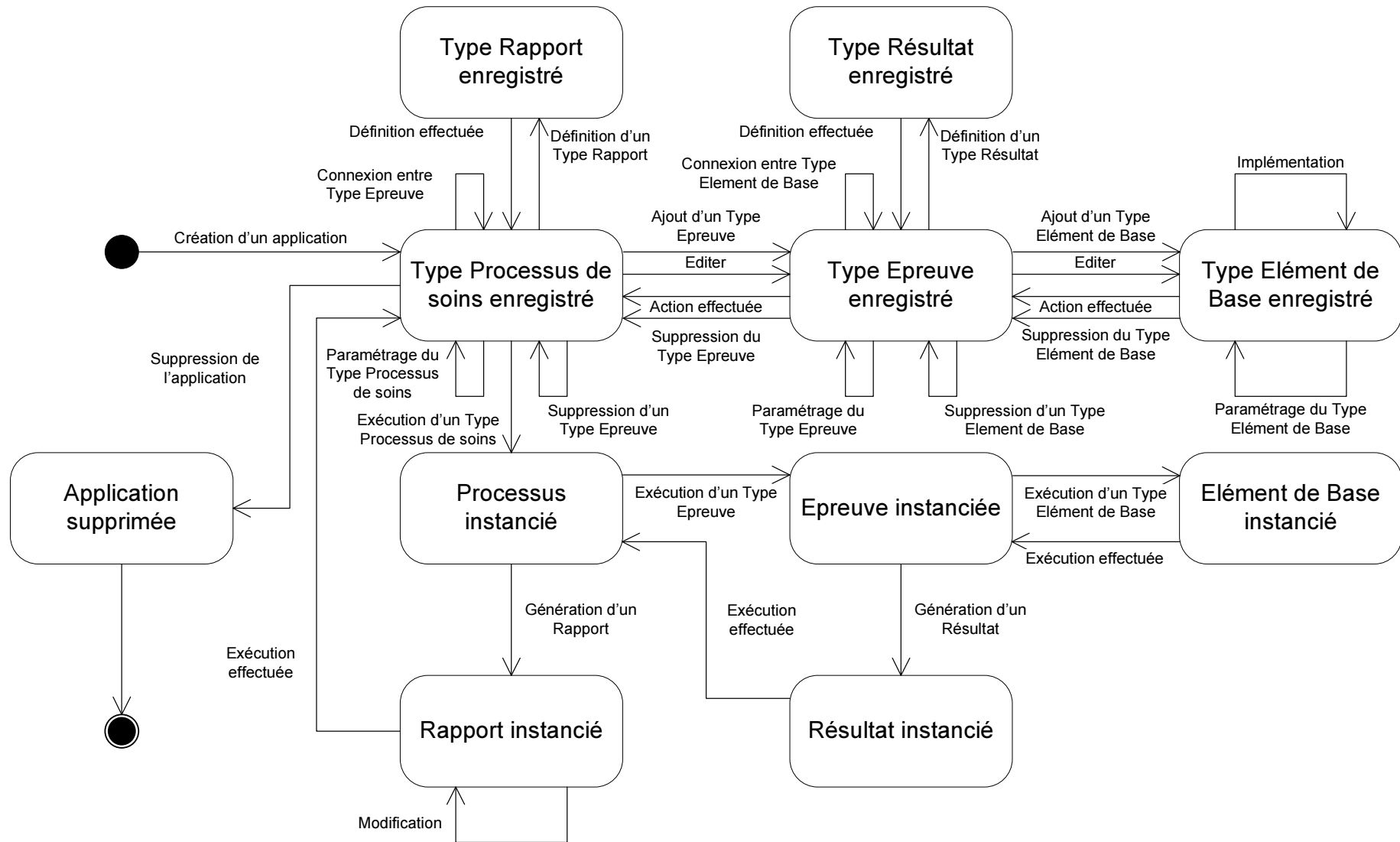


Figure 95 : Diagramme d'état-transition d'une application de la PAMMA

5) L'élaboration des interfaces utilisateur

FICHE GUIDE – FG11	
Activité 3 : Analyse des cas d'utilisation	
Sous-activité 3.5 : Elaboration des interfaces utilisateur (IHM)	
Objectif	Modéliser les interfaces utilisateur de tous les cas d'utilisation pour donner une vue concrète de l'application aux utilisateurs
Point de départ	Cas d'utilisation décrits de manière textuelle et interactions entre acteurs et objets du système définies pour tous les cas d'utilisation
Point d'arrivée	Interfaces utilisateur définies pour tous les cas d'utilisation
Démarche d'élaboration	
<p>Pour chaque cas d'utilisation (CU) :</p> <ol style="list-style-type: none"> 1 - Identifier toutes les « entrées » de l'IHM qui correspondent aux paramètres des messages du DSE pour chaque CU. 2 - Représenter les dispositifs d'« entrées » sous forme de composants IHM : <ul style="list-style-type: none"> - zone de saisie à l'écran - liste de choix - boutons radios... 3 - Ajouter à l'IHM les composants de validation (boutons). 4 - Représenter les «sorties » de l'IHM qui correspondent aux résultats du message du DES pour chaque CU. 	

Tableau 17 : Fiche Guide 11 pour UP7 (d'après (Gabay & Gabay, 2008))

Les maquettes d'interfaces servent à montrer quelles informations ont besoin d'être affichées. Ces maquettes sont de véritables prototypes de l'interface future (Beaudouin-Lafon & Mackay, 2007) et peuvent, à ce titre, être présentées aux utilisateurs afin de pouvoir évaluer leur utilisabilité (Anderson et al., 2001; Ferré et al., 2001). Ceci fait partie intégrante de la démarche de développement itératif embrassée par l'UP. Il est donc important de suivre un certain nombre de directives concernant l'utilisabilité (Juristo, Moreno, & Sanchez-Segura, 2006, 2007) afin de minimiser le nombre d'itérations à réaliser au cours de ces activités du processus de développement et de maximiser la satisfaction des utilisateurs lorsqu'ils utilisent le système. Le **Tableau 18** donne un aperçu des principales caractéristiques à prendre en compte (pour les directives complètes, consulter (Juristo et al., 2006)) que nous avons utilisées pour la conception de nos maquettes d'IHM.

Au sujet des figures de cette sous-section, on précisera que des flèches peuvent indiquer des relations de précedence entre écrans présentés sur une même figure.

Caractéristiques d'utilisabilité	Mécanique d'utilisabilité	Objectif
Retour	État du système	Informers les utilisateurs du statut interne du système.
	Interaction	Informers les utilisateurs que le système a pris en compte une interaction de l'utilisateur.
	Avertissement	Informers les utilisateurs de toute action ayant des conséquences importantes.
	Action prolongée	Informers les utilisateurs que le système réalise une action pouvant prendre un certain temps.
Annulation	Annulation globale	Annuler les actions du système à plusieurs niveaux.
	Annulation spécifique	Annuler plusieurs actions sur un objet.
	Abandon d'opération	Annuler l'exécution d'une action ou de toute l'application.
	Retourner	Revenir à un état particulier dans une séquence d'exécution de commandes.
Prévention d'erreurs	Saisie de texte structuré	Aider à prévenir des erreurs de l'utilisateur lors de la saisie de données.
Assistance	Exécution pas-à-pas	Aider à réaliser des tâches qui nécessitent différentes étapes.
Profil d'utilisateur	Préférences	Enregistrer chaque option de l'utilisateur lorsqu'il utilise les fonctions du système.
	Espace d'objets personnels	Enregistrer chaque option de l'utilisateur lorsqu'il utilise l'interface du système
	Favoris	Enregistrer certaines zones d'intérêt pour l'utilisateur.
Aide	Aide multi niveaux	Fournir différents niveaux d'aide destinés à différents types d'utilisateurs.
Agrégation de commandes	Agrégation de commandes	Exprimer des actions complexes à l'aide de séquences de commandes de portée moindre.

Tableau 18 : Liste de caractéristiques d'utilisabilité fonctionnelle et de leurs mécaniques associées (d'après (Juristo et al., 2006))

Nous allons maintenant présenter nos maquettes d'IHM dans l'ordre dans lequel elles seront rencontrées pendant les navigations des utilisateurs sur le système. A ce titre, nous considérons que les fichiers de la PAMMA sont déjà déployés sur une machine hôte et qu'ils sont correctement configurés. Ainsi, lorsqu'un utilisateur débute son utilisation du système,

il doit commencer par activer ce dernier (**Figure 96 (a)**). Toute tentative d'activation pourra être soit une réussite (**Figure 96 (b)**) soit un échec (**Figure 96 (c)**)

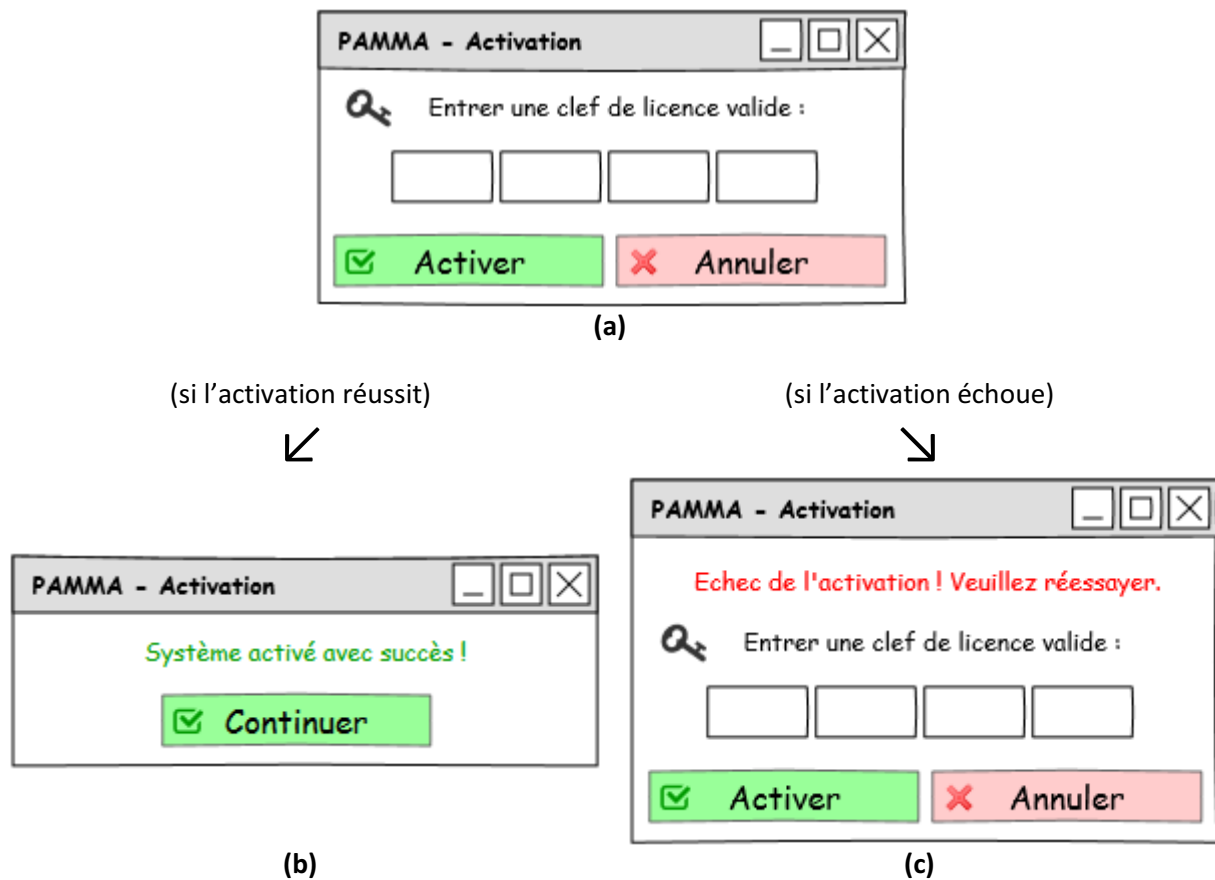


Figure 96 : Prototypage d'IHM pour l'activation du système

Pour tous les écrans de la **Figure 96**, si l'utilisateur presse le bouton « Annuler », l'application se ferme. Si l'activation est un succès, il suffit de presser le bouton « Continuer » pour accéder à l'écran de gestion des profils (**Figure 97**). D'autre part, à l'avenir cette interface d'activation ne sera plus présentée et l'utilisateur arrivera directement à l'écran de gestion des profils. La gestion des profils n'est pas spécifique à notre application pour le traitement de la dyslexie et elle se présente avant la navigation sur cette dernière, telle que nous l'avons précédemment décrite (voir la sous-section **L'élaboration du schéma de navigation générale**). Cette gestion des profils permet la création (**Figure 99**), la modification (**Figure 100**), la suppression (**Figure 101**), l'utilisation (**Figure 98**), l'importation (**Figure 102**) et l'exportation (**Figure 103**) de profils d'utilisateurs comme le montre la **Figure 97**. Cette interface permet la mutualisation de l'action de sélection du profil (voir les diagrammes de gestion des profils de la sous-section **L'élaboration des diagrammes de séquence**). Hormis la création et l'importation d'un profil, ces actions nécessitent une authentification de l'utilisateur à l'aide du mot de passe du profil cible telle que nous l'avons présentée dans le cas de l'utilisation d'un profil (**Figure 98**). On pourra également signaler que le détail des informations à fournir pour la création, ou la modification, d'un profil n'est pas clairement établi (**Figure 99** et **Figure 100**). Il faudra que ces informations permettent l'identification non-ambigüe des utilisateurs, en particulier pour distinguer facilement les différents types d'utilisateurs tels que les professionnels de santé et les patients. Les statuts qu'un profil pourrait prendre sont « actif », « suspendu » ou « en attente de validation ».

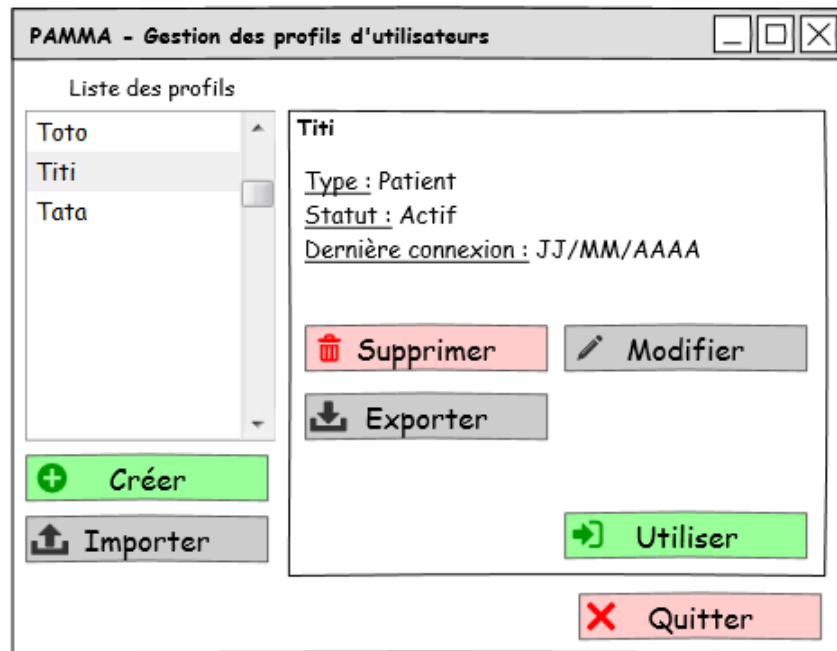
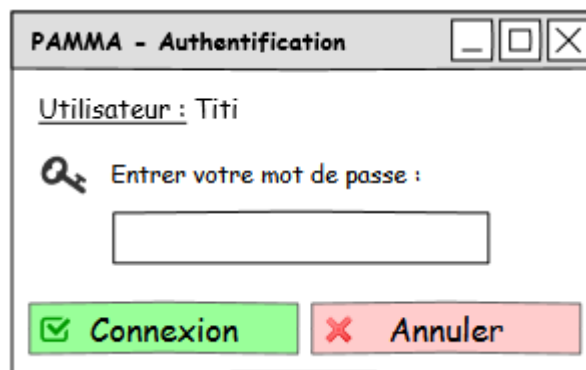


Figure 97 : Prototype d'IHM pour la gestion des profils



(a)

(si l'authentification réussit)



(b)

(si l'authentification échoue)



(c)

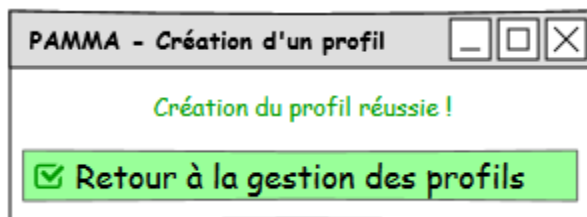
Figure 98 : Prototype d'IHM pour l'utilisation d'un profil



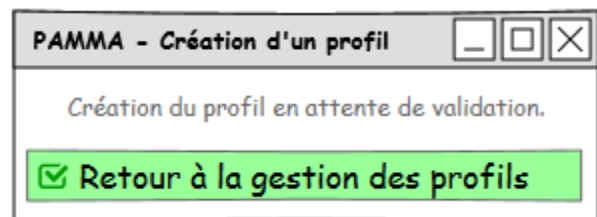
(a)

(si la création est réussie)

(si la création est en attente de validation)

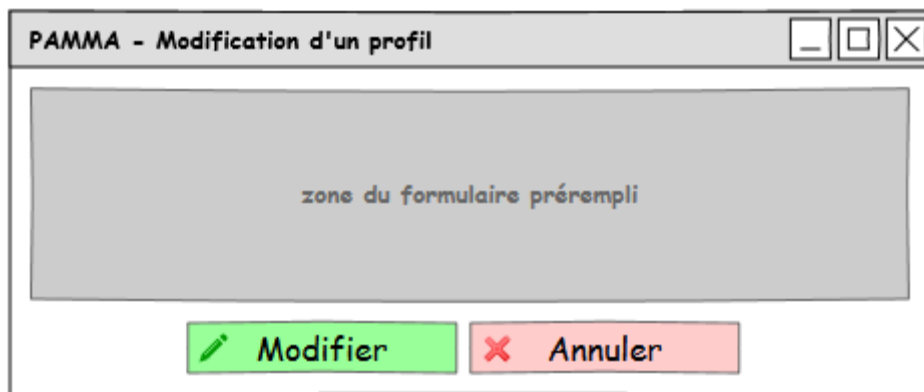


(b)



(c)

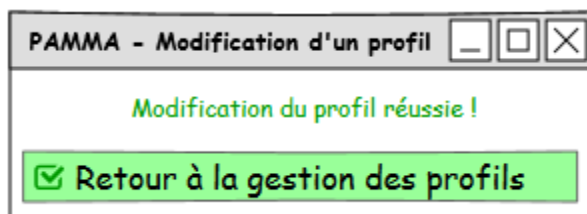
Figure 99 : Prototype d'IHM pour la création d'un profil



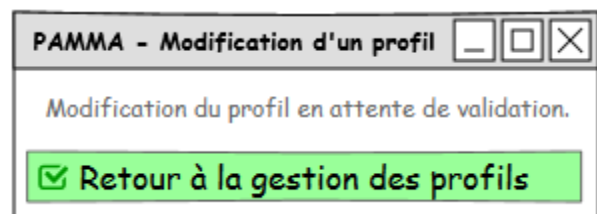
(a)

(si la modification est réussie)

(si la modification est en attente de validation)



(b)



(c)

Figure 100 : Prototype d'IHM pour la modification d'un profil

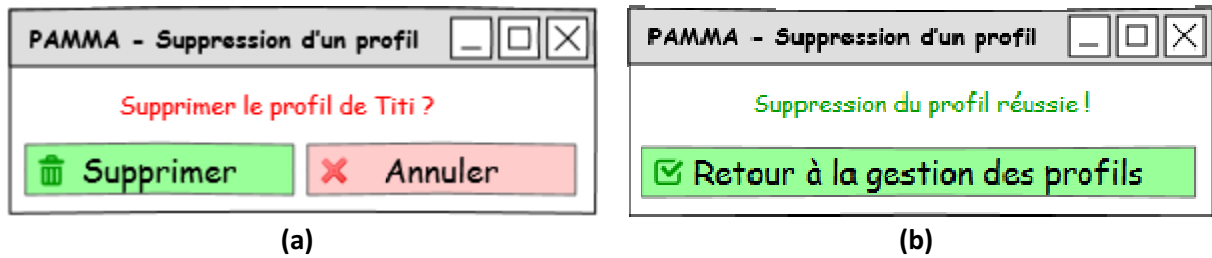


Figure 101 : Prototype d'IHM pour la suppression d'un profil

On peut remarquer que la suppression d'un profil (Figure 101) est précédée d'un écran de confirmation (Figure 101 (a)) destiné à s'assurer que l'utilisateur veut réellement exécuter cette opération. Dans cette même optique de rendre notre interface le plus claire possible pour les utilisateurs, et conformément aux recommandations de Juristo et al. (2006), nous avons introduit des écrans indiquant lorsque le système lance un traitement qui peut prendre quelques secondes (comme, par exemple, dans le cas de l'importation d'un profil (Figure 102 (b)) ou dans celui d'une exportation (Figure 103 (b)). Un code couleur et pictographique est également utilisé pour faciliter la lisibilité de l'IHM. D'autre part, lorsqu'un bouton d'annulation est présent, il permet de revenir à l'écran précédent.

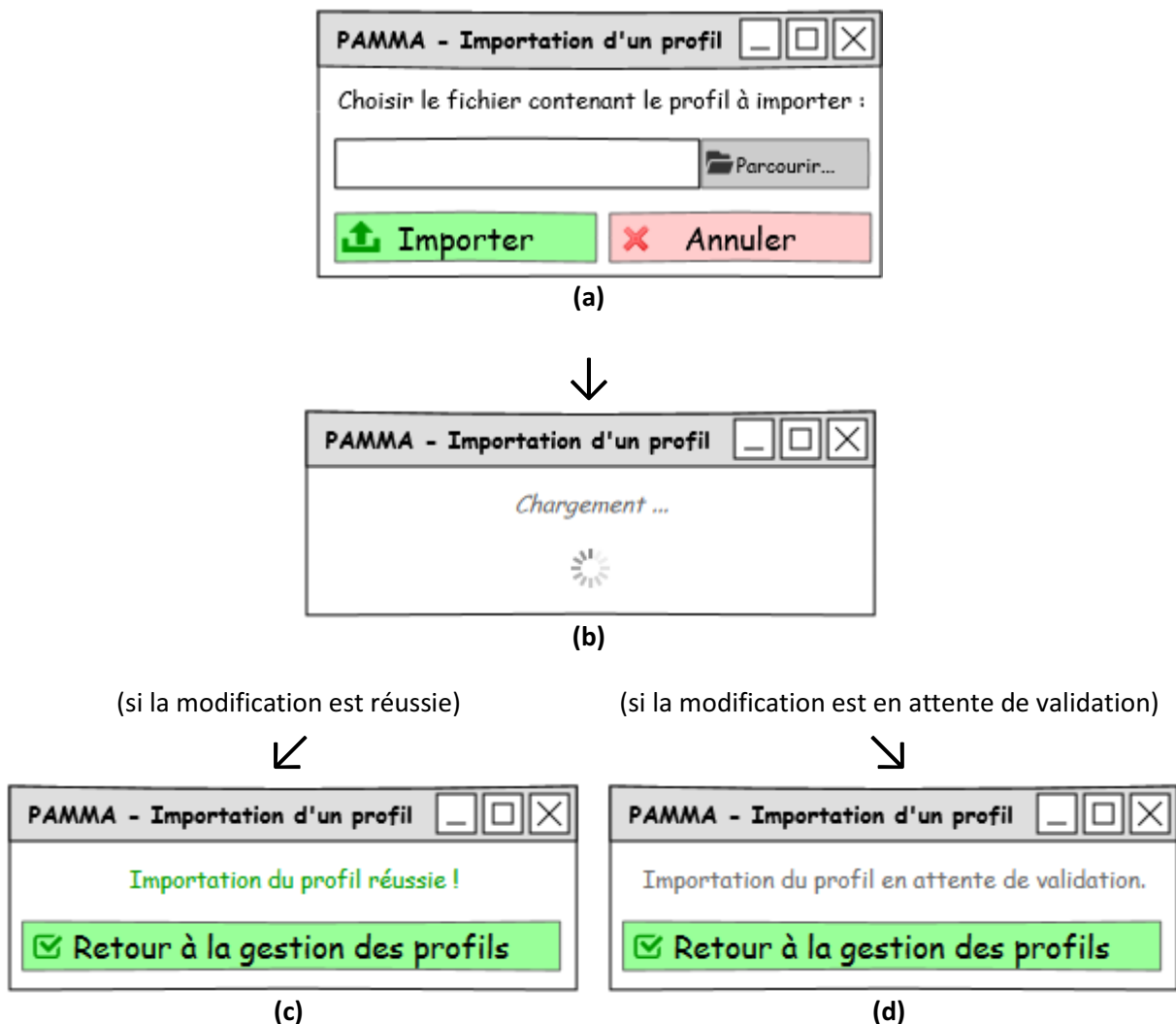


Figure 102 : Prototype d'IHM pour l'importation d'un profil

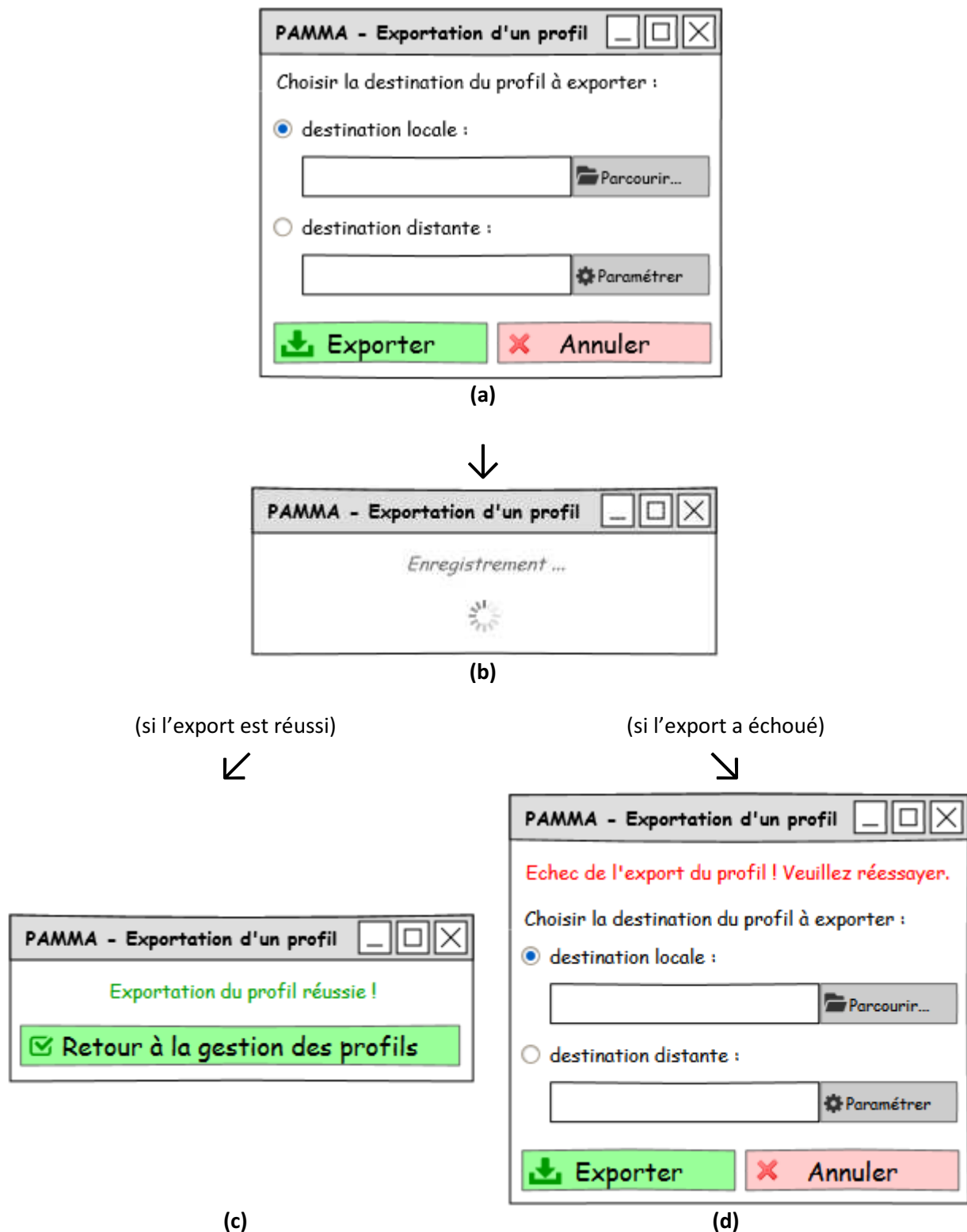


Figure 103 : Prototypage d'IHM pour l'exportation d'un profil

On notera que l'exportation d'un profil vers un espace de stockage distant, tel qu'un FTP ou un Cloud, nécessite un paramétrage (Figure 103 (a)) composé de l'URL du dépôt et des identifiants de connexion à ce dernier. D'autre part, comme nous l'avons déjà signalé, certaines de ces opérations pourront requérir la validation d'un administrateur pour être prises en compte (comme, par exemple, la modification d'un profil (Figure 100 (c))).

Seule l'opération consistant à utiliser un profil (**Figure 98**) va permettre de réellement lancer l'application de traitement de la dyslexie. Nous précisons également que ce lancement ne sera effectif que si l'authentification est un succès (**Figure 98 (b)**). Dans ce dernier cas, l'utilisateur va alors pouvoir accéder à l'accueil de l'application. Comme nous l'avons déjà signalé (voir la sous-section **L'élaboration du schéma de navigation générale**), il existe deux types d'accueil : un pour les professionnels de santé qui correspond à l'accueil du serveur (**Figure 104**) et un pour les patients qui correspond à celui du client (**Figure 105**).

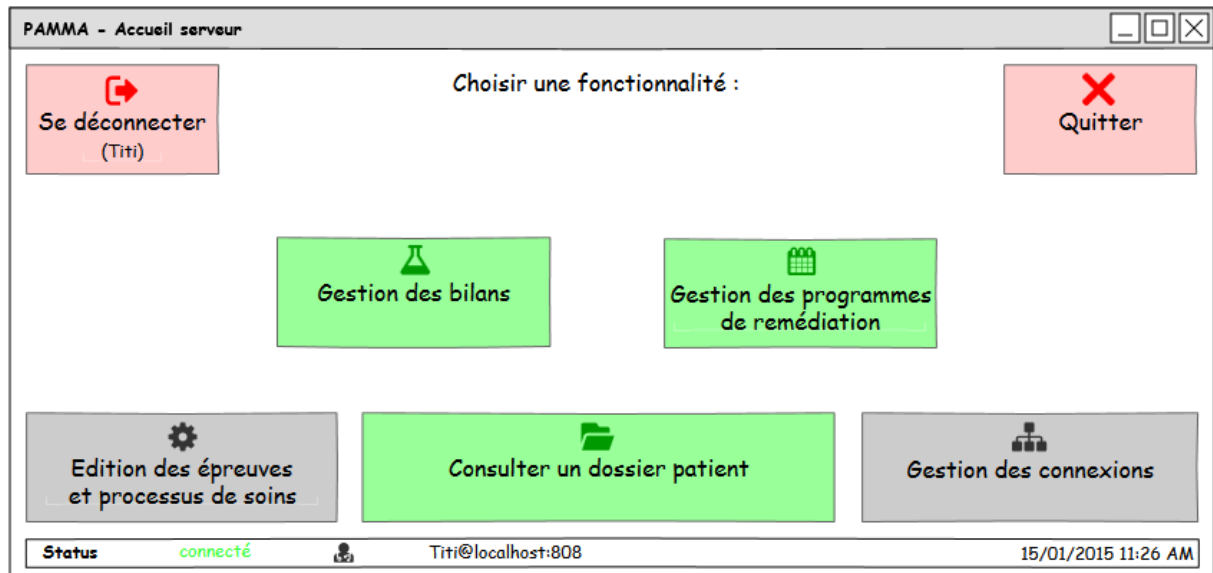


Figure 104 : Prototype d'IHM de l'écran d'accueil du serveur

L'interface de l'accueil du client (**Figure 105**) présente une sorte de restriction des fonctionnalités proposées par celle de l'accueil du serveur (**Figure 104**). Certaines fonctions sont d'ailleurs identiques, telles que la déconnexion du profil (**Figure 106**), qui renvoie en cas de confirmation l'utilisateur à la gestion des profils (**Figure 97**), la gestion des connexions (**Figure 108**) et la fermeture de l'application (**Figure 107**).

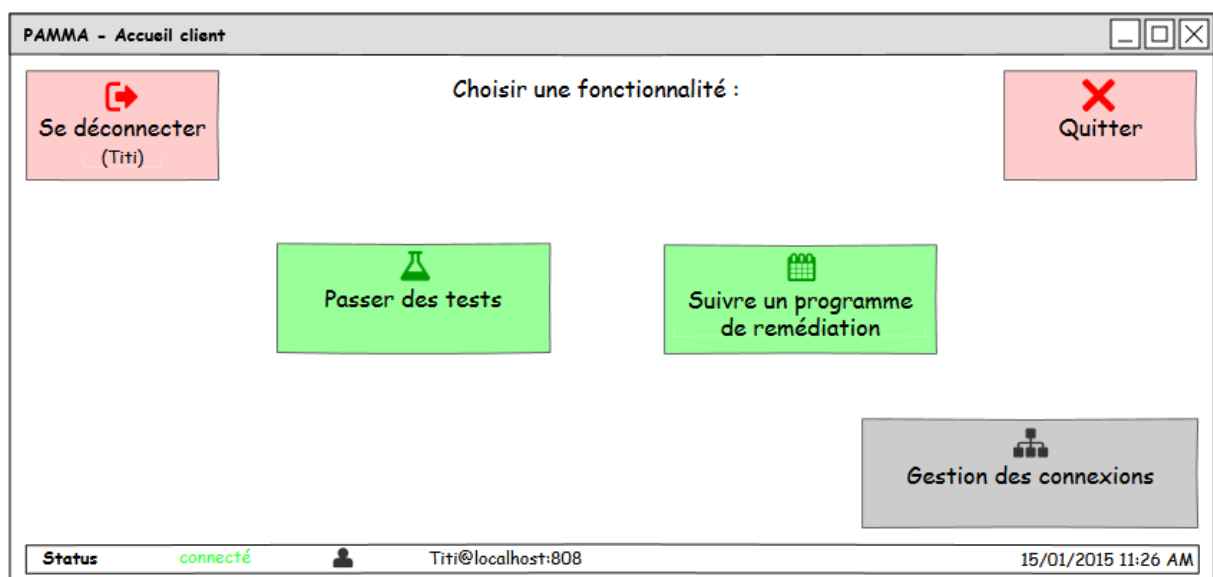


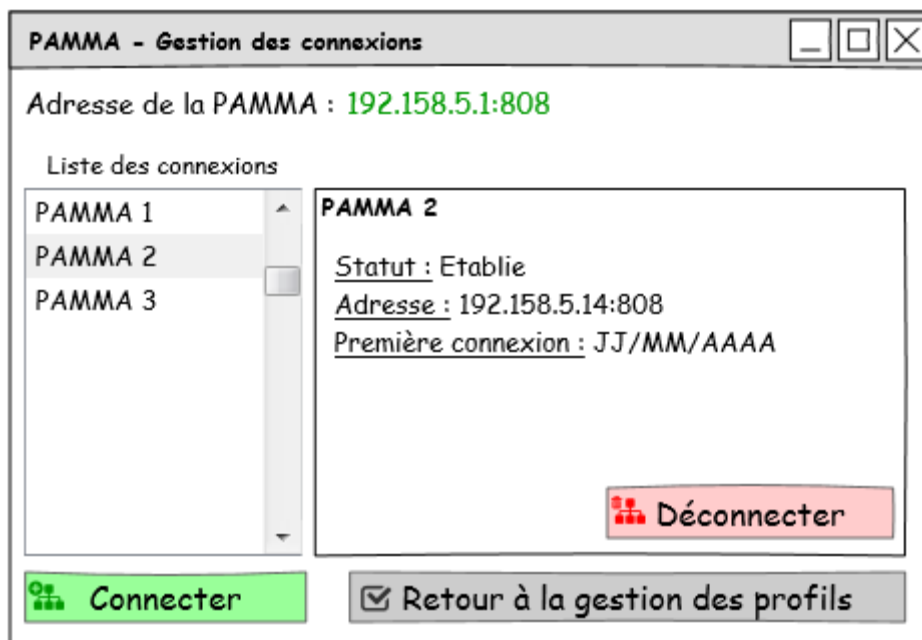
Figure 105 : Prototype d'IHM de l'écran d'accueil du client



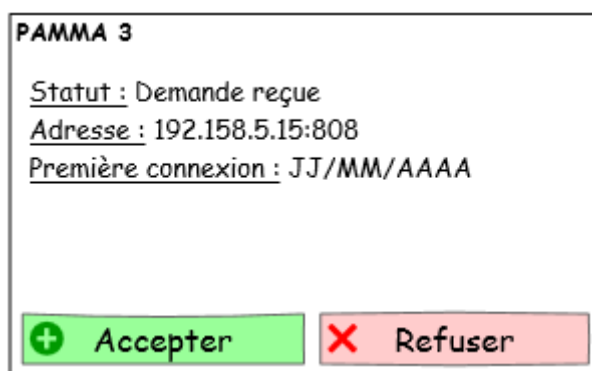
Figure 106 : Prototype d'IHM de la déconnexion du profil en cours d'utilisation



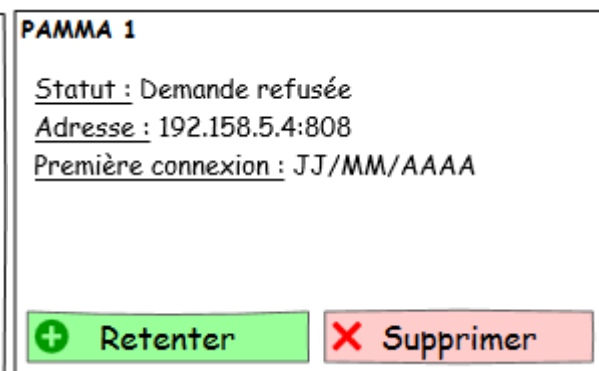
Figure 107 : Prototype d'IHM de la fermeture de l'application



(a)



(b)



(c)

Figure 108 : Prototype d'IHM de gestion des connexions entre PAMMA

La fonction de gestion des connexions (**Figure 108**) permet, au serveur comme au client, d'établir une connexion active avec une autre PAMMA. Deux PAMMA ainsi connectées se retrouvent en mode online (voir la sous-section **L'élaboration du schéma de contexte du domaine**). Les statuts que vont prendre ces connexions pourraient être « demande envoyée », « demande reçue » que l'utilisateur peut accepter ou décliner (**Figure 108 (b)**), « établie » si la demande échangée entre les PAMMA a été acceptée (**Figure 108 (a)**) et « demande refusée » si la cible a décliné la demande (**Figure 108 (c)**). Dans ce dernier cas, il est toujours possible de réexpédier simplement une nouvelle demande (comme représenté sur la **Figure 109 (b)**).

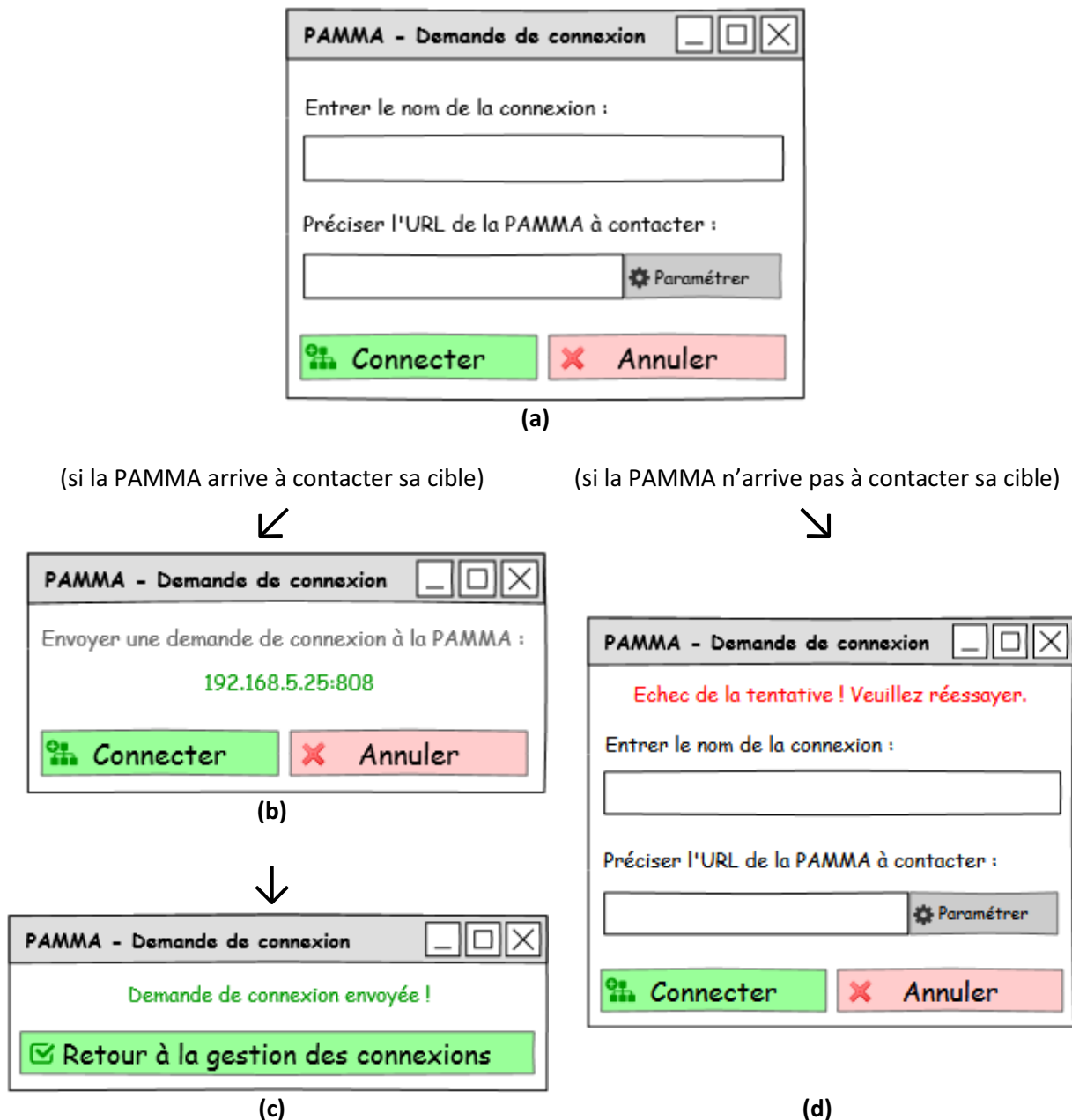


Figure 109 : Prototypé d'IHM de demande de connexion entre PAMMA

Dans le cas de la demande de connexion (**Figure 109**), il faut préciser l'adresse de la PAMMA à contacter. Si le système n'arrive pas à localiser une PAMMA à cette adresse, il invitera l'utilisateur à tenter de nouveaux paramètres de connexion (**Figure 109 (d)**). Dans le cas

d'une déconnexion, l'utilisateur devra confirmer son intention (**Figure 110 (a)**) puis la prise en compte de son choix lui sera notifié (**Figure 110 (b)**).

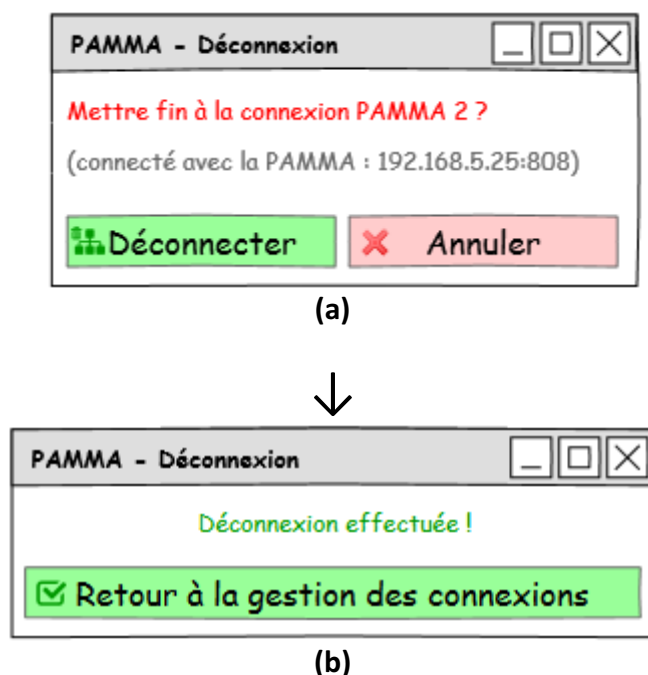
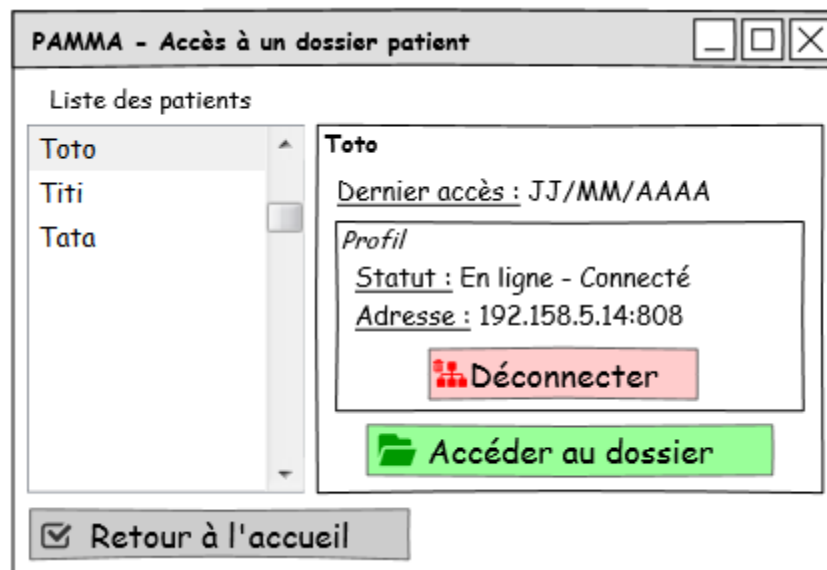


Figure 110 : Prototype d'IHM de clôture d'une connexion entre PAMMA

Nous allons maintenant présenter les fonctionnalités qui ne sont accessibles que pour le serveur : la consultation d'un dossier patient (**Figure 112**), la gestion des bilans (**Figure 113**) et la gestion des programmes de remédiation (**Figure 116**). Chacune de ces fonctions est précédée par un accès à un dossier patient (**Figure 111**). En effet, il faut toujours sélectionner le dossier, et a fortiori le patient auquel il appartient, sur lequel les informations médicales vont être lues et/ou écrites. A ce titre, l'écran d'accès présente une liste des patients traités par le professionnel de santé dont le profil est en cours d'utilisation. Pour chacun de ces patients, il sera possible d'accéder à son dossier, ou le cas échéant de créer ce dossier (**Figure 111 (d)**). D'autre part, pour chacun des patients disposant d'un dossier personnel, il sera possible de savoir si ce patient est actuellement en train d'utiliser un client. Ainsi, un statut précisant l'utilisation du profil du patient pourrait prendre les états « en ligne – connecté » si le patient utilise actuellement une PAMMA connectée à celle du praticien (**Figure 111 (a)**), « en ligne – déconnecté » s'il utilise actuellement une PAMMA mais n'est pas connecté à celle du professionnel (**Figure 111 (b)**) et finalement « hors ligne » s'il n'est pas actuellement en train d'utiliser une PAMMA ou s'il n'est pas connecté à un réseau lui permettant d'être contacté (**Figure 111 (c)**). Sur ce même écran peuvent se trouver des boutons permettant une connexion/déconnexion rapide (**Figure 111 (a)** et **(b)**), ce qui évite de systématiquement devoir retourner à la gestion des connexions (**Figure 108**). On signalera qu'un dossier pourrait ne plus être accessible au praticien, ce qui se traduirait par un message d'erreur lorsque ce dernier tenterait de le consulter (**Figure 111 (e)**).

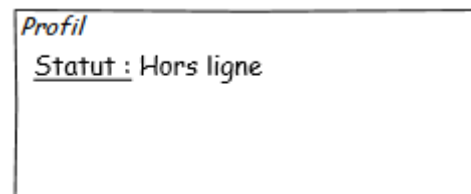
Nous signalons que nous ne détaillerons pas le fonctionnement de l'éditeur dans cette sous-section. Nous reviendrons dessus plus loin dans ce manuscrit (voir la section **L'intégration des processus métiers**).



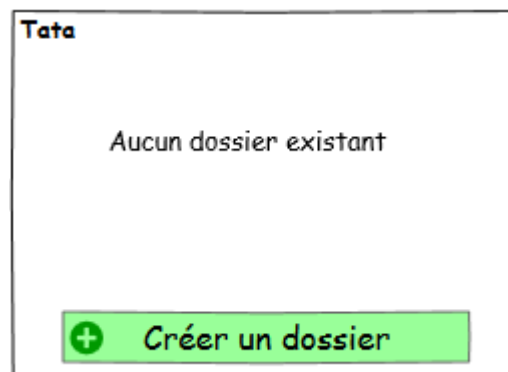
(a)



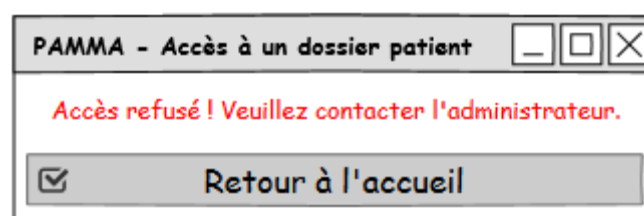
(b)



(c)



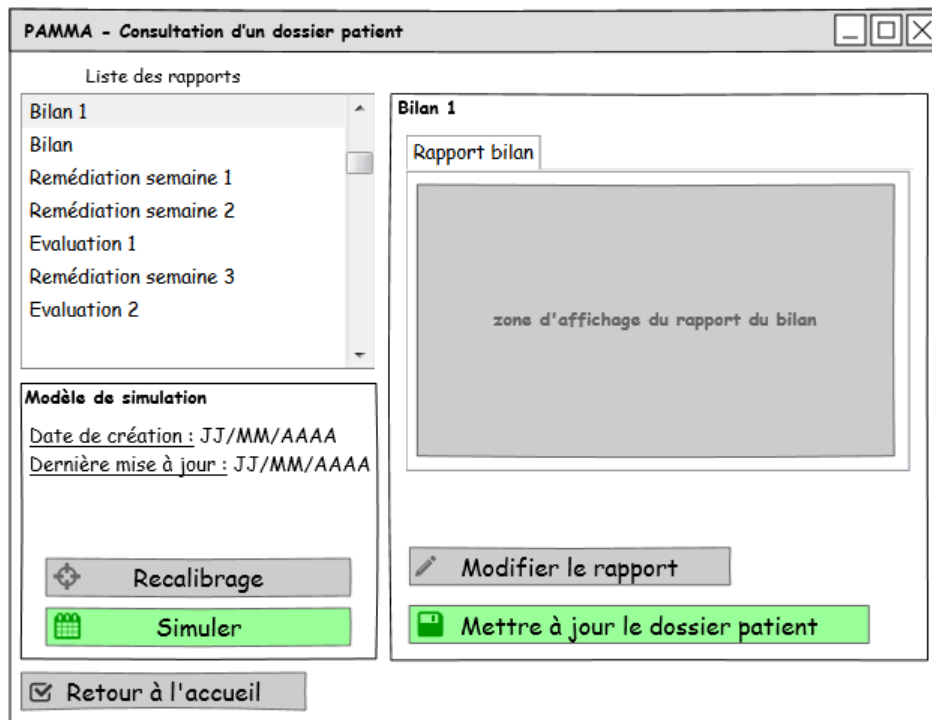
(d)



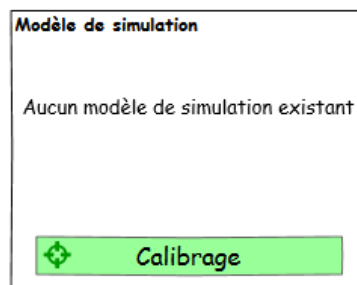
(e)

Figure 111 : Prototype d'IHM pour l'accès à un dossier patient

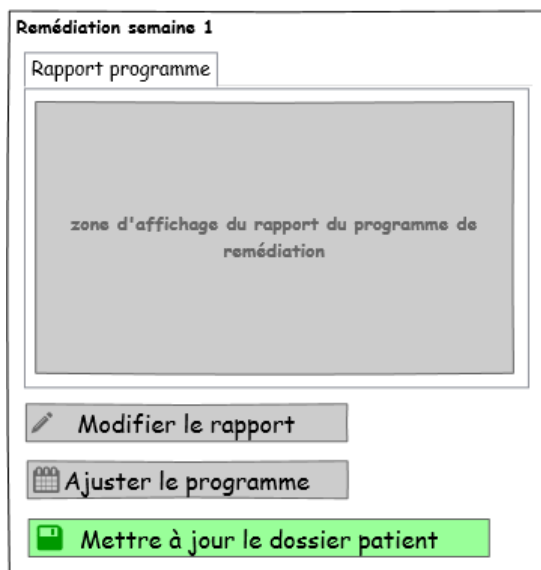
Une fois l'accès au dossier d'un patient, il est possible de le consulter (**Figure 112**). On peut ainsi consulter les différents rapports, les éditer si besoin (**Figure 112 (a), (c) et (d)**), calibrer (ou recalibrer) le modèle de simulation et l'exécuter sur demande (**Figure 112 (a) et (b)**).



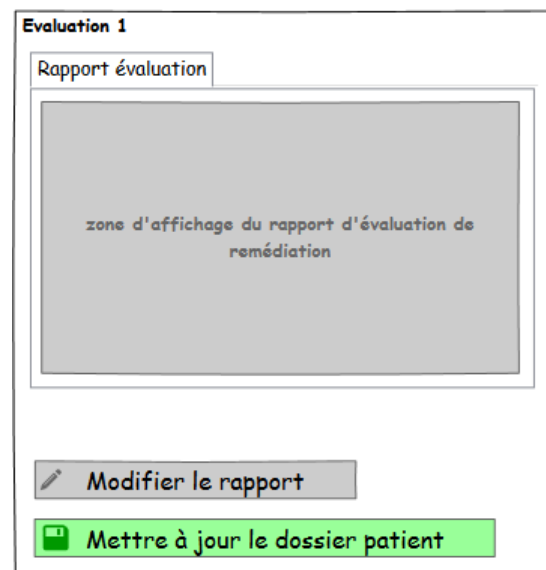
(a)



(b)

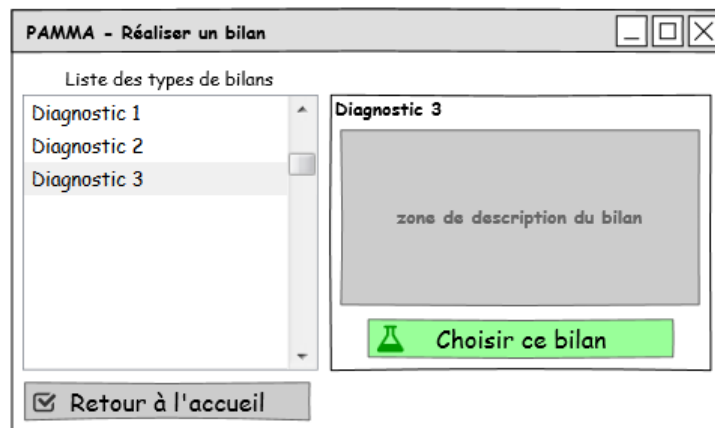


(c)

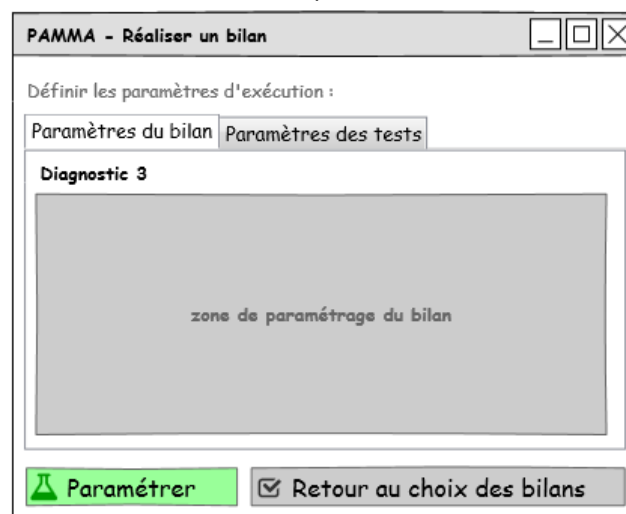


(d)

Figure 112 : Prototypé d'IHM pour la consultation d'un dossier patient



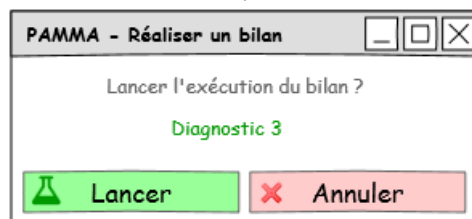
(a)



(b)



(c)



(d)

Figure 113 : Prototype d'IHM pour la réalisation de bilans

La réalisation des bilans va consister, pour le praticien, à choisir un bilan (**Figure 113 (a)**), le paramétrer (**Figure 113 (b)**), transmettre ces paramètres au client (**Figure 113 (c)**) et finalement confirmer ces choix avant exécution (**Figure 113 (d)**). Les tests constitutifs du bilan retenu sont ensuite présentés sur le client (**Figure 114 (a)** et **(b)**) pendant que le serveur a la possibilité de piloter cette exécution (**Figure 114 (c)**). Les résultats obtenus durant les tests apparaissent au fur et à mesure sur l'écran du professionnel de santé et, lorsque tous les tests ont été passés, le bouton « analyser et conclure » s'affiche. Ce bouton permet d'obtenir le rapport du bilan ainsi qu'une aide diagnostique (**Figure 115 (a)**). Le praticien va alors pouvoir demander des examens complémentaires et poser le diagnostic.

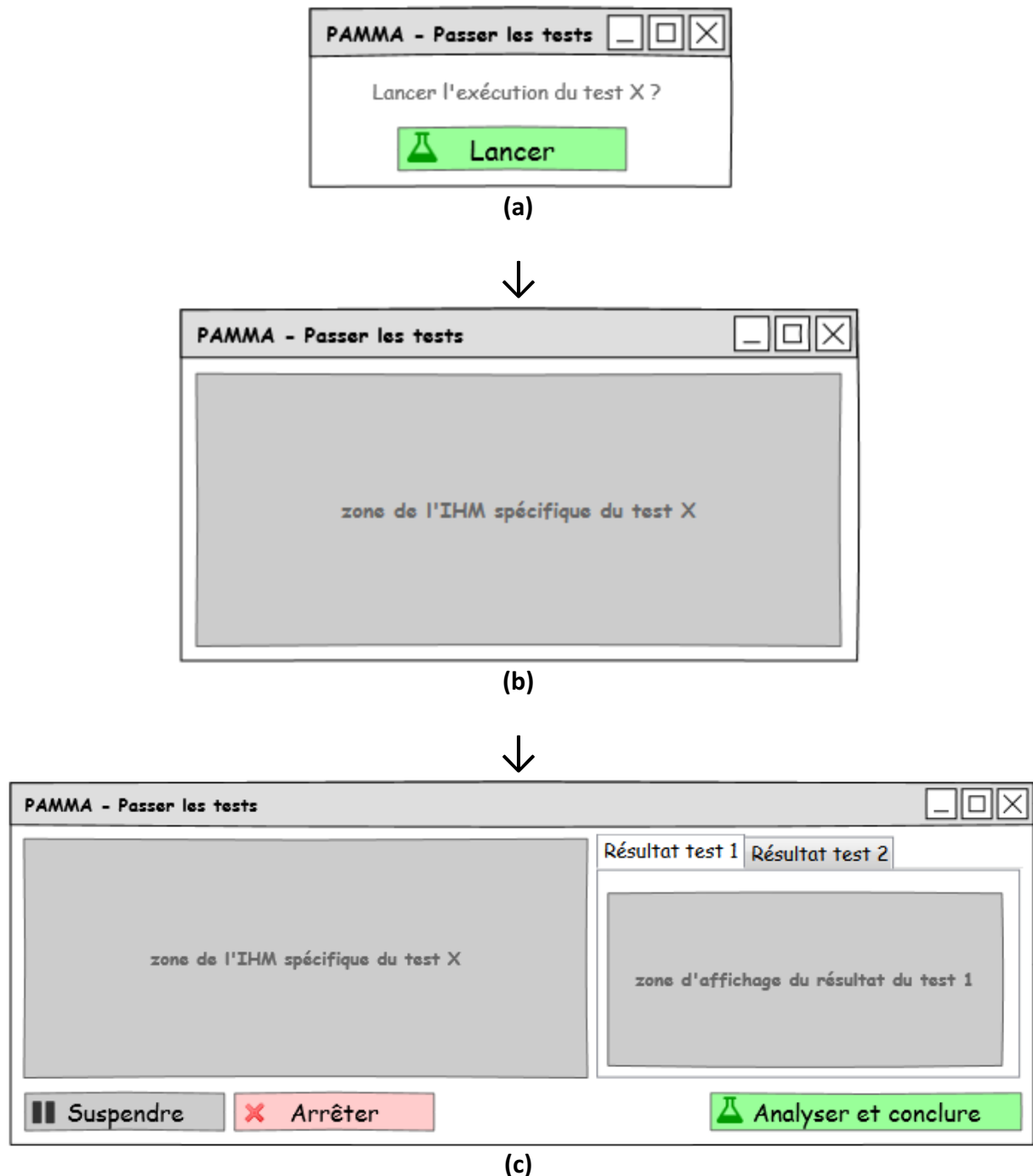
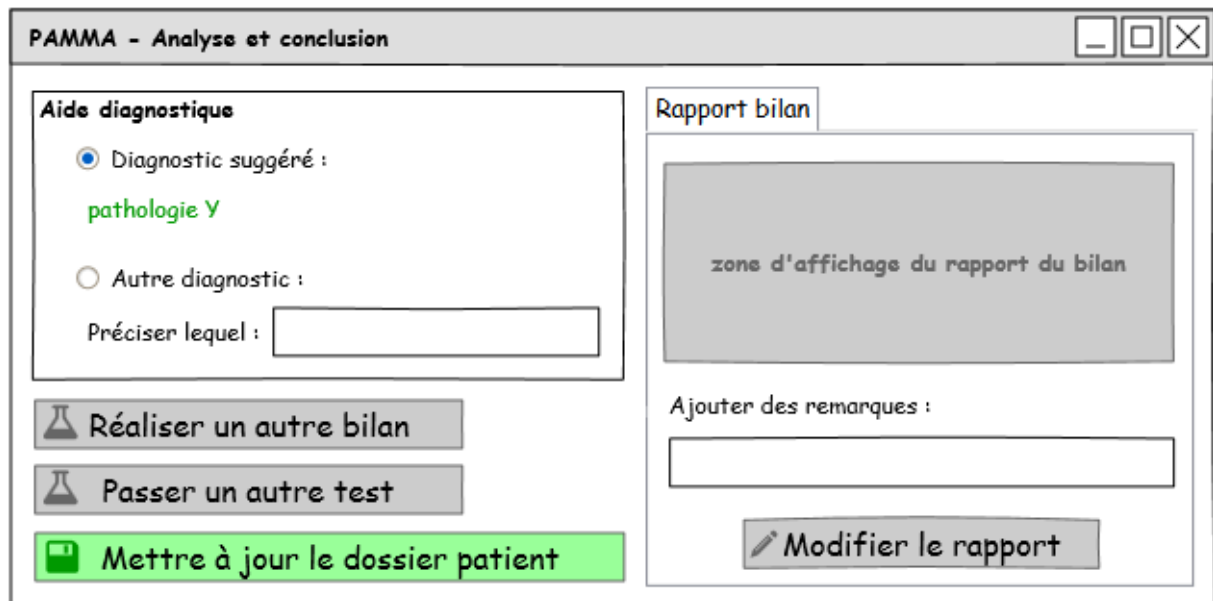
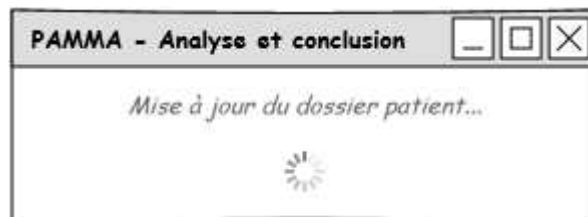


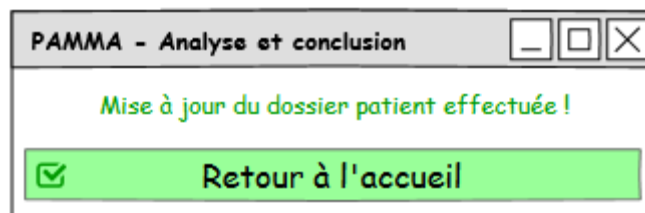
Figure 114 : Prototype d'IHM pour passer les tests d'un bilan



(a)



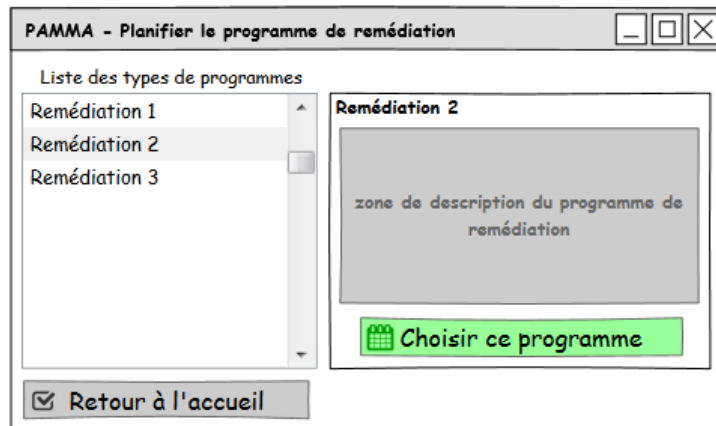
(b)



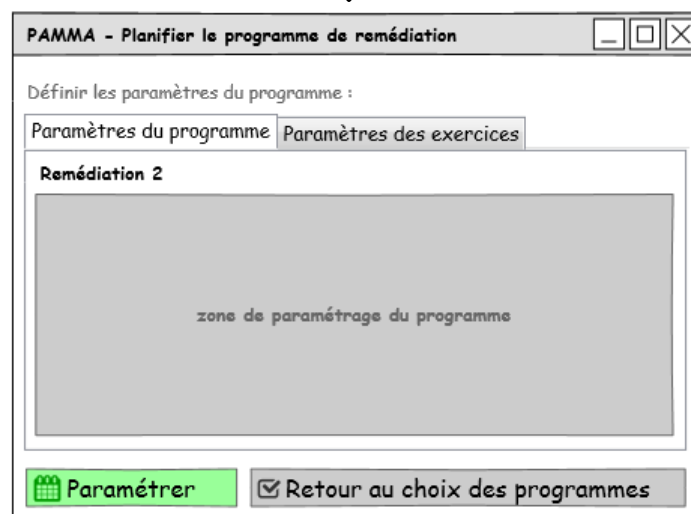
(c)

Figure 115 : Prototype d'IHM pour l'analyse et la conclusion d'un bilan

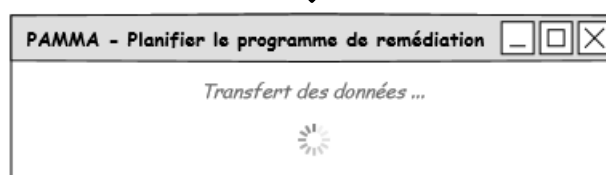
D'une manière similaire aux bilans, la planification des programmes de remédiation va consister, pour le praticien, à choisir un programme (**Figure 116 (a)**), le paramétrer (**Figure 116 (b)**), transmettre ces paramètres au client (**Figure 116 (c)**) et finalement confirmer ces choix avant planification (**Figure 116 (d)**). Les exercices constitutifs du programme retenu seront présentés sur le client (**Figure 117 (a)** et **(b)**) pendant que le serveur a la possibilité de les visualiser si les PAMMA sont connectées (**Figure 117 (c)**). Les résultats obtenus durant les exercices apparaissent au fur et à mesure sur l'écran du professionnel de santé et, lorsque tous les exercices ont été passés, le bouton « analyser les résultats » s'affiche. Ce bouton permet d'obtenir le rapport du programme ainsi qu'à ajuster ce dernier (**Figure 119 (a)**). Le praticien peut également attacher un rapport d'évaluation au dossier du patient.



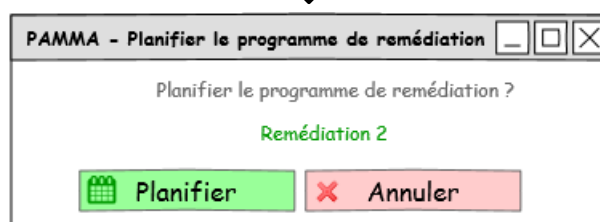
(a)



(b)



(c)



(d)

Figure 116 : Prototypage d'IHM pour la planification d'un programme de remédiation

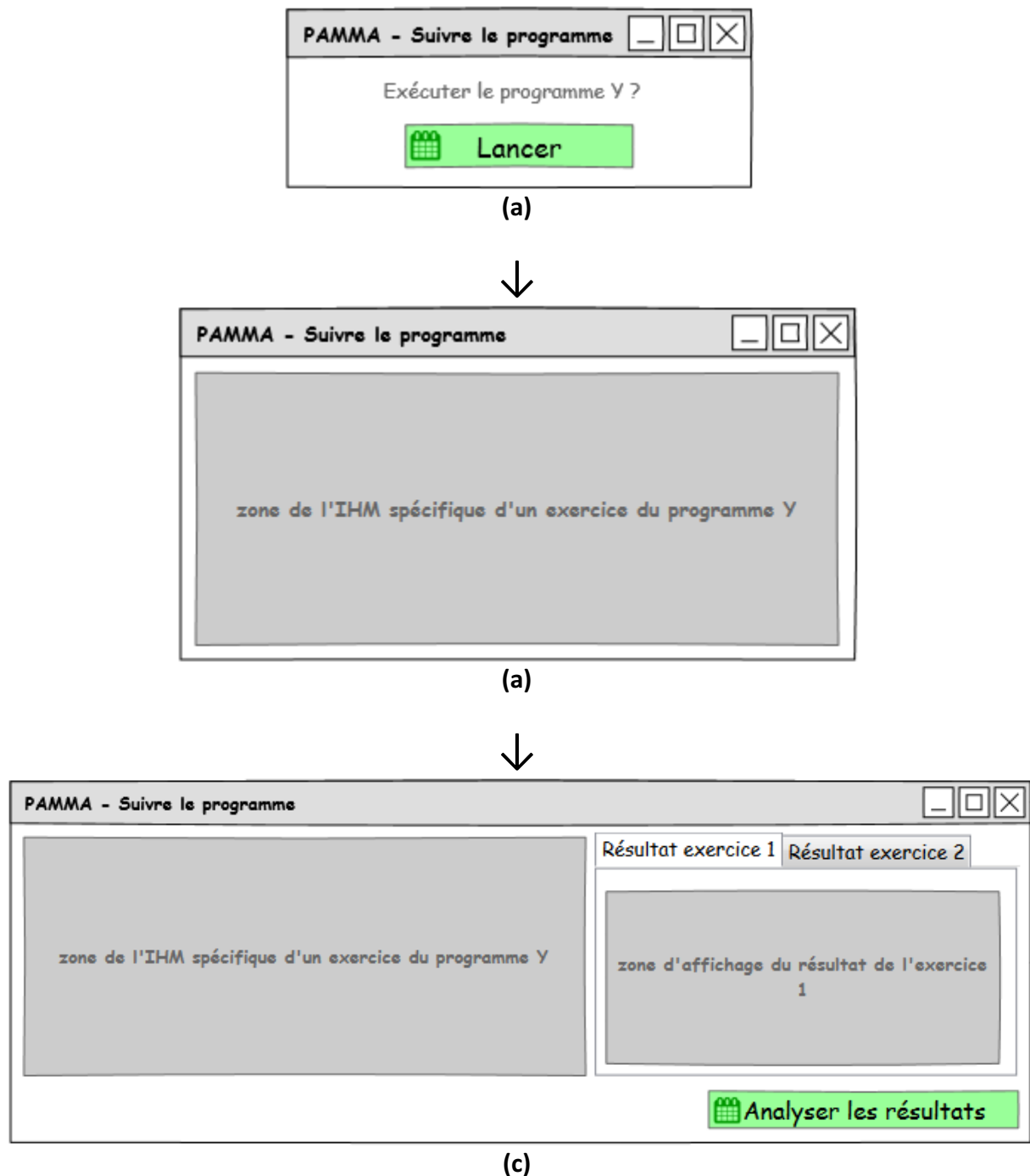


Figure 117 : Prototype d'IHM pour la réalisation des exercices d'un programme de remédiation

Afin d'évaluer l'efficacité du programme de remédiation, le professionnel de santé peut simuler le suivi du programme de remédiation à l'aide d'un modèle de simulation préalablement calibré (**Figure 111 (a)**). Cette simulation, qui s'effectue sans la présence du client, va produire des résultats (**Figure 118**) qui pourront être analysés de la même manière que ceux générés par l'action du patient. Comme pour l'exécution de tests (**Figure 114 (a)**) ou d'exercices (**Figure 117 (a)**), un écran de temporisation précède l'exécution en tant que telle (**Figure 118 (a)**). Cet écran permet à l'utilisateur de prendre les dispositions qui s'imposent avant le lancement de ce traitement gourmand en ressources pour la machine

hébergeant la PAMMA. De telles dispositions sont typiquement des mesures de suspension d'autres processus dont le fonctionnement, en parallèle de celui demandé par l'utilisateur, pourrait provoquer des effets de ralentissement.

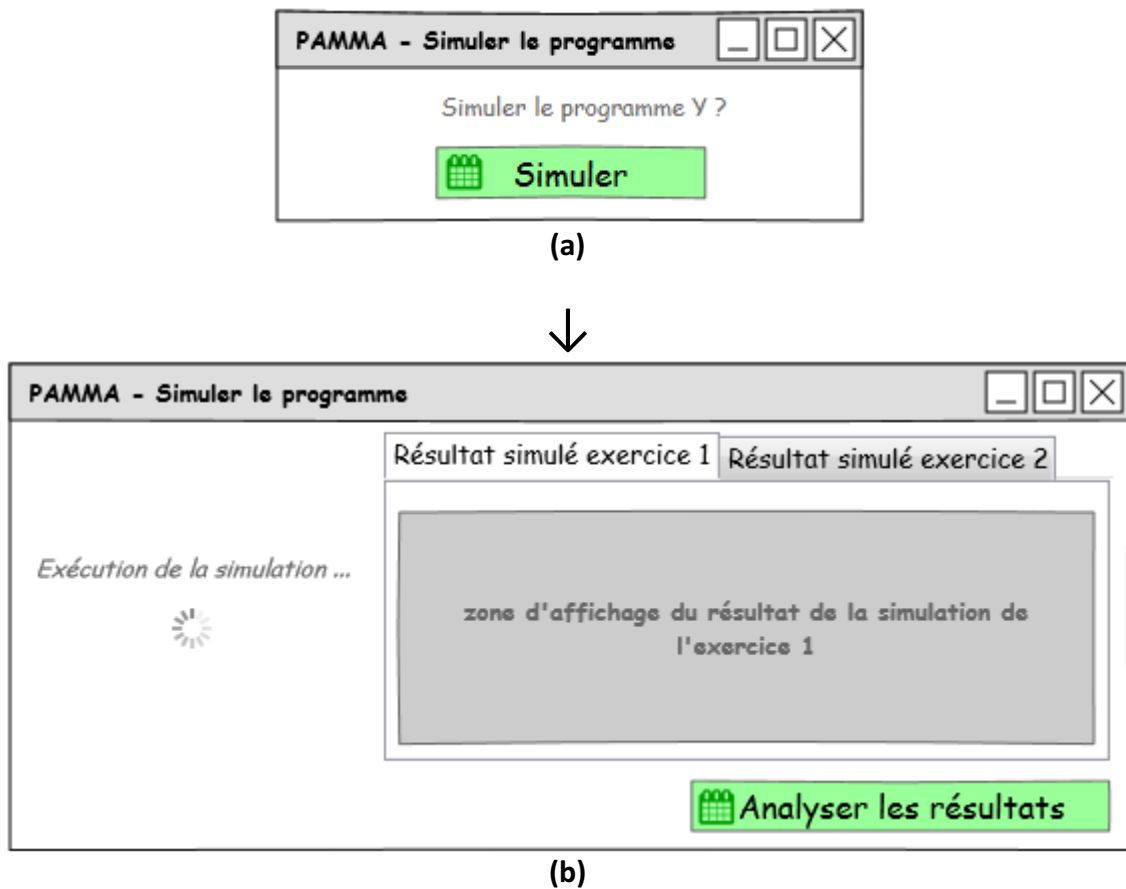
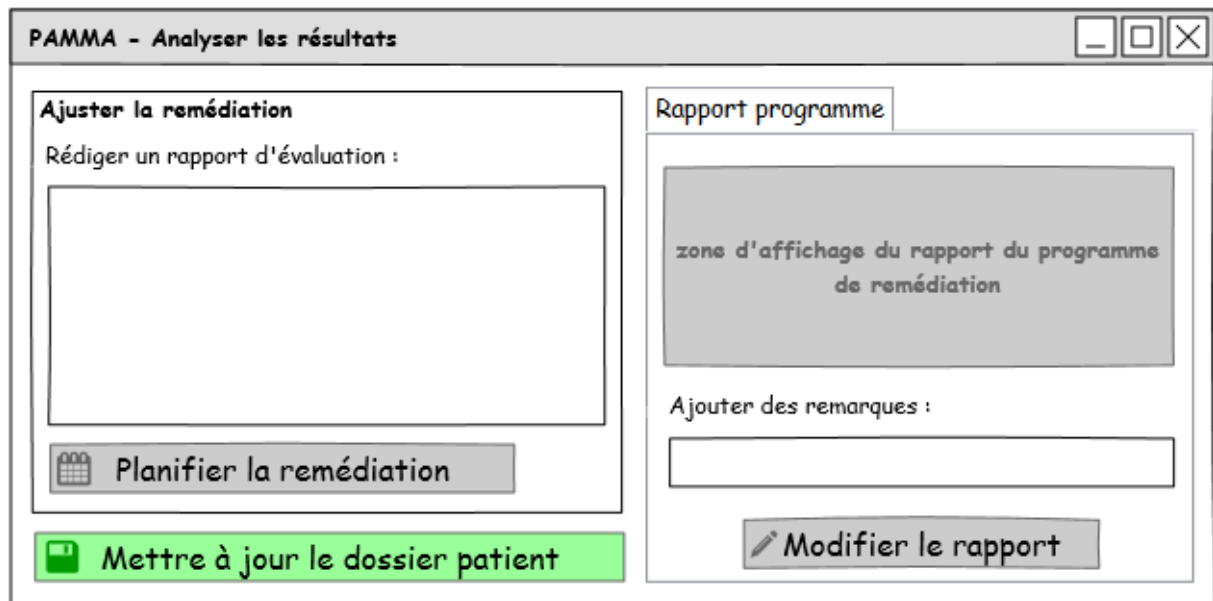


Figure 118 : Prototype d'IHM pour la simulation des exercices d'un programme de remédiation

Pour conclure avec le maquetage d'IHM, on précise que ces différentes interfaces pourraient bénéficier des précisions et améliorations suivantes :

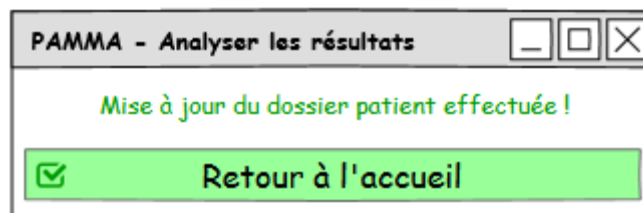
- Chaque liste pourrait bénéficier d'un champ de recherche pour en faciliter l'utilisation. Un tel champ permet à l'utilisateur d'effectuer rapidement une sélection des éléments de la liste.
- La liste des rapports disponibles lors de la consultation d'un dossier patient doit pouvoir être représentée sous la forme d'une frise chronologique.
- Des boutons d'aide contextuelle pourraient être présents au moins sur chaque écran requérant un choix de la part de l'utilisateur. Une telle aide a pour but d'expliquer précisément toutes les fonctionnalités et possibilités offertes à l'utilisateur.
- La « croix » dans la barre de titre de chaque écran doit pouvoir émuler contextuellement l'action des boutons « annuler », « retour » ou « quitter ».
- La réalisation d'un processus de soin, indépendamment du fait que ce soit un bilan ou un programme de remédiation, devrait s'accompagner, du côté serveur, d'une représentation graphique de ce processus. Une telle représentation permet d'explicitier l'intégralité de l'exécution de ce processus et d'indiquer à l'utilisateur l'état de chacune des étapes le composant (voir la sous-section **L'exécution**).



(a)



(b)



(c)

Figure 119 : Prototype d'IHM pour l'analyser des résultats d'un programme de remédiation

6) L'élaboration des diagrammes de classe

FICHE GUIDE – FG12	
Activité 3 : Analyse des cas d'utilisation	
Sous-activité 3.6 : Elaboration des diagrammes de classe (DCL)	
Objectif	Définir les classes pour chaque cas d'utilisation
Point de départ	Interactions entre acteurs et objets du système définies pour tous les cas d'utilisation et concepts métiers définis dans la modélisation métier
Point d'arrivée	Classes et associations définies pour chaque cas d'utilisation dans les DCL par cas d'utilisation
Démarche d'élaboration	
Pour chaque cas d'utilisation (CU) :	
<p>1 - Identifier les classes du CU en prenant comme base les objets définis dans le diagramme de séquence du CU et les concepts métiers.</p> <p>2 - Préciser les attributs des classes à partir de ceux identifiés dans le DSE (paramètres des messages) et des « entrées » de l'IHM.</p> <p>3 - Préciser les opérations des classes à partir des messages du DSE et des actions et activités du DET. Un message « entrant » d'un objet correspond à une opération de la classe sollicitée.</p> <p>4 - Déterminer les relations entre les classes :</p> <ul style="list-style-type: none"> - nom de l'association - multiplicité - type d'association (composition, agrégation, association qualifiée, dépendance, héritage...) 	

Tableau 19 : Fiche Guide 12 pour UP7 (d'après (Gabay & Gabay, 2008))

Pour tous les diagrammes de classe suivants, nous avons regroupé ensemble les cas qui manipulent les mêmes objets et qui sont similaires d'un point de vue fonctionnel. Pour chaque diagramme, nous préciserons quels sont ces regroupements.

Notre premier diagramme de classe concerne la précondition aux cas d'utilisation « Installation et activation du système » (**Figure 120**). Ce diagramme représente les classes impliquées dans le diagramme de séquence du même nom (**Figure 65**). On peut apporter quelques précisions au sujet de l'attribut « étatInstallation » de la classe « Ordinateur » : cet attribut modélise l'état de l'installation sur la machine qui héberge la PAMMA et pourrait prendre les valeurs suivantes : « aucune », « fichiers déployés », « fichiers configurés » et « PAMMA installée ».

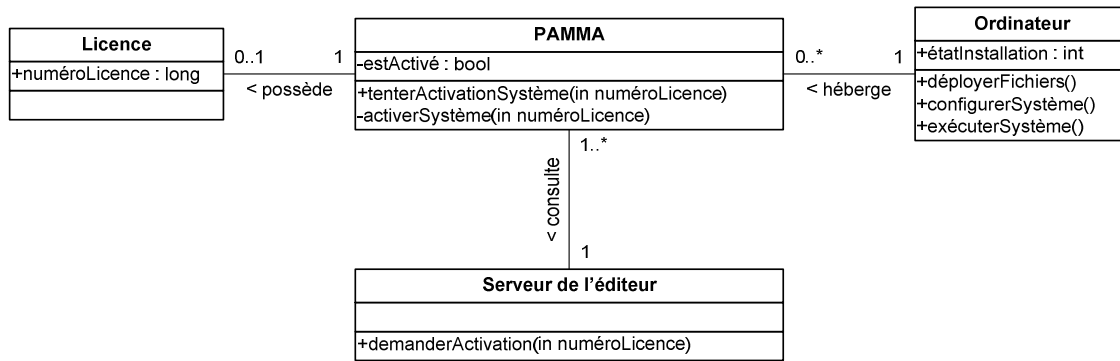


Figure 120 : Diagramme de classe de la précondition aux cas d'utilisation « Installation et activation du système »

Le diagramme suivant est celui de la précondition aux cas d'utilisation « Authentification de l'utilisateur » (**Figure 121**). Comme dans le cas précédent, ce diagramme ne reprend que les éléments du diagramme de séquence éponyme (**Figure 66**). Par ailleurs, nous avons ajouté des attributs à la classe « Session » pour lui donner plus de consistance : « id » est l'identifiant unique de la session et « clef » est sa clef de cryptage privée. Nous reviendrons plus loin dans ce manuscrit sur les aspects liés à la sécurité du système (voir sous-section **La sécurité**).

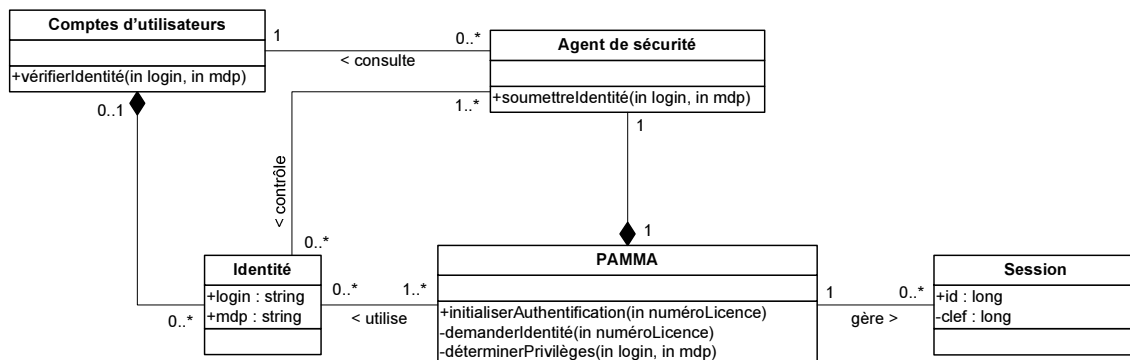


Figure 121 : Diagramme de classe de la précondition aux cas d'utilisation « Authentification de l'utilisateur »

Nous avons représenté ensemble toutes les classes de la précondition aux cas d'utilisation « Accéder à un dossier patient » et du cas d'utilisation 18 « Consultation d'un dossier patient » (**Figure 122**). Effectivement, cette précondition est indispensable à ce cas d'utilisation et elle vise à en permettre la manipulation des classes que ce cas requiert. Bien que nous n'ayons pas voulu complexifier ce diagramme de classe, il se pourrait qu'à l'implémentation il s'avère pertinent d'impliquer l'agent de sécurité précédemment présenté (**Figure 121**) dans l'accès à un dossier patient. Nous rappelons également que cette précondition est requise pour les cas d'utilisation 3 « Réaliser un bilan » et 7 « Planifier le programme de remédiation ».

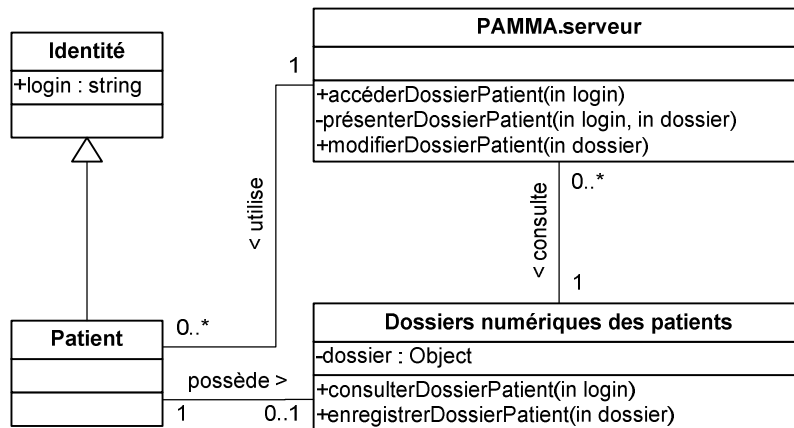


Figure 122 : Diagramme de classe de la précondition aux cas d'utilisation « Accéder à un dossier patient » et du cas d'utilisation 18 « Consultation d'un dossier patient »

Naturellement, nous avons regroupé les classes des cas d'utilisation 1 « Définir des processus de soins » et 2 « Définir des épreuves » (**Figure 123**). Les classes métiers ont été rassemblées sous la dénomination « Descriptions des applications », ce qui représente, par exemple, une base de données contenant toutes les données indispensables à l'instanciation des classes métiers que nous avons déjà décrites (**Figure 45** et **Figure 46**). Il est ainsi possible d'établir une relation de composition, que nous n'avons pas représentée ici pour ne pas inutilement alourdir ce diagramme, entre les classes métiers et les « Descriptions des applications ». Cette relation apparaîtra sur notre diagramme de classe de synthèse (voir la sous-section **L'élaboration du diagramme de classe récapitulatif**).

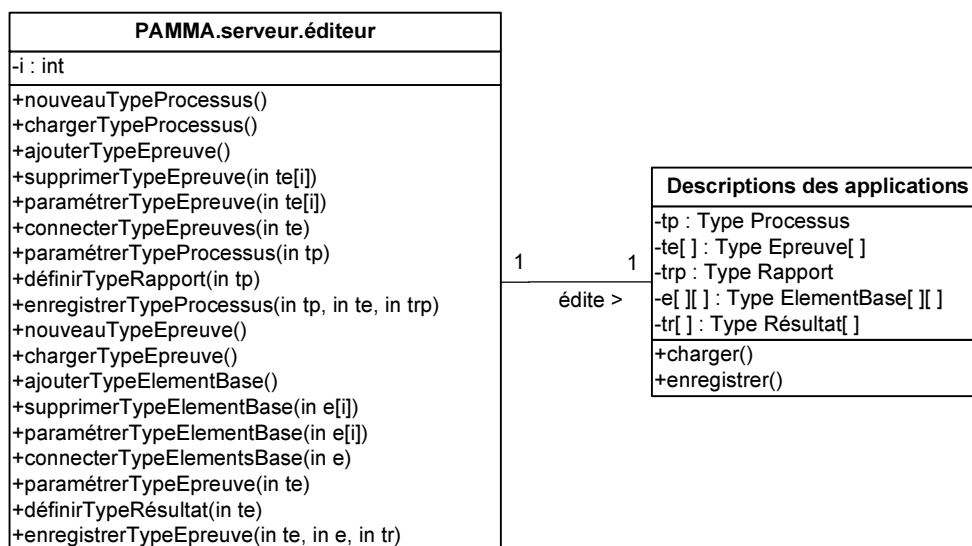


Figure 123 : Diagramme de classe des cas d'utilisation 1 « Définir des processus de soins » et 2 « Définir des épreuves »

La réalisation complète d'un bilan n'est pas exclusivement constituée du cas d'utilisation 3 « Réaliser un bilan » mais également des cas 4 « Passer les tests » et 5 « Analyse et conclusion ». Nous avons choisi d'ajouter à ces classes celles du cas d'utilisation 6

« Calibrage de la simulation » qui s'appuient directement sur les données produites par les trois cas précédemment cités (**Figure 124**).

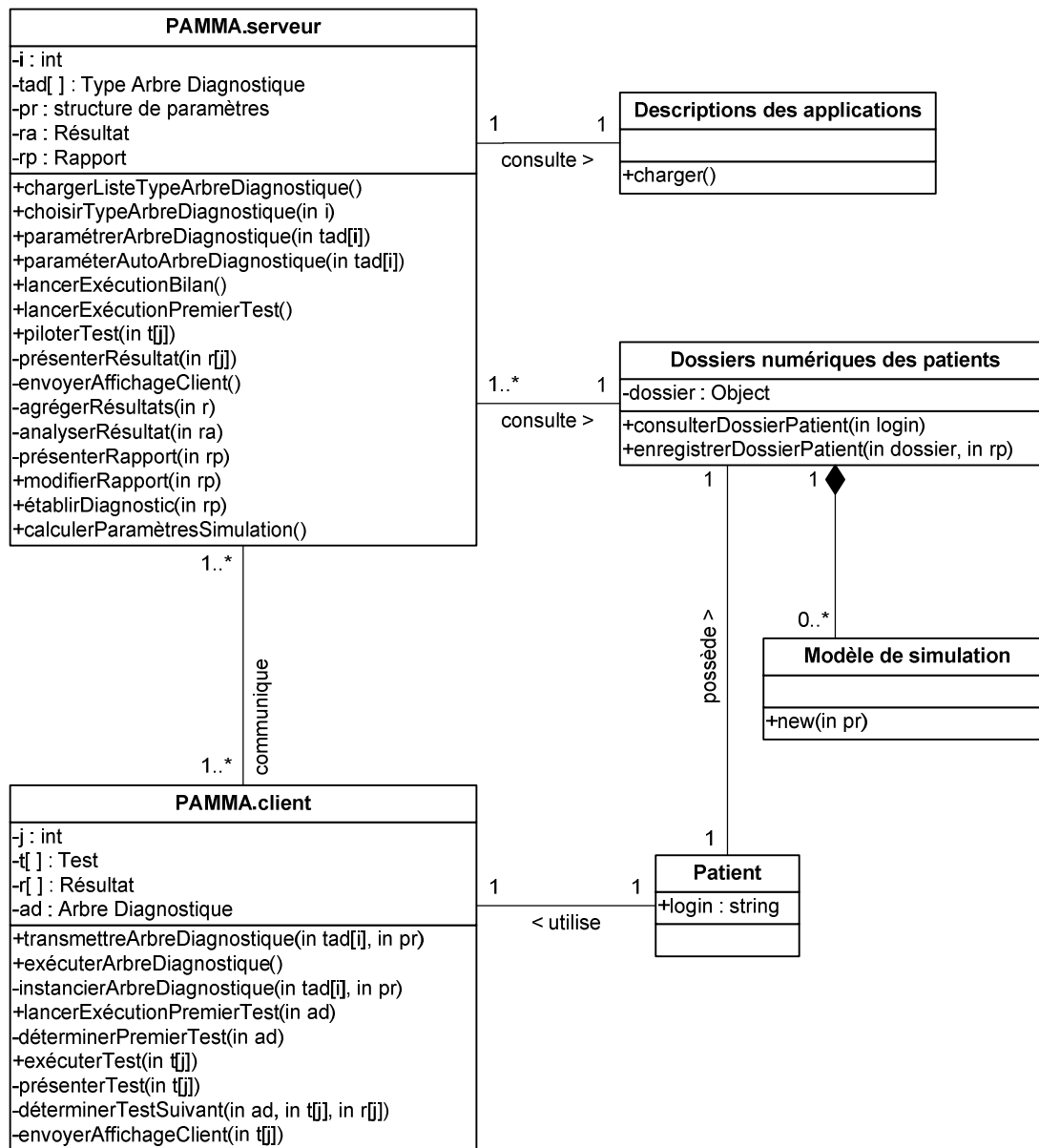


Figure 124 : Diagramme de classe des cas d'utilisation de 3 « Réaliser un bilan » à 6 « Calibrage de la simulation »

Comme pour notre précédent diagramme de classe, le suivi complet d'un programme de remédiation n'est pas exclusivement composé du cas d'utilisation 7 « Planifier le programme de remédiation » mais également des cas 8 « Suivre le programme », 9 « Simuler le suivi du programme », 10 « Analyser les résultats » et 11 « Ajuster la remédiation ». Nous avons choisi d'ajouter à ces classes celles du cas d'utilisation 12 « Recalibrage de la simulation » (**Figure 125**).

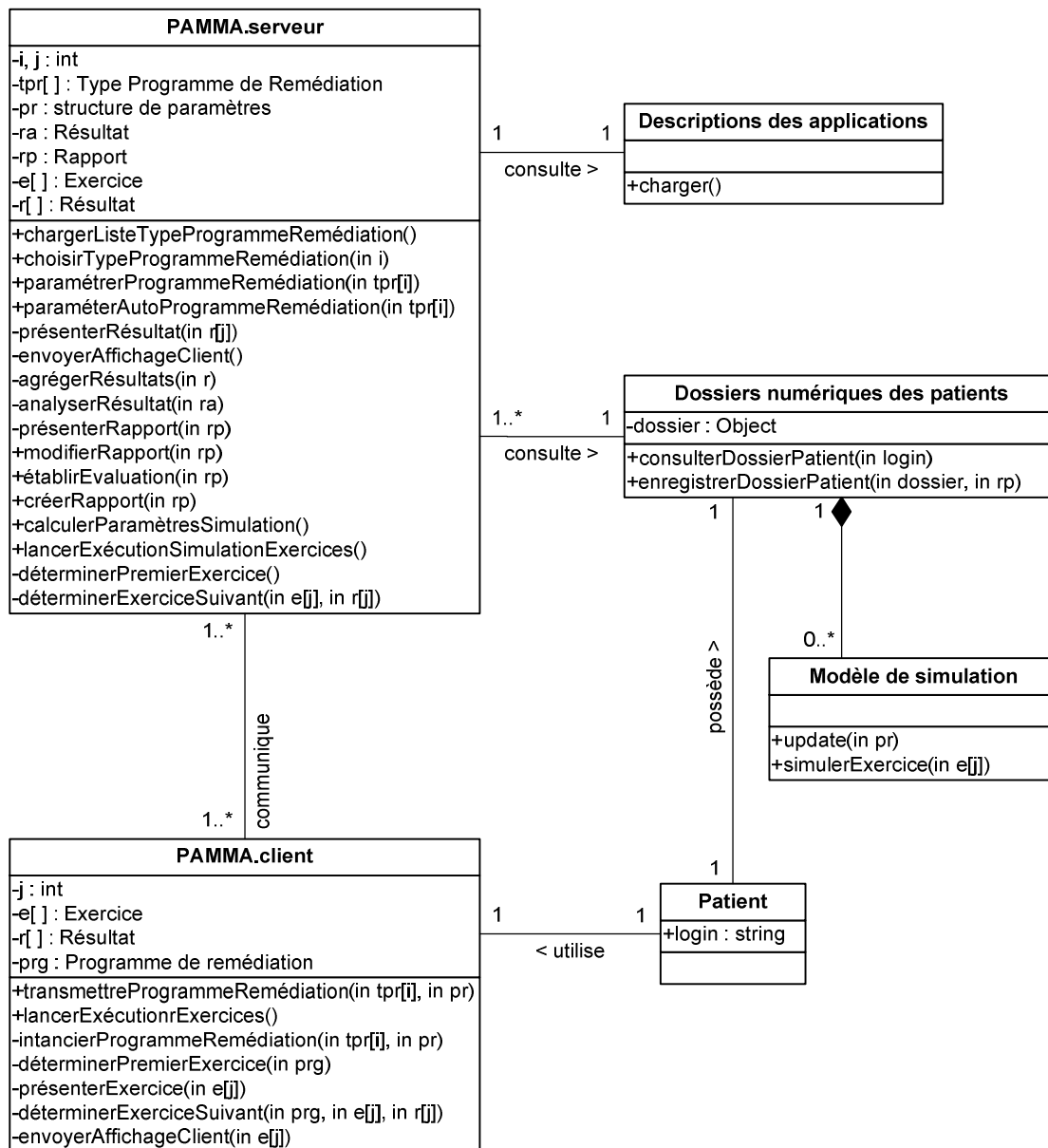


Figure 125 : Diagramme de classe des cas d'utilisation de 7 « Planifier le programme de remédiation » à 12 « Recalibrage de la simulation »

Le diagramme suivant représente l'ensemble des cas d'utilisation de gestion et d'utilisation des profils d'utilisateurs, c'est-à-dire les cas d'utilisation 13 « Création d'un profil », 14 « Modification d'un profil », 15 « Suppression d'un profil », 16 « Import/Export d'un profil » et 17 « Connexion/Déconnexion d'un profil » (**Figure 126**). On notera que les éléments pour la validation d'une action sur un profil par un administrateur du système sont également intégrés au diagramme.

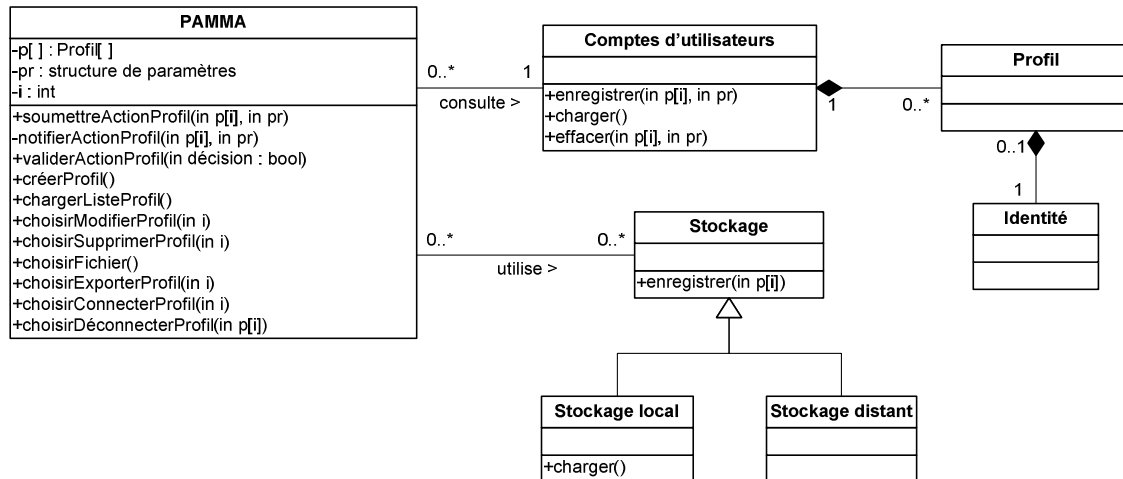


Figure 126 : Diagramme de classe de la précondition aux cas d'utilisation « Valider une action sur un profil » et des cas d'utilisation de 13 « Création d'un profil » à 17 « Connexion/Déconnexion d'un profil »

Notre dernier diagramme de classe présente les cas d'utilisation de connexion/déconnexion entre PAMMA (**Figure 127**). A ce titre, on été regroupés les cas d'utilisation 19 « Connexion client-serveur » et 20 « Déconnexion client-serveur ».

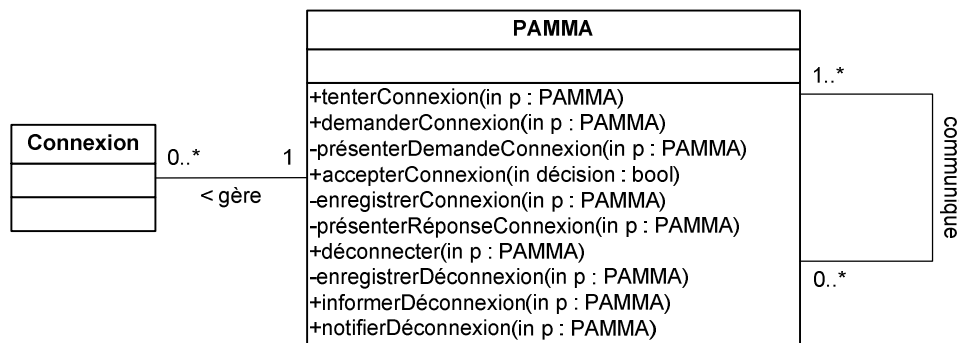


Figure 127 : Diagramme de classe des cas d'utilisation 19 « Connexion client-serveur » et 20 « Déconnexion client-serveur »

D) La synthèse de l'analyse

La quatrième activité de l'UP7 consiste à affiner l'analyse des cas d'utilisation précédemment réalisée. Elle doit mener à l'obtention de deux résultats :

- Le regroupement de l'ensemble des structures de données.
- La vérification que l'analyse couvre effectivement tous les cas d'utilisation métier.

Ces résultats prennent la forme de diagrammes de classe et d'une matrice de validation. On veut en particulier faire apparaître les classes suivantes (Obrenović & Starčević, 2006) :

- Les classes limites, décrivant tous les interacteurs du système, utilisateurs ou autres systèmes.
- Les classes de contrôle, qui décrivent certains traitements.
- Les classes d'entité, décrivant les données persistantes du système.

Nous signalons que nous apporterons ultérieurement dans ce document des précisions quant à l'analyse des différentes modalités intégrées à notre interface multimodale (voir le chapitre **L'intégration métier verticale : support de la multimodalité**). En effet, l'utilisation de cette interface, et a fortiori l'analyse qui peut en être faite, sort du périmètre de l'application générique que nous décrivons dans le présent chapitre.

Au terme de cette activité, on remarque que l'analyse, au sens d'UP, est couverte.

1) L'élaboration du diagramme de classe récapitulatif

FICHE GUIDE – FG13	
Activité 4 : Synthèse de l'analyse	
Sous-activité 4.1 :	
Elaboration du diagramme de classe récapitulatif	
Objectif	Regrouper l'ensemble des classes dans un seul diagramme pour avoir une vision globale du système étudié
Point de départ	Tous les diagrammes de classe des cas d'utilisation étudiés
Point d'arrivée	Regroupement des classes en un seul DCL
Démarche d'élaboration	
<p>1 - Récupérer l'ensemble des DCL de tous les CU.</p> <p>2 - Mettre en commun, pour les mêmes classes, les attributs et opérations de chaque classe définis dans les DCL par CU. Chaque classe doit être décrite de manière exhaustive.</p> <p>3 - Mettre en commun toutes les associations entre les classes définies dans les DCL par CU.</p> <p>4 - Déterminer les relations entre les différents DCL des CU. Mettre en place les nouvelles relations nécessaires dans le DCL récapitulatif.</p>	

Tableau 20 : Fiche Guide 13 pour UP7 (d'après (Gabay & Gabay, 2008))

Dans cette sous-activité, nous avons rassemblé l'ensemble des diagrammes de classe que nous avons précédemment réalisés. La **Figure 128** présente ainsi l'intégralité des classes nécessaires à tous les cas d'utilisations identifiés (voir la sous-section **La description des cas d'utilisation**). Ces classes sont autant d'objets et d'agents indispensables à l'accomplissement de la prise en charge de la dyslexie à l'aide de notre système. Parmi ces classes, une partie est exclusivement vouée à être instanciée en mémoire persistante, qui est destinée à la gestion de données pérennes, alors que le reste se limitera à une instanciation en mémoire volatile, qui est seulement destinée à la manipulation des données pendant l'exécution du programme.

Nous noterons que des diagrammes, moins fonctionnels et beaucoup plus techniques, seront présentés plus loin dans ce chapitre (voir la sous-section **L'élaboration des diagrammes de classe techniques**). Ces diagrammes nous permettront la description précise des cas d'utilisations dans le contexte de notre framework de programmation (voir la sous-section **La perspective du méta-système**), qui est basé sur le paradigme MVC (voir la sous-section **Le design pattern Modèle-Vue-Contrôleur**) et le concept d'agents (voir la sous-section **Les systèmes multi-agents**).

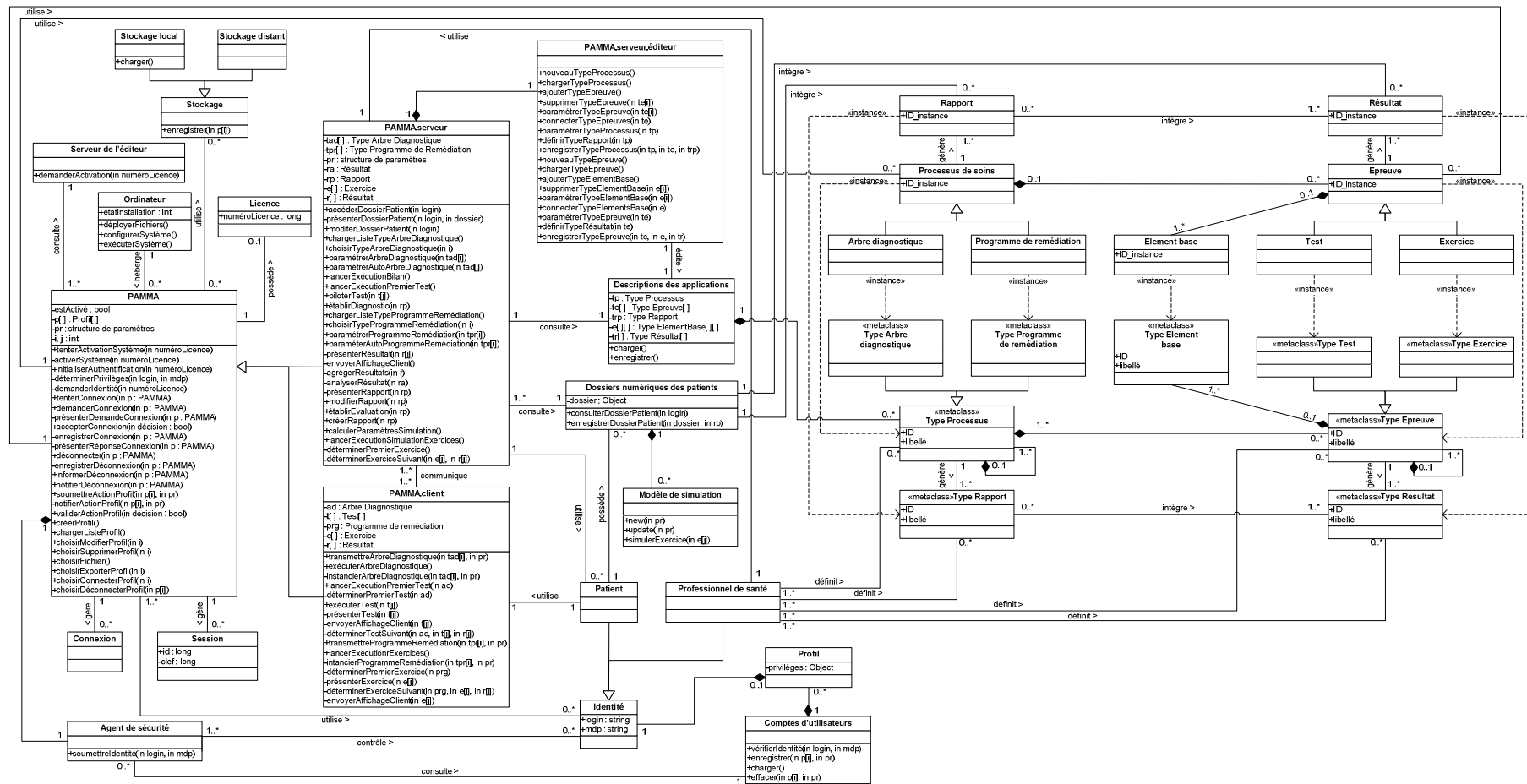


Figure 128 : Diagramme de classe récapitulatif de l'ensemble du système

2) L'élaboration de la matrice de validation

FICHE GUIDE – FG14	
Activité 4 : Synthèse de l'analyse	
Sous-activité 4.2 : Elaboration de la matrice de validation	
Objectif	Vérifier la complétude de l'analyse du système et établir la traçabilité entre les CU métiers (activité exigences fonctionnelles) et les CU d'analyse (activité analyse des CU)
Point de départ	Les CU métiers et d'analyse
Point d'arrivée	Complétude de l'analyse du système vérifiée et traçabilité entre les CU effectuée
Démarche d'élaboration	
<p>1 - Reprendre les CU métiers et d'analyse.</p> <p>2 - Établir une correspondance entre les CU métiers et les CU d'analyse <i>via</i> la matrice de validation.</p> <p>(NB : des cas d'utilisation peuvent être présents uniquement dans l'activité d'analyse des CU car ils répondent à une problématique spécifique liée par exemple à la gestion des utilisateurs par un administrateur)</p> <p>3 - Conclure sur la complétude de l'activité d'analyse des CU en s'assurant que tous les CU métier ont été repris dans l'activité d'analyse des CU.</p>	

Tableau 21 : Fiche Guide 14 pour UP7 (d'après (Gabay & Gabay, 2008))

Afin de vérifier que notre analyse de cas est complète et englobe tous les cas d'utilisation métier, nous avons réalisé une matrice de validation (**Tableau 22**). Cette dernière nous permet également de mettre en correspondance les cas d'utilisation métier, identifiés pendant l'analyse fonctionnelle (voir la sous-section **L'élaboration du diagramme des cas d'utilisation système**), et ceux spécifiés durant l'analyse des cas d'utilisation (voir la sous-section **La description des cas d'utilisation**).

A l'issue de cette sous-activité, nous concluons, à l'aide de cette matrice, que notre analyse des cas d'utilisation est complète.

Cas d'utilisation métier	Cas d'utilisation analyse
	Installation et activation du système (Figure 65)
« Définir des processus de soins » (Figure 50)	Cas 1 - « Définir des processus de soins » (Figure 68)
« Définir des épreuves » (Figure 51)	Cas 2 - « Définir des épreuves » (Figure 69)
	Accéder à un dossier patient (Figure 67)
« Réaliser un bilan » (Figure 52)	Cas 3 - « Réaliser un bilan » (Figure 70)
« Passer les tests » (Figure 53)	Cas 4 - « Passer les tests » (Figure 71)
« Analyse et conclusion » (Figure 54)	Cas 5 - « Analyse et conclusion » (Figure 72)
« Calibrage de la simulation » (Figure 55)	Cas 6 - « Calibrage de la simulation » (Figure 73)
« Planifier le programme de remédiation » (Figure 56)	Cas 7 - « Planifier le programme de remédiation » (Figure 74)
« Suivre le programme » (Figure 57)	Cas 8 - « Suivre le programme » (Figure 75)
« Simuler le suivi du programme » (Figure 58)	Cas 9 - « Simuler le suivi du programme » (Figure 76)
« Analyse et ajustement » (Figure 59)	Cas 10 - « Analyser les résultats » (Figure 77) Cas 11 - « Ajuster la remédiation » (Figure 78)
« Recalibrage de la simulation » (Figure 60)	Cas 12 - « Recalibrage de la simulation » (Figure 79)
	Cas 13 - « Création d'un profil » (Figure 81)
	Authentification de l'utilisateur (Figure 66)
	Cas 14 - « Modification d'un profil » (Figure 82)
	Cas 15 - « Suppression d'un profil » (Figure 83)
	Cas 16 - « Import/Export d'un profil » (Figure 84)
	Cas 17 - « Connexion/Déconnexion d'un profil » (Figure 85)
	Cas 18 - « Consultation d'un dossier patient » (Figure 86)
	Cas 19 - « Ouverture/Fermeture du serveur aux clients » (Figure 87)
	Cas 20 - « Connexion/Déconnexion d'un serveur » (Figure 88)

Tableau 22 : Matrice de validation des cas d'utilisation

E) La conception

La cinquième et dernière activité que nous effectuerons se concentre sur la manière de réaliser techniquement le système informatique. Elle veut décrire comment le système sera construit lors de son implémentation. Nous verrons plus loin dans ce manuscrit comment une partie de ce travail de modélisation peut être utilisée dans l'implémentation systématique de SMA (voir la sous-sous-section **Le niveau 2 de l'éditeur**). En effet, elle a pour but de définir et de mettre en place les choix d'architecture technique, et conduit à quatre résultats :

- Les choix techniques retenus.
- Les scénarii techniques pour chaque cas d'utilisation.
- Les structures de données techniques par cas d'utilisation.
- Le paquetage de l'ensemble de ces structures.

La modélisation ainsi achevée se composera de diagrammes de séquence, de classe et de paquetage. A l'aide de cette dernière activité, nous souhaitons étendre les résultats de l'analyse en y intégrant pragmatiquement des considérations techniques. La conception préliminaire est normalement couverte à la fin de cette activité, et il est alors possible de passer à l'implémentation du système informatique.

Nous signalons que nous n'approfondirons pas le fonctionnement de l'éditeur dans cette section. Effectivement, ce composant du système est lié au langage de modélisation retenu pour la spécification des applications destinées à fonctionner sur la PAMMA et nous reviendrons sur ce sujet plus loin dans ce manuscrit (voir la section **L'intégration des processus métiers**)).

1) La réalisation des choix techniques

FICHE GUIDE – FG15	
Activité 5 : Conception	
Sous-activité 5.1 : Réalisation des choix techniques	
Objectif	Choisir l'architecture technique cible et les technologies associées
Point de départ	Étapes d'expression des besoins et d'analyse terminées (Activités 1, 2, 3, 4)
Point d'arrivée	Architecture technique et technologies associées choisis
Démarche d'élaboration	
<p>1 - Choix de l'architecture technique en fonction des contraintes techniques imposées lors de l'expression des besoins (ex. contraintes de performances), du contexte client (environnement technique cible) et de l'état de l'art des technologies.</p> <p>2 - Choix des technologies utilisées en fonction de l'architecture sélectionnée et des contraintes techniques.</p>	

Tableau 23 : Fiche Guide 15 pour UP7 (d'après (Gabay & Gabay, 2008))

Comme nous l'avons décrit dans notre partie précédente (voir la section **L'architecture du système**), nous souhaitons utiliser un framework de programmation, basé sur le paradigme MVC et le concept d'agents (**Figure 22**), pour structurer l'architecture de notre système. Ainsi, le système peut être décomposé selon trois dimensions :

- Une dimension pour mesurer la portée fonctionnelle selon trois types de couches : les couches du domaine métier, les couches de la section métier et les couches applicatives.
- Une dimension qui renvoie à la technicité et qui est constituée de trois couches : la couche de la Vue, la couche du Modèle et la couche du Contrôleur.
- Une dimension qui répartit en trois couches le niveau d'abstraction conceptuelle de la programmation mise en œuvre : la couche binaire, la couche orientée objet et la couche orientée agent.

Ce découpage en couches selon trois dimensions permet la définition de vingt-sept couches distinctes. Elles nous permettront d'organiser les classes et, à ce titre, elles seront la base de notre regroupement en paquetages (voir la sous-section **L'élaboration du diagramme de paquetage**). Pour l'instant, recentrons-nous sur l'architecture de notre dispositif sans prendre en compte ce framework de programmation. La **Figure 129** offre une vue d'ensemble de la partie logicielle de notre système. Nous y avons fait apparaître les deux plates-formes, le client et le serveur, qui disposent chacune d'une base logicielle commune que nous avons nommée le « noyau de la PAMMA ». Nous avons aussi identifié quelles parties du système, étant couramment sollicitées, peuvent être externalisées sous la forme de services/modules de gestion de données persistantes :

- Les comptes d'utilisateurs
- Les dossiers numériques des patients

- Les descriptions des applications

Ces services peuvent, par exemple, être assurés par des bases de données et ne sont pas nécessairement à l'usage exclusif de notre dispositif. En effet, dans le cas des comptes d'utilisateurs et des dossiers numériques des patients, il paraît assez pertinent de vouloir les mettre en commun avec d'autres éventuelles applications d'un SI préexistant. Nous nous contenterons donc d'y accéder au travers d'agents d'interface. On peut également signaler que certains de ces services (les comptes d'utilisateurs et les descriptions des applications) sont nécessaires au fonctionnement de la PAMMA et sont donc exploitables directement par le noyau, alors que d'autres sont spécifiques à des applications développées sur la PAMMA, comme, par exemple, dans le cas de notre application de prise en charge de la dyslexie, et interagissent uniquement avec ces applications.

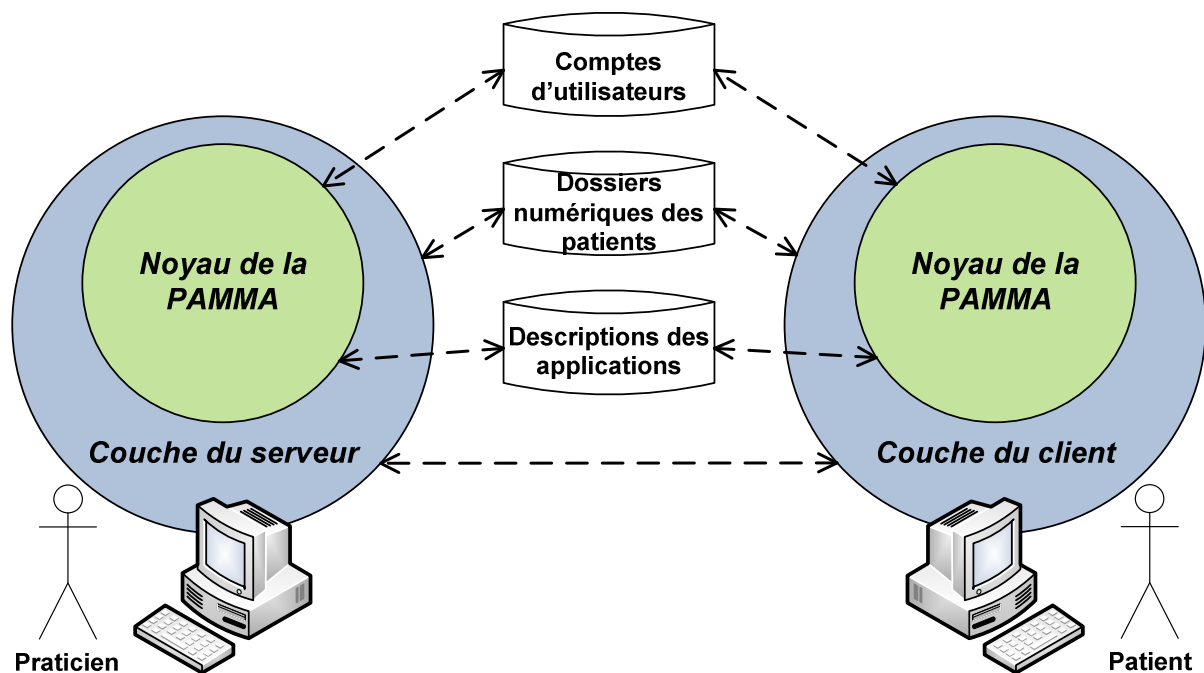


Figure 129 : Architecture globale du dispositif

Dans le reste de cette section, nous expliciterons, en particulier, le fonctionnement des éléments techniques du noyau de la PAMMA suivants :

- Un service de pages blanches
- Un service de pages jaunes
- Un médiateur/facilitateur

Finalement, on notera que nous ne précisons pas, pour l'instant, l'emploi pour l'implémentation de la PAMMA d'autres technologies, ni langages de programmation ni plates-formes spécifiques, afin de garder le maximum de souplesse pour une future implémentation.

2) L'élaboration des diagrammes de séquence technique

FICHE GUIDE – FG16	
Activité 5 : Conception	
Sous-activité 5.2 : Elaboration des diagrammes de séquence technique	
Objectif	Représenter les interactions entre les acteurs et tous les objets du système (incluant les objets techniques) pour tous les cas d'utilisation en considérant les différents scénarii associés. Représenter les objets en utilisant le paradigme MVC (Burbeck, 1992).
Point de départ	Architecture choisie, DSE et DCL de l'activité 3
Point d'arrivée	Interactions entre les acteurs et tous les objets du système (incluant les objets techniques) définies pour tous les cas d'utilisation
Démarche d'élaboration	
Pour chaque cas d'utilisation (CU) et chaque scénario identifié :	
<p>1 - Reporter le ou les acteurs du DCU impliqués dans le scénario.</p> <p>2 - Représenter les objets du système impliqués dans le scénario en prenant comme base ceux identifiés lors du DSE de l'activité 3 :</p> <ul style="list-style-type: none"> - Objets « Vue » - Objets « Contrôleur » - Objets « Modèle » - Objets techniques (Collection, Date...) identifiés uniquement lors de cette étape <p>3 - Représenter les messages (synchrones/asynchrones) entre les acteurs et les objets et entre les objets eux-mêmes. La chronologie des échanges doit être respectée.</p> <p>4 - Modéliser les fragments d'interaction si nécessaire.</p>	

Tableau 24 : Fiche Guide 16 pour UP7 (d'après (Gabay & Gabay, 2008))

Pendant cette sous-activité, nous allons refaire nos précédents diagrammes de séquence (voir la sous-section **L'élaboration des diagrammes de séquence**) pour y faire apparaître l'influence de notre framework de programmation (voir la sous-section **La perspective du méta-système**). A ce titre, nous préciserons quels éléments appartiennent à la Vue, au Contrôleur ou au Modèle et nous présenterons les agents du système. D'un point de vue MA, les étapes essentielles sont l'identification des agents intervenant dans le domaine d'application ainsi que la description de leurs interactions (Bernon et al., 2009). Cependant nous ne représenterons pas réellement les messages entre agents, pour lesquels il faudrait choisir un langage précis de communication entre agents, qui est défini par un ensemble de performatives/primitives et le choix d'un langage déclaratif, mais nous préciserons juste la sémantique de l'échange, au travers de fonctions disposant de paramètres.

Ainsi, nous introduirons les nouveaux stéréotypes suivants :

- <<Vue>>
- <<Contrôleur>>
- <<Modèle>>
- <<Agent>>

Nous signalons également que nous allons présenter nos DSE de manière à ce qu'ils reflètent la navigation proposée par nos différentes interfaces (voir la sous-section **L'élaboration des interfaces utilisateur**). Nous présenterons aussi le fonctionnement d'éléments du noyau de la PAMMA que nous n'avons qu'évoqués jusque là. D'autre part, on notera que nous ne représenterons que les agents de plus haut niveau des holarchies et des hiérarchies d'agents d'IHM et de gestion des données (voir les sous-sous-sections **Modéliser les interactions avec les utilisateurs** et **Modéliser la gestion des données**).

Notre premier diagramme reprend la précondition aux cas d'utilisation « Installation et activation du système » (**Figure 130**). On remarque que seule la modélisation des éléments appartenant à la PAMMA, c'est-à-dire les éléments prototypés <<Système>>, bénéficie de l'utilisation de notre framework MA.

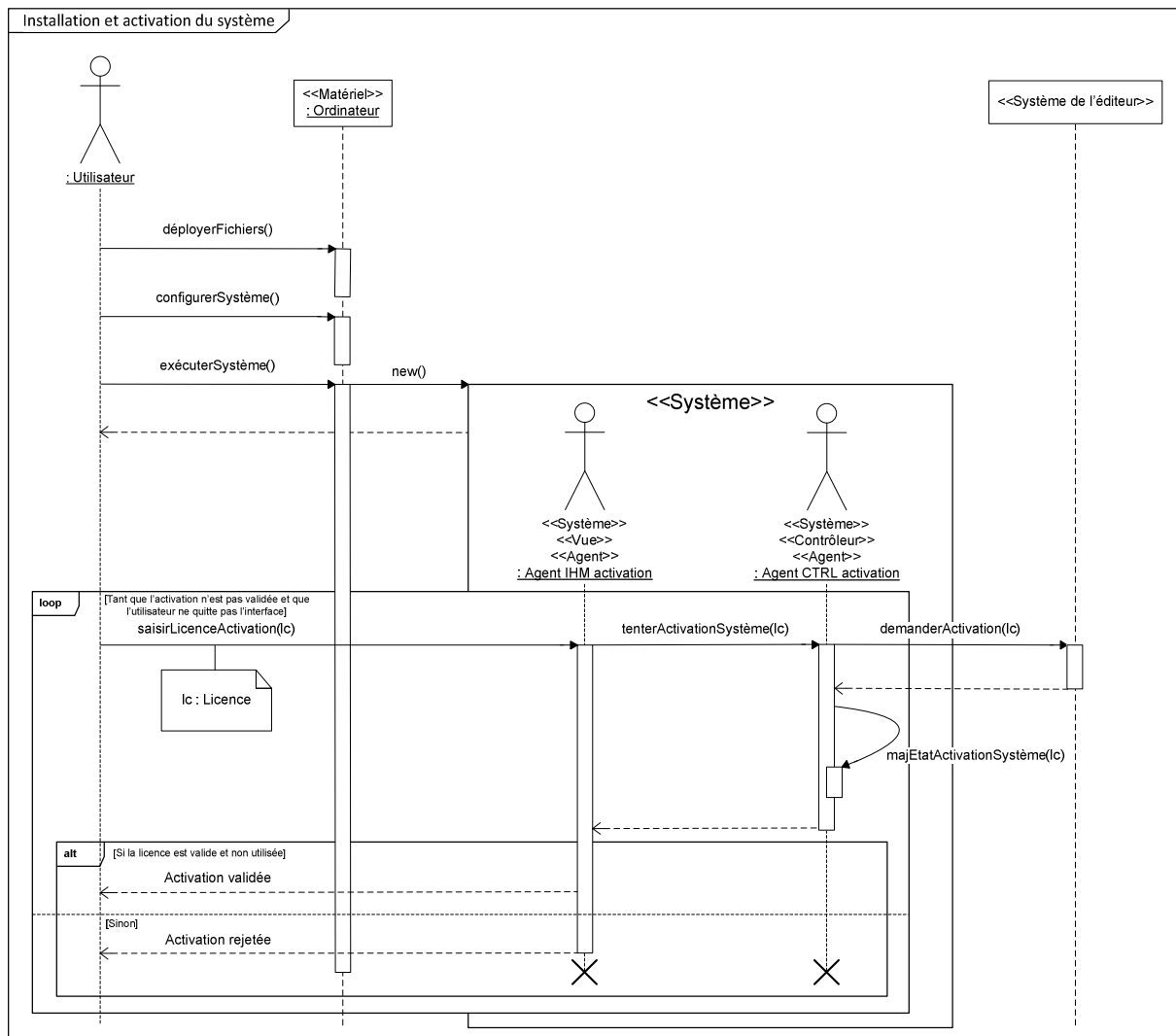


Figure 130 : Diagramme de séquence technique de la précondition aux cas d'utilisation « Installation et activation du système »

Les diagrammes des deux autres préconditions, que sont l'authentification de l'utilisateur (**Figure 131**) et l'accès à un dossier patient (**Figure 132**), sont peu modifiés par la mise en œuvre de notre framework. On notera, néanmoins, que les services de gestion de données qu'ils sollicitent sont assurés par un agent qui masque au reste de système l'implémentation technologique retenue pour le stockage de ces données.

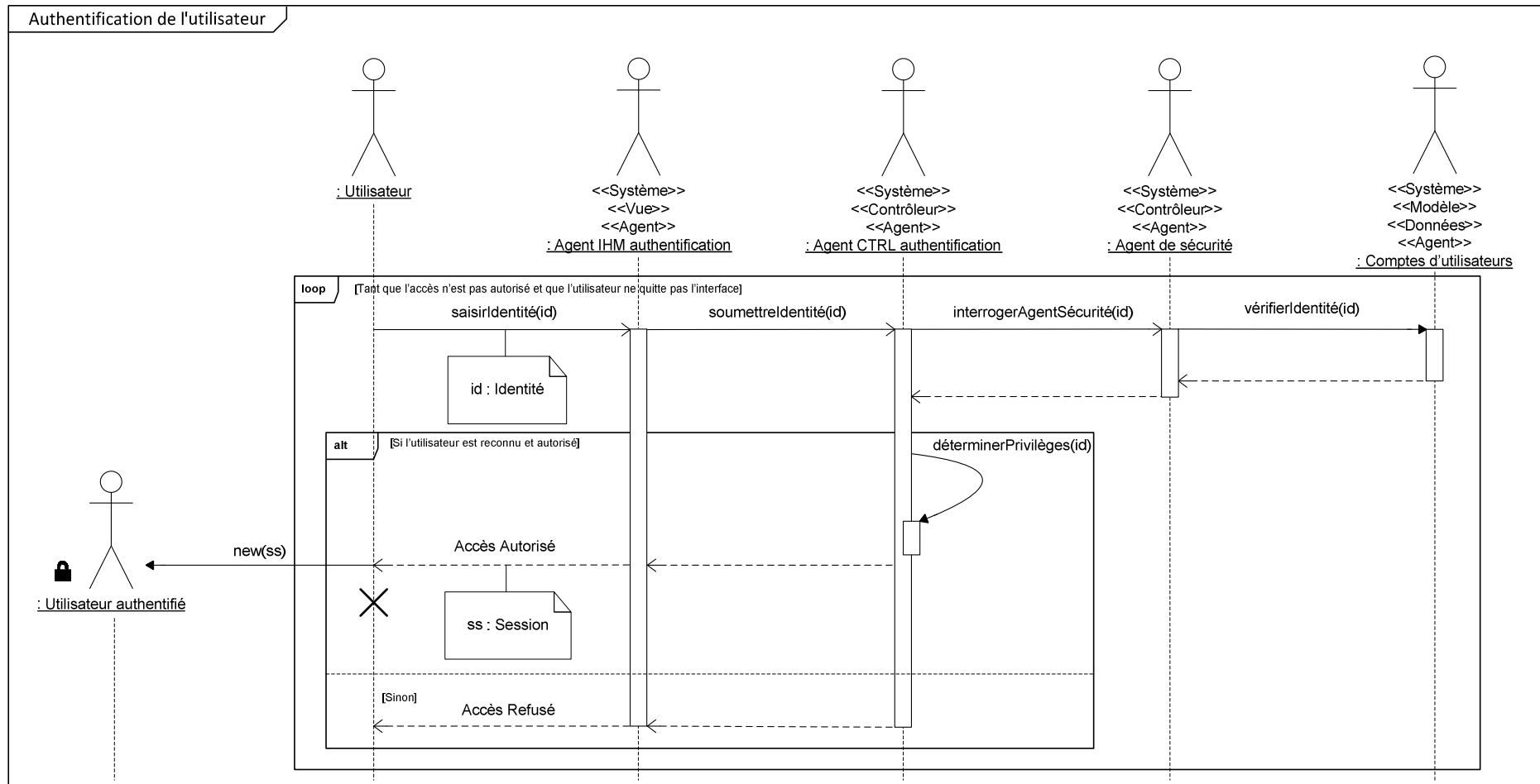


Figure 131 : Diagramme de séquence technique de la précondition aux cas d'utilisation « Authentification de l'utilisateur »

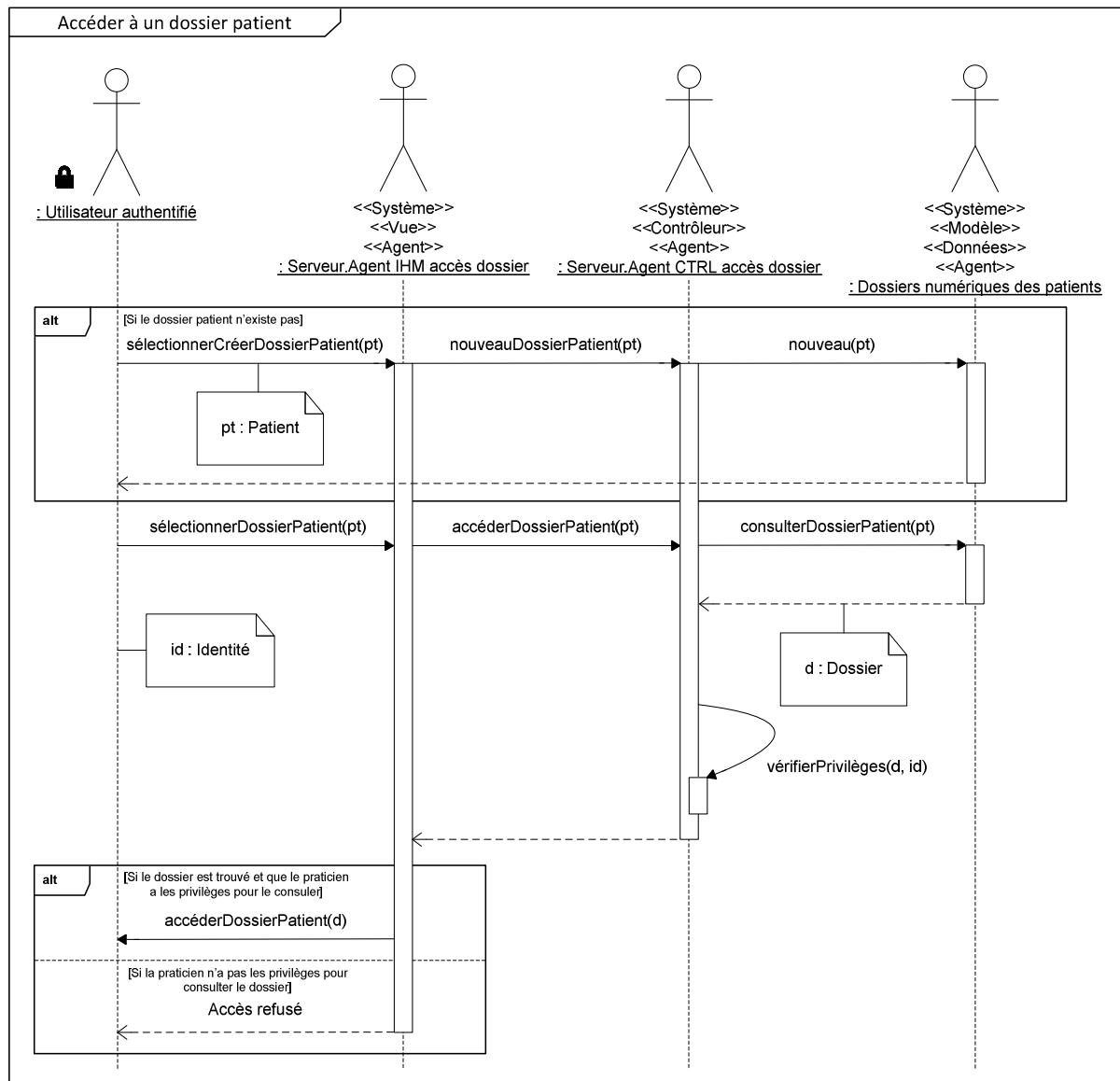


Figure 132 : Diagramme de séquence technique de la précondition aux cas d'utilisation « Accéder à un dossier patient »

Nous allons maintenant présenter la gestion des profils auxquels l'utilisateur est confronté dès qu'il lance le système (voir la sous-section **L'élaboration des interfaces utilisateur**). Cette dernière couvre la précondition aux cas d'utilisation « Valider une action sur un profil » (**Figure 133**) et les cas d'utilisation de 13 à 17 (**Figure 134**). Nous avons ainsi regroupé ces cas d'utilisation pour donner une vue d'ensemble des fonctionnalités offertes par l'écran de gestion des profils (**Figure 97**). Nous précisons que, dans le cas de la suppression d'un profil, nous supposons que l'utilisateur a confirmé son choix sur la fenêtre de confirmation (**Figure 101 (a)**). D'autre part, pour ne pas trop surcharger le diagramme, nous n'avons pas fait apparaître les changements de rôle de « Utilisateur » à « Utilisateur authentifié » après une authentification réussie (**Figure 43**).

Pour coller au schéma de navigation lors de l'utilisation du système, nous avons choisi d'explicitier l'utilisation de l'accueil, aussi bien du serveur (**Figure 135**) que du client (**Figure 136**). A la fin de cette sous-activité, nous aurons ainsi référencé tous les agents liés à l'IHM.

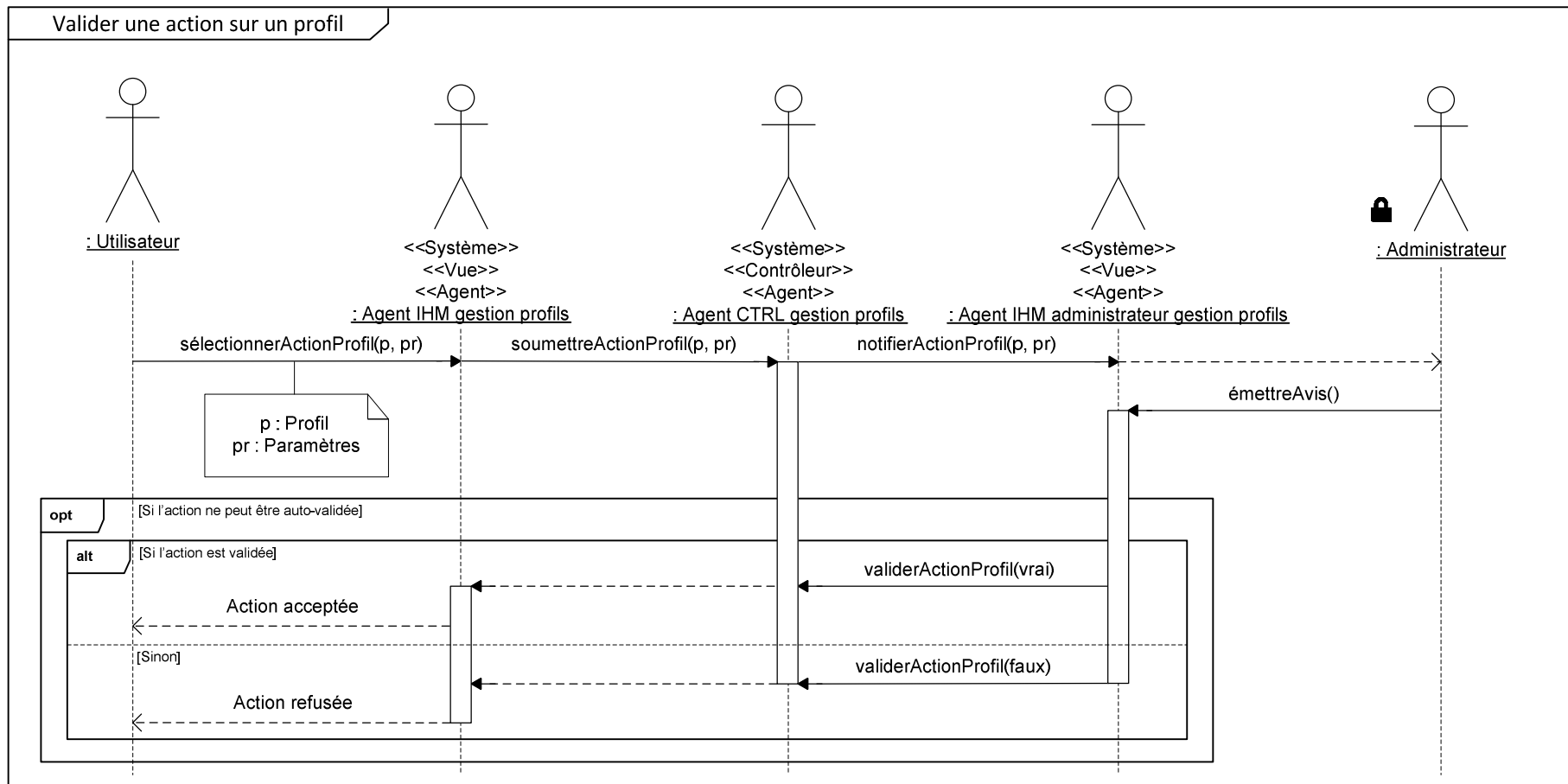


Figure 133 : Diagramme de séquence technique de la précondition aux cas d'utilisation « Valider une action sur un profil »

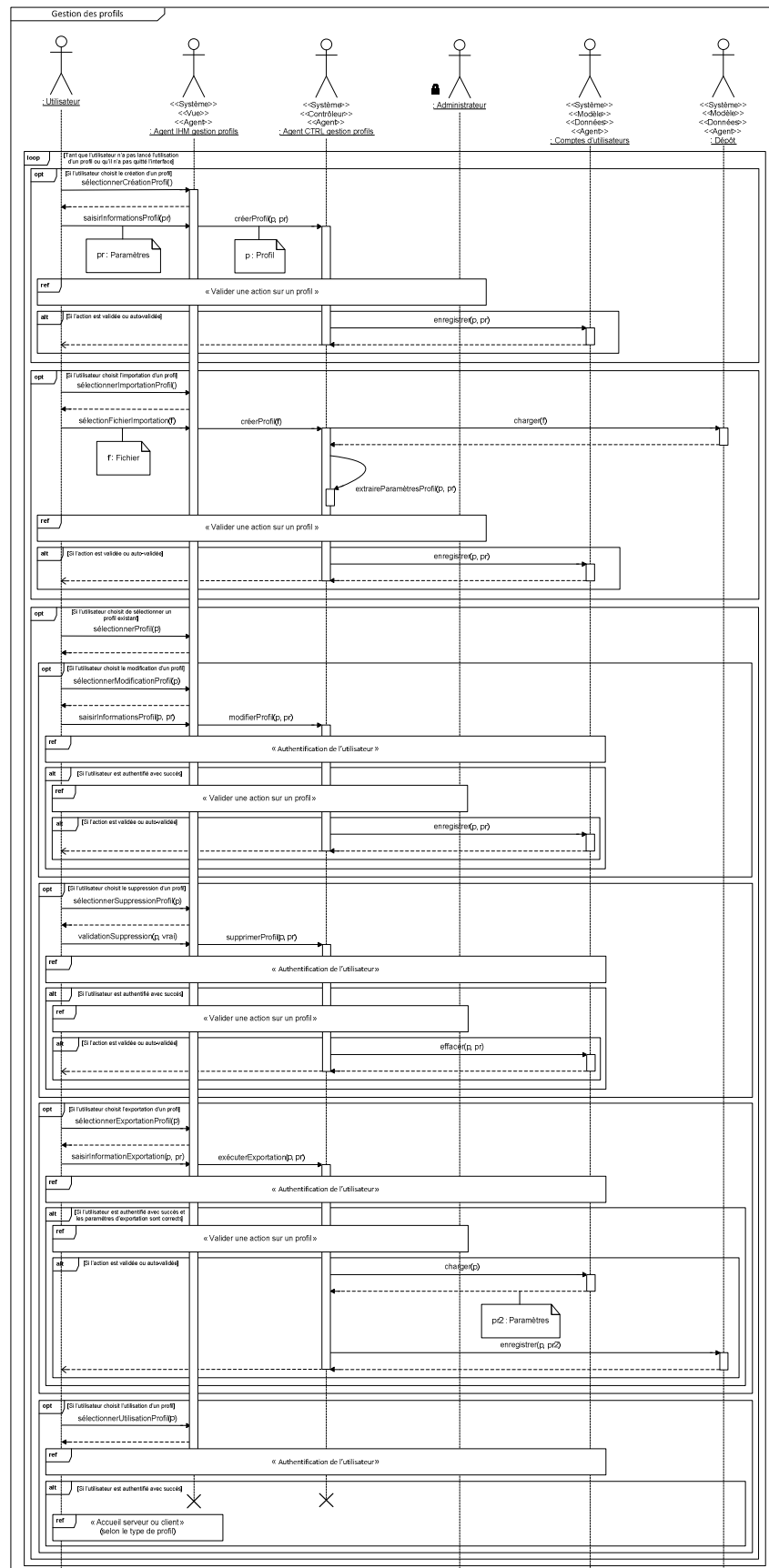


Figure 134 : Diagramme de séquence technique de gestion des profils (cas d'utilisation de 13 « Création d'un profil » à 17 « Connexion/Déconnexion d'un profil »)

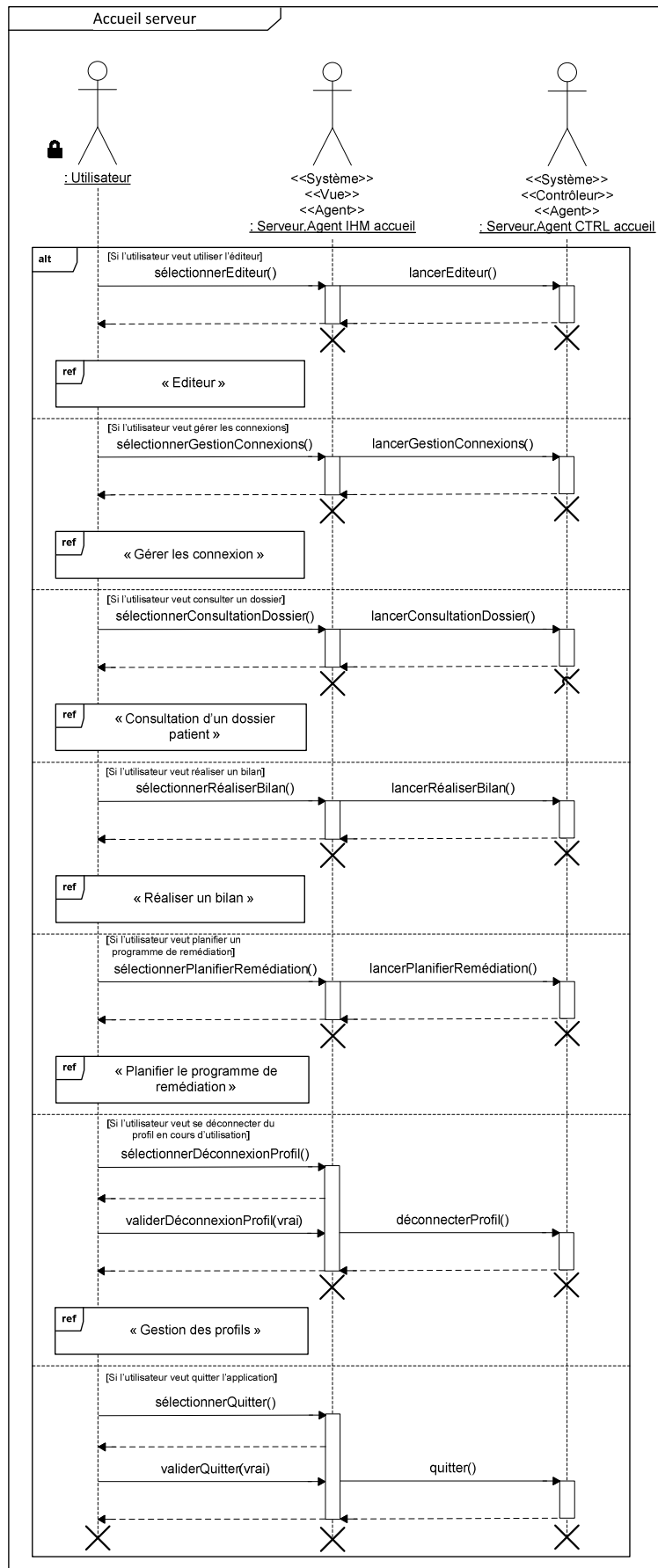


Figure 135 : Diagramme de séquence technique de l'utilisation de l'accueil du serveur

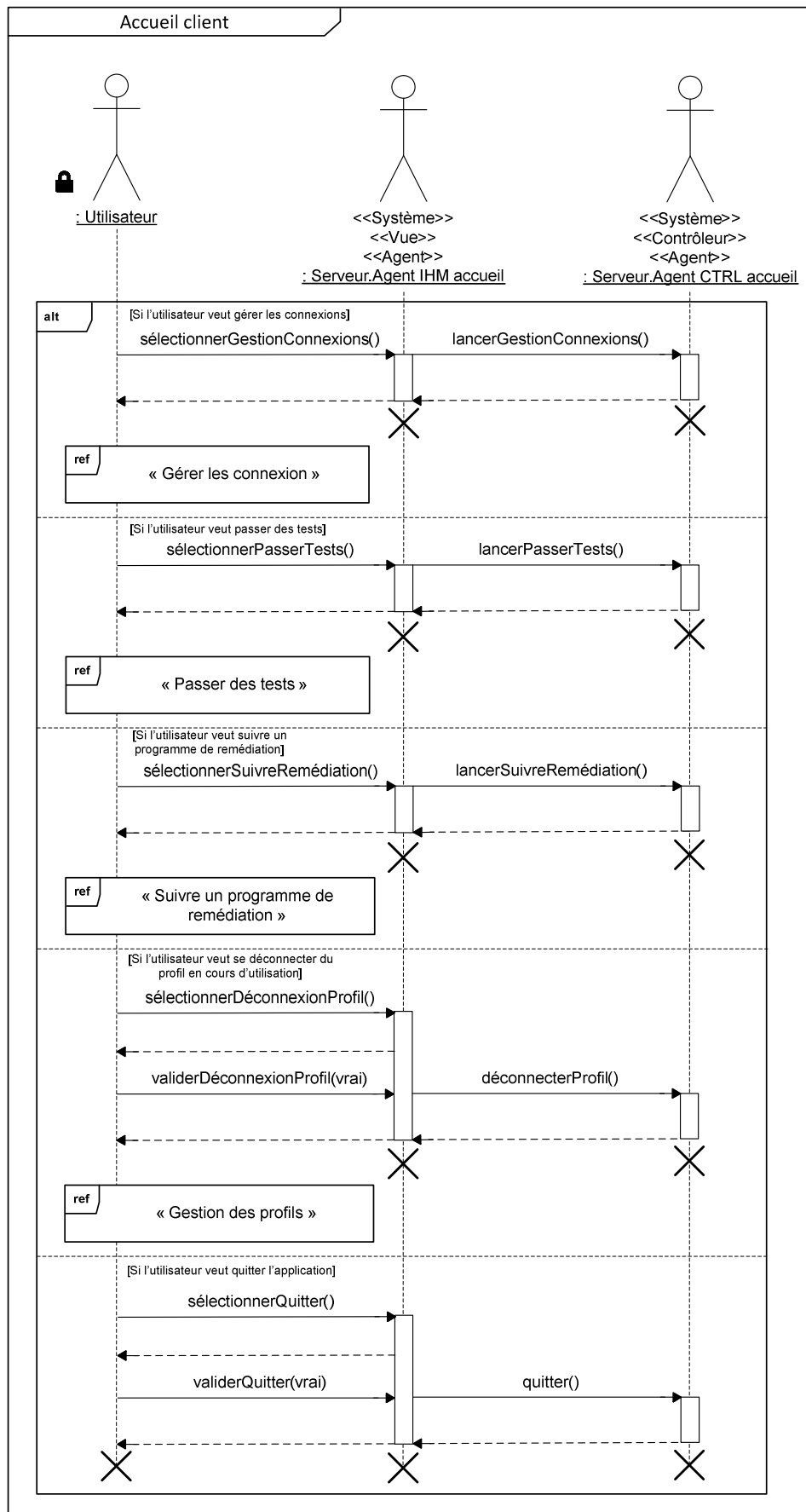


Figure 136 : Diagramme de séquence technique de l'utilisation de l'accueil du client

Le serveur et le client ont accès à la gestion des connexions entre PAMMA. Cette gestion permet la réalisation des cas d'utilisation 19 et 20 (**Figure 137**). D'autre part, au niveau de chaque PAMMA, nous avons choisi de représenter les connexions par des agents dédiés, des agents de connexion. Les échanges entre des PAMMA connectées de la sorte passeront exclusivement par ces agents.

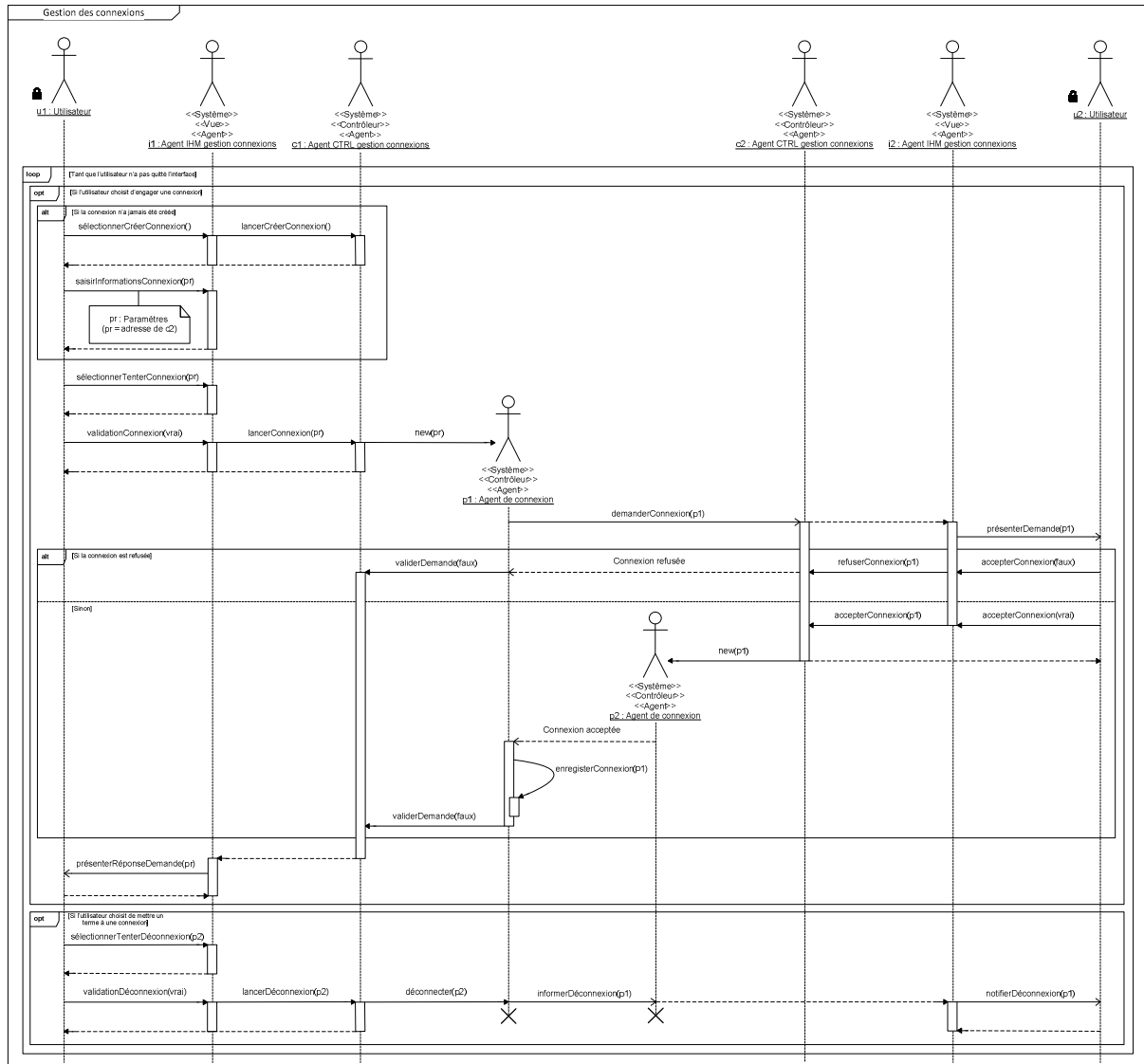


Figure 137 : Diagramme de séquence technique de l'utilisation de l'écran de gestion des connexions (cas d'utilisation 19 « Connexion client-serveur » et 20 « Déconnexion client-serveur »)

Nous avons ensuite représenté l'utilisation de l'écran de consultation d'un dossier patient (**Figure 138**) conformément à l'IHM que nous avons prototypée (**Figure 112**). Cet écran rassemble les fonctionnalités liées à la consultation et la modification des rapports ainsi qu'à l'utilisation des modèles de simulation.

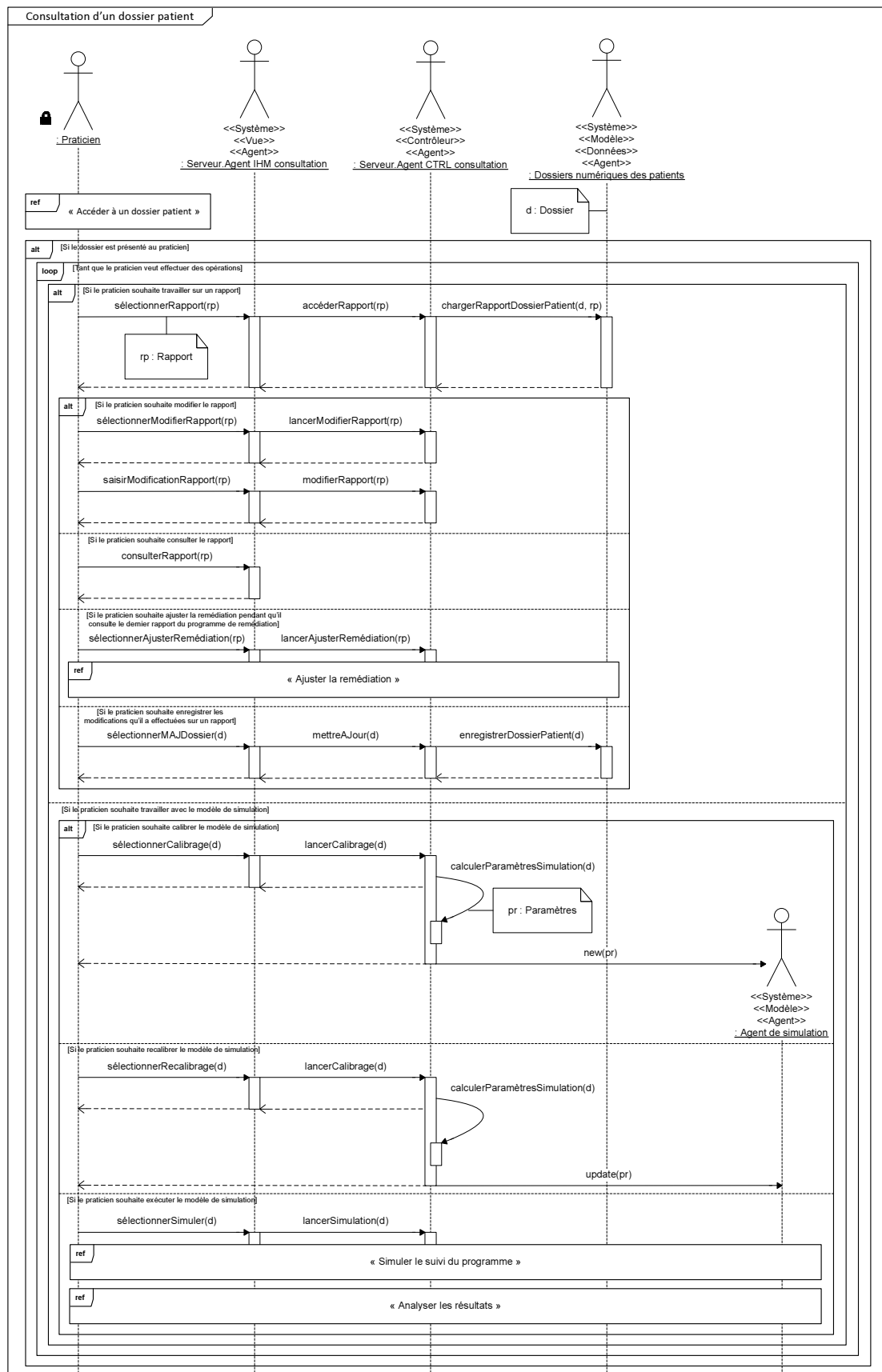


Figure 138 : Diagramme de séquence technique de la consultation d'un dossier patient (cas d'utilisation 6 « Calibrage de la simulation », 12 « Recalibrage de la simulation » et 18 « Consultation d'un dossier patient »)

Nos diagrammes suivants viennent conclure les diagrammes de séquence techniques de notre application générique pour la prise en charge de la dyslexie. A ce titre, nous avons rassemblé, d'un côté, les cas d'utilisation impliqués dans l'aide diagnostique (**Figure 139**) et, d'un autre, ceux liés à la remédiation des patients (**Figure 140**). Sur chacun de ces diagrammes, qui est constitué de plusieurs séquences correspondant chacune à un cas d'utilisation, nous faisons simultanément apparaître les agents du serveur ainsi que ceux du client. Comme nous l'avons écrit un peu plus haut, ils communiquent entre eux via les agents de connexion. D'autre part, nous avons également parfait notre modélisation en clarifiant bien que le serveur gère les méta-classes, alors que le client gère les classes (sauf dans le cas de la simulation où le serveur les gère toutes).

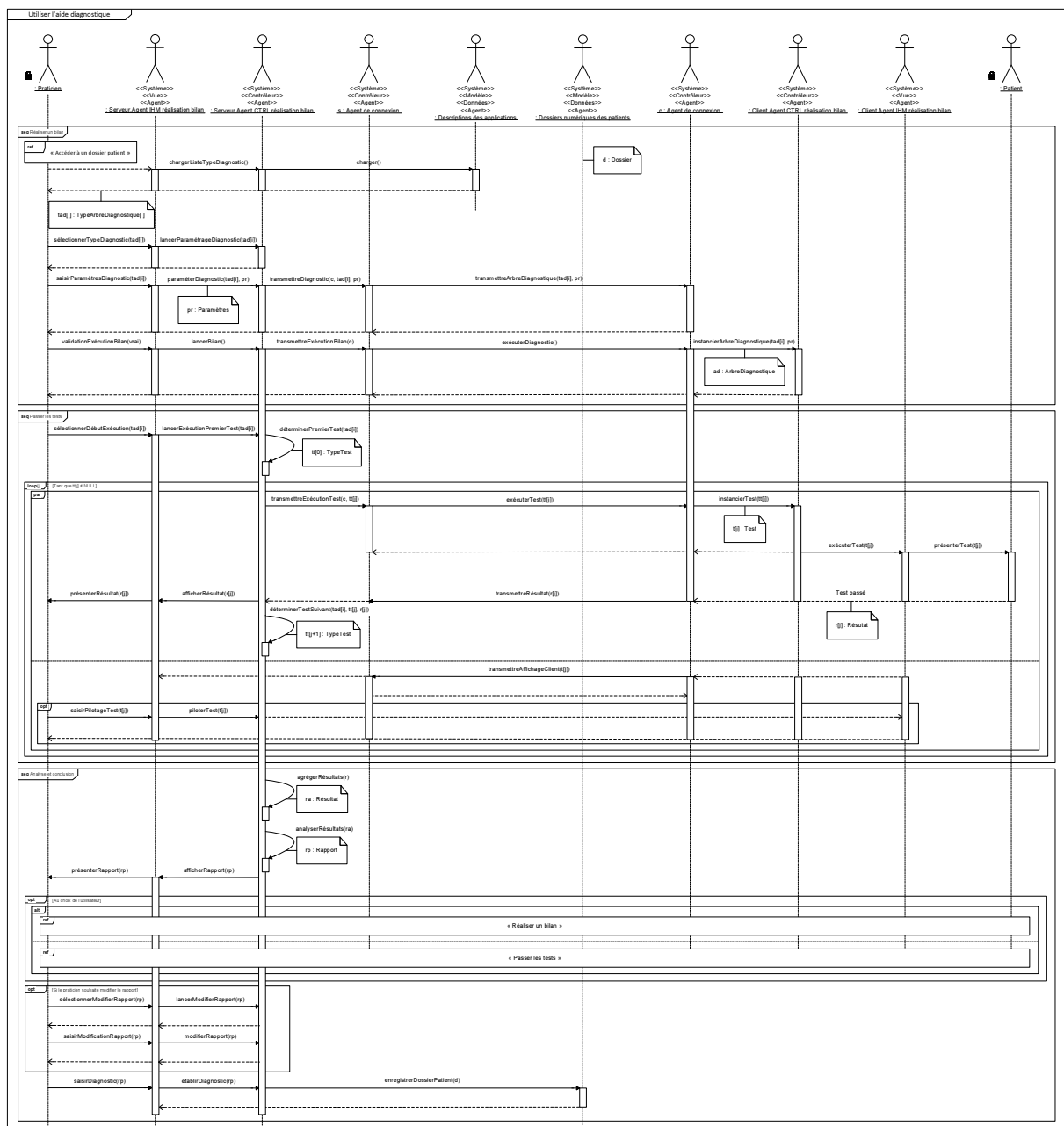


Figure 139 : Diagramme de séquence technique d'utilisation de l'aide diagnostique (cas d'utilisation de 3 « Réaliser un bilan » à 5 « Analyse et conclusion »)

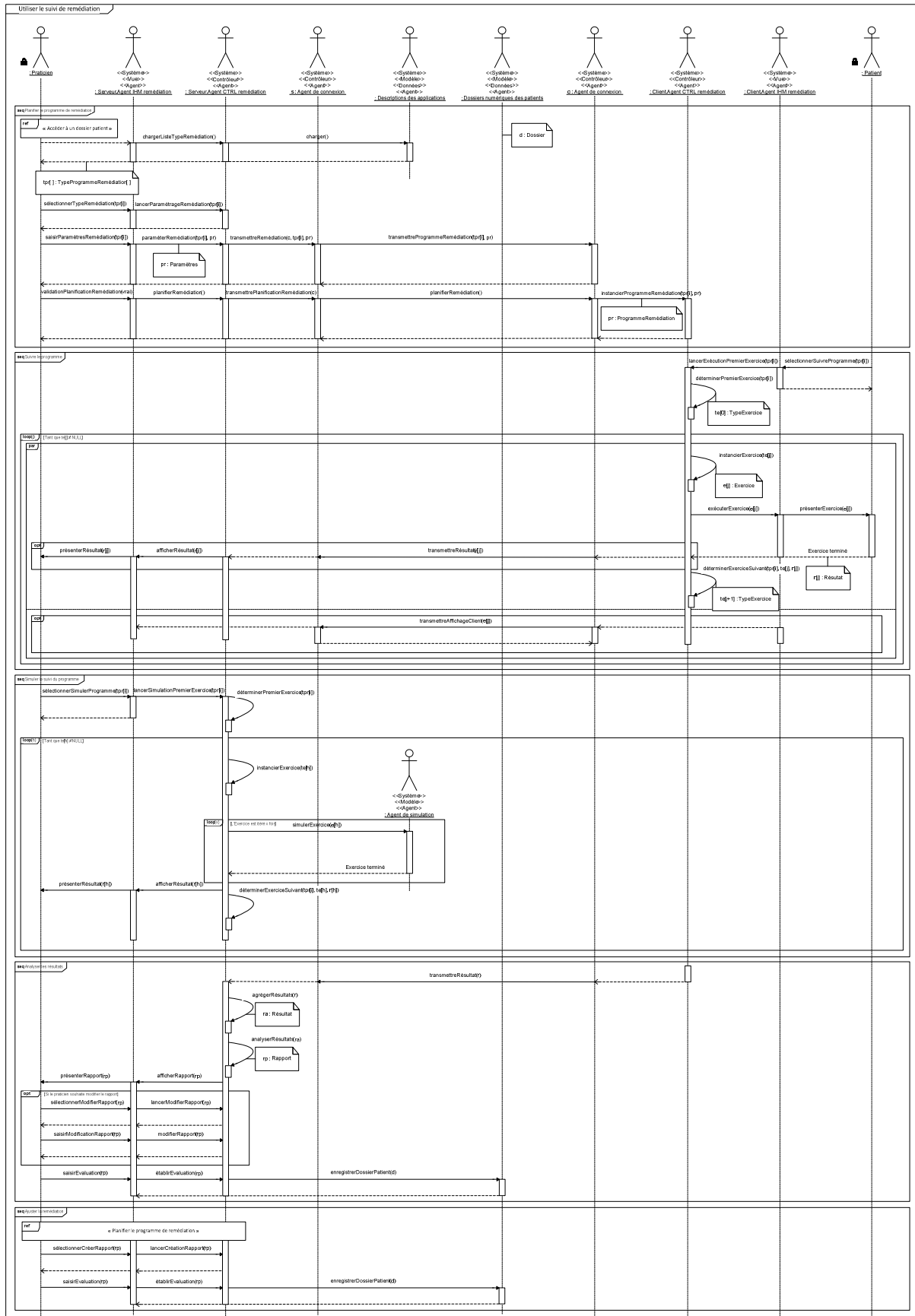


Figure 140 : Diagramme de séquence technique d'utilisation du suivi de remédiation (cas d'utilisation de 7 « Planifier le programme de remédiation » à 11 « Ajuster la remédiation »)

Les derniers diagrammes que nous voulons présenter concernent les services indispensables à la réalisation ultérieure d'application sur la PAMMA. En effet, notre choix technologique d'utiliser notre propre framework (voir la sous-section **La perspective du méta-système**) nous pousse à préciser les grandes lignes du fonctionnement de ces services. Ainsi, nous avons réalisé des diagrammes de séquence techniques pour les services suivants :

- Un service de pages blanches (**Figure 141**)
- Un service de pages jaunes (**Figure 142**)
- Un médiateur/facilitateur (**Figure 143**)

Concernant le service de pages blanches, nous avons modélisé des séquences permettant de déclarer l'inscription et la désinscription d'un agent, de rechercher un agent et de monitorer l'ensemble des agents inscrits (**Figure 141**). Ce service permet de référencer les agents présents sur une PAMMA et d'obtenir leurs adresses respectives. Nous reviendrons ultérieurement sur le système d'adressage (**Figure 169**).

A propos du service de pages jaunes (comme retrouvé, par exemple, dans (Kirn et al., 2003)), il est à noter qu'il est assez similaire au service de pages blanches (**Figure 142**). Cependant, il ne référence que les agents souhaitant mettre à disposition, à l'ensemble des agents de la PAMMA, une de leurs compétences sous la forme d'un service. Par ailleurs, un même agent peut avoir inscrit plusieurs de ses compétences à ces pages jaunes.

Le médiateur/facilitateur sert à permettre la communication entre agents qui ne connaissent pas nécessairement leurs adresses respectives. Ce service propose en réalité deux types de fonctionnalités (Rodríguez-Domínguez et al., 2012) :

- Celles de routage, principalement utilisées pour la transmission de messages synchrones pour une communication « un-à-un ». Dans ce cas, le pattern utilisé est « Request-Reply ».
- Celles de brokage, qui permet la transmission de messages asynchrones dans un contexte d'interaction « un-à-plusieurs » et qui est fréquemment utilisé pour l'envoi de notifications. Ici, il s'agit du pattern « Publish-Subscribe » qui permet de délivrer un message à tous les agents abonnés à un même topique.

Notre diagramme de séquence technique du médiateur/facilitateur (**Figure 143**) est ainsi composé de séquences décrivant comment router un message, s'abonner ou se désabonner à un topique du broker et comment y publier. De plus, le médiateur/facilitateur peut utiliser d'autres technologies en interne (par exemple, CORBA (Object Management Group, 2014d)) et, à ce titre, bénéficier d'une décomposition holarchique d'agents spécifiques à chacune d'elles. On signalera également que nous avons effectué une recherche de l'adresse de l'agent à l'aide de son nom dans les pages blanches, mais qu'il serait possible, à la place, de le rechercher à partir d'une compétence dans les pages jaunes.

Pour conclure, bien qu'on ne l'ait pas représenté ici, tous ces services peuvent faire partie d'un réseau de PAMMA et donc communiquer avec leurs homologues d'une autre PAMMA. A ce titre, s'ils ne peuvent satisfaire une requête, ils peuvent éventuellement la propager à ces homologues pour tenter d'obtenir une réponse favorable à l'agent émetteur de la requête. On signalera également que, pour des raisons d'efficacité, les envois de messages pour le monitoring des agents et services, effectués par les pages blanches et pages jaunes, peuvent être réalisés à l'aide des fonctionnalités de broker d'un médiateur/facilitateur.

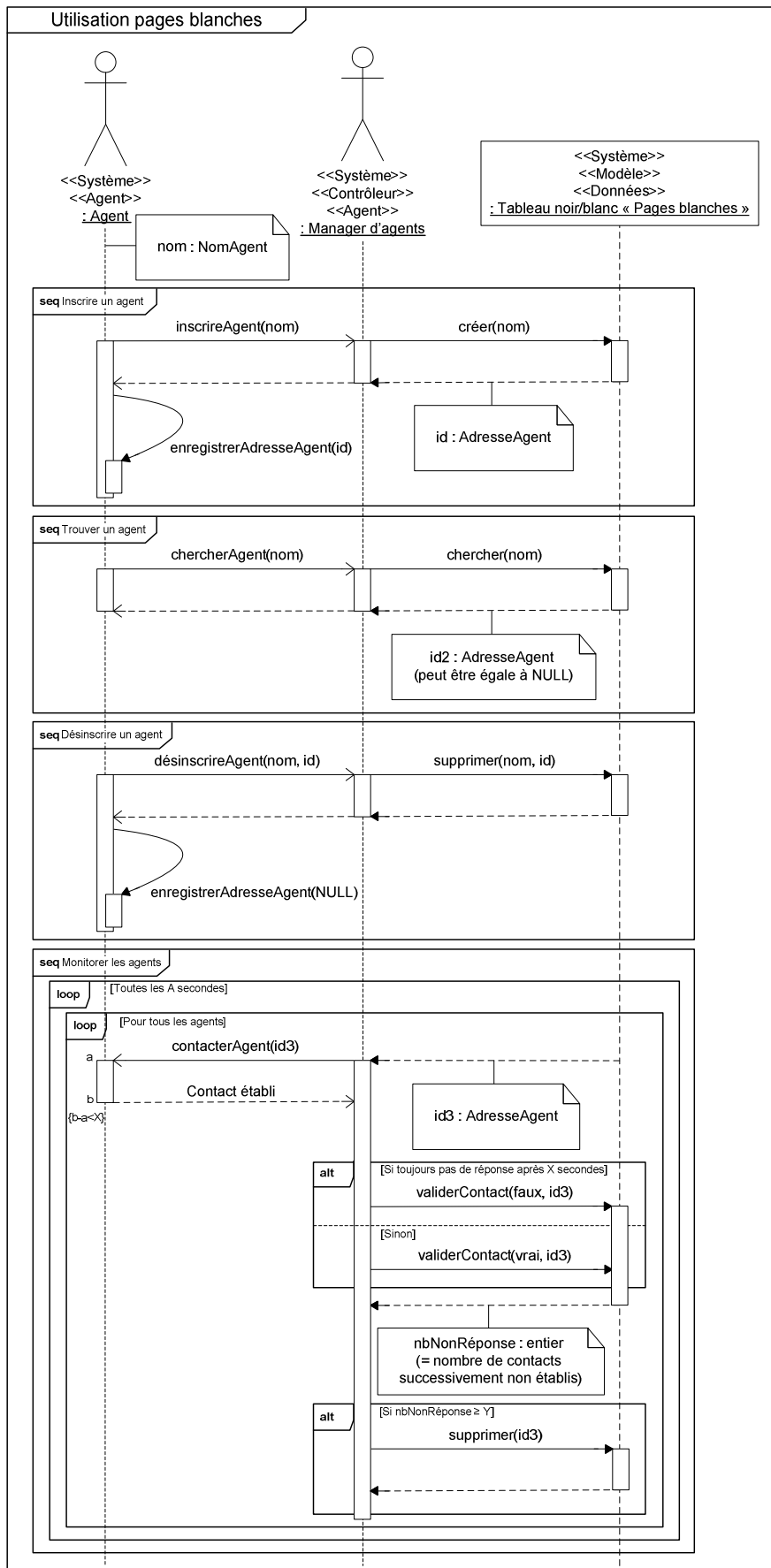


Figure 141: Diagramme de séquence technique d'un service de pages blanches

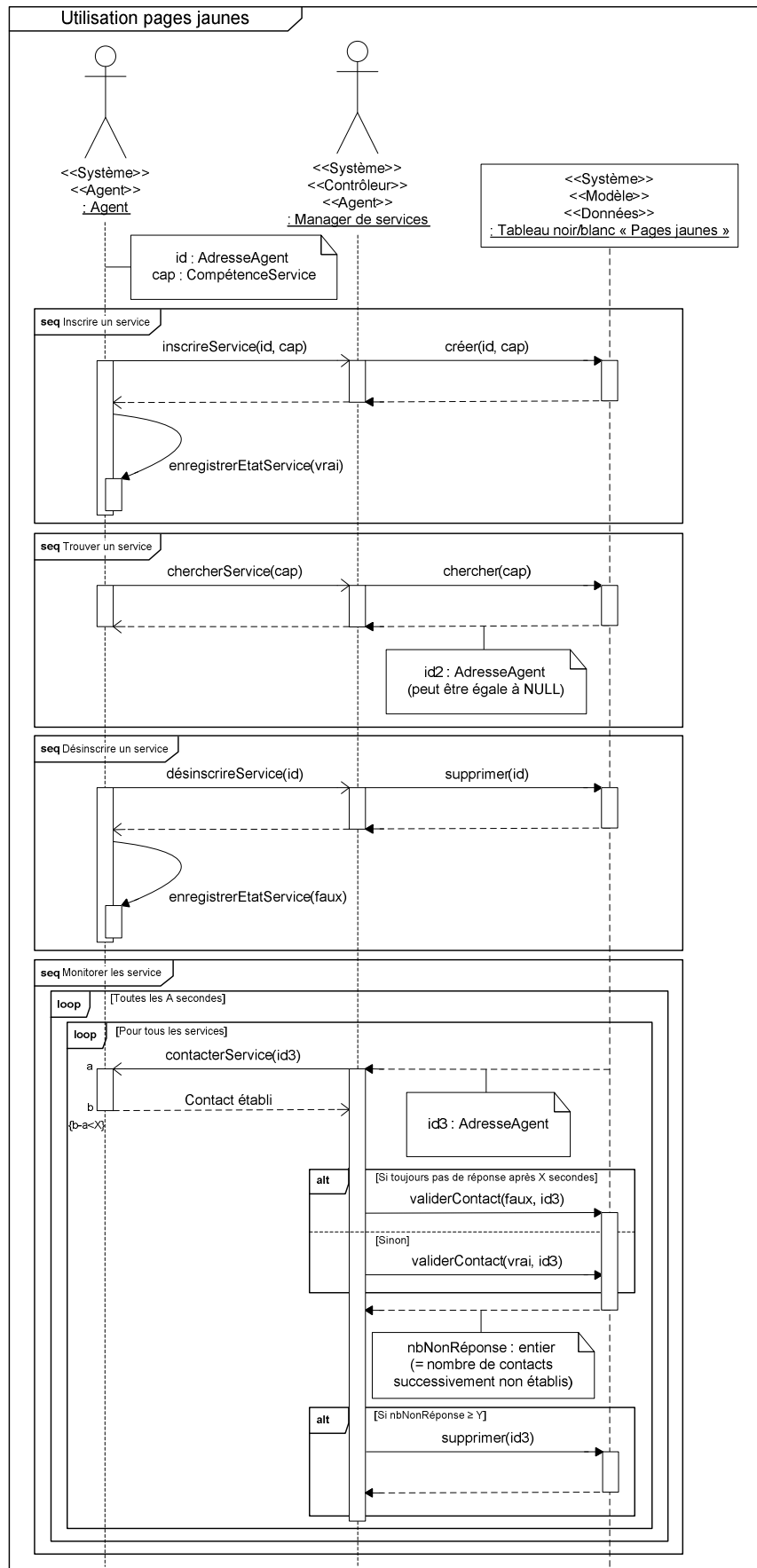


Figure 142 : Diagramme de séquence technique d'un service de pages jaunes

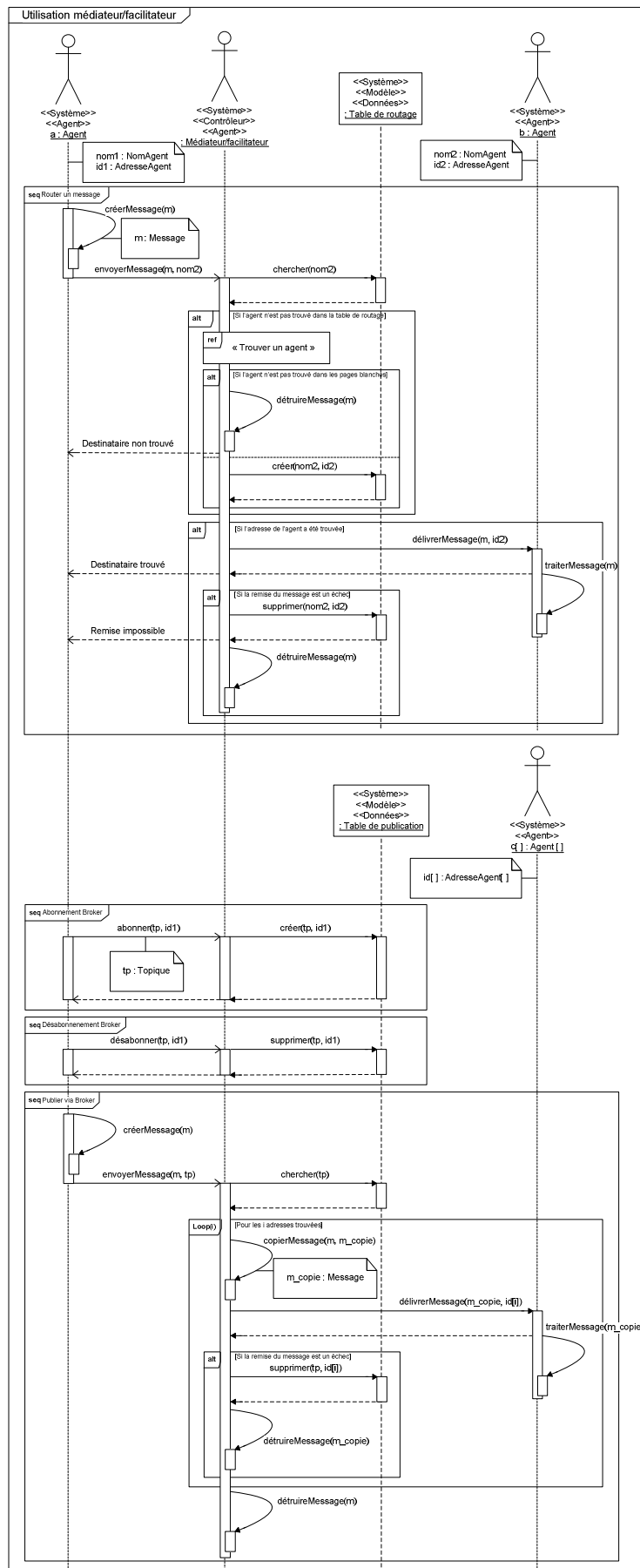


Figure 143 : Diagramme de séquence technique d'un médiateur/facilitateur

3) L'élaboration des diagrammes de classe techniques

FICHE GUIDE – FG17	
Activité 5 : Conception	
Sous-activité 5.3 : Elaboration des diagrammes de classe techniques	
Objectif	Définir toutes les classes (incluant les classes techniques) et associations pour tous les cas d'utilisation. Représenter les classes en utilisant le paradigme MVC (Burbeck, 1992).
Point de départ	Interactions entre les acteurs et tous les objets du système (incluant les objets techniques) définies pour tous les cas d'utilisation et DCL de l'activité 3
Point d'arrivée	Toutes les classes (incluant les classes techniques) et associations définies pour tous les cas d'utilisation
Démarche d'élaboration	
Pour chaque cas d'utilisation (CU) :	
<p>1 - Représenter les classes du CU en prenant comme base les objets définis dans le diagramme de séquence technique du CU et le DCL de l'activité 3. Répartir les classes en plusieurs catégories :</p> <ul style="list-style-type: none"> - Objets « Vue » - Objets « Contrôleur » - Objets « Modèle » - Classes techniques (Collection, Date...) identifiées uniquement lors de cette étape 	
<p>2 - Préciser les attributs des classes et leurs caractéristiques (visibilité, type, valeur initiale...) à partir de ceux identifiés dans le DSE (paramètres des messages).</p>	
<p>3 - Préciser les opérations des classes et leurs caractéristiques (paramètres avec type, type de résultat) à partir des messages du DES.</p>	
<p>4 - Déterminer les relations entre les classes :</p> <ul style="list-style-type: none"> - Nom de l'association - Multiplicité - Type d'association (composition, agrégation, association qualifiée, dépendance, héritage...) - Contraintes (ordonnées, non ordonnées...) 	

Tableau 25 : Fiche Guide 17 pour UP7 (d'après (Gabay & Gabay, 2008))

Pour chacun des cas d'utilisation pour lesquels nous venons de faire un diagramme de séquence technique, nous allons spécifiquement présenter toutes les classes d'objets et d'agents impliquées. Comme dans notre précédente sous-activité d'élaboration de diagrammes de classe (voir la sous-section **L'élaboration des diagrammes de classe**), nous procédons à des regroupements de cas en fonction de leur utilité fonctionnelle.

Notre premier diagramme de classe technique concerne la précondition aux cas d'utilisation « Installation et activation du système » (**Figure 144**) et il est relativement similaire à sa

version indépendante du choix technologique de notre framework (**Figure 120**). Comme pour nos autres cas d'utilisation impliquant une interaction avec l'utilisateur au travers d'une IHM, l'utilisation de ce framework a pour conséquence l'apparition d'agents de la Vue et du Contrôleur.

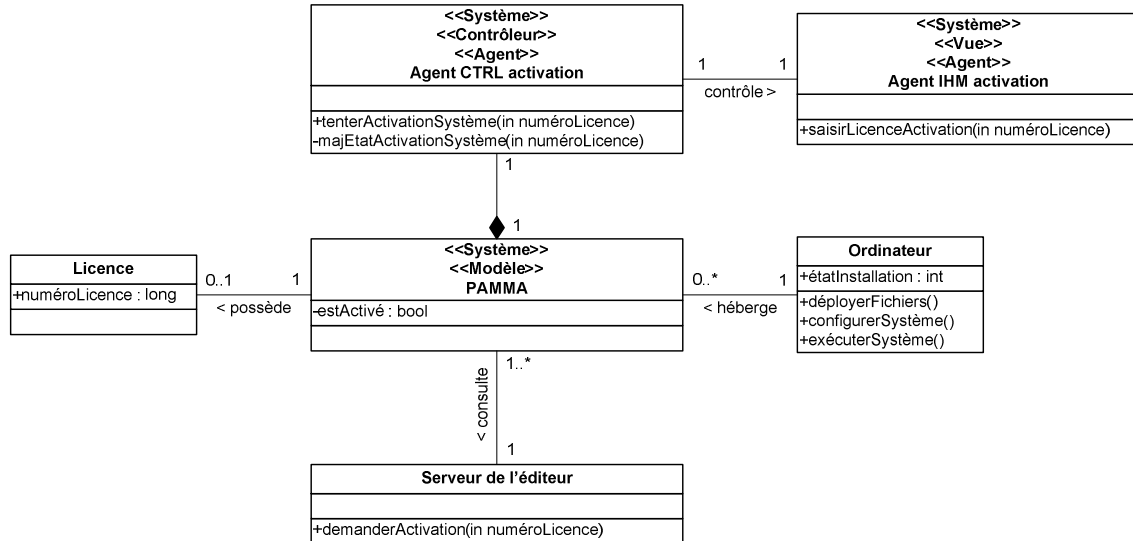


Figure 144 : Diagramme de classe technique de la précondition aux cas d'utilisation « Installation et activation du système »

Le diagramme suivant, qui est celui de la précondition aux cas d'utilisation « Authentification de l'utilisateur » (**Figure 145**), ne subit également que peu de modifications par rapport à sa précédente version (**Figure 121**).

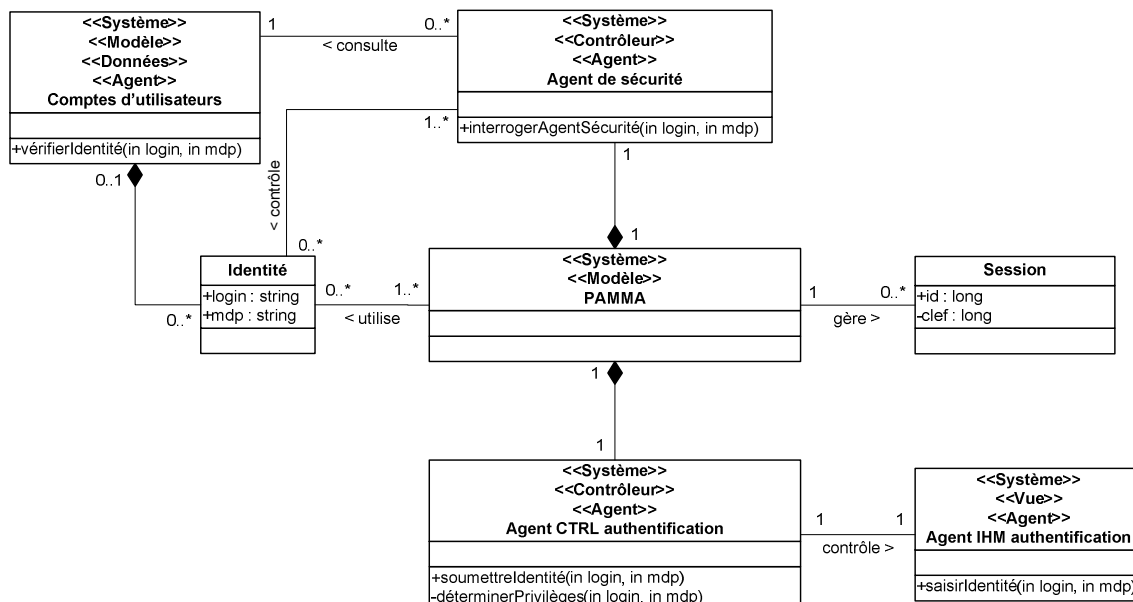


Figure 145 : Diagramme de classe technique de la précondition aux cas d'utilisation « Authentification de l'utilisateur »

L'ensemble des cas d'utilisation de gestion et d'utilisation des profils d'utilisateurs sont groupés ensemble (**Figure 146**). On remarquera l'apparition d'agents pour la gestion de données appartenant au Modèle.

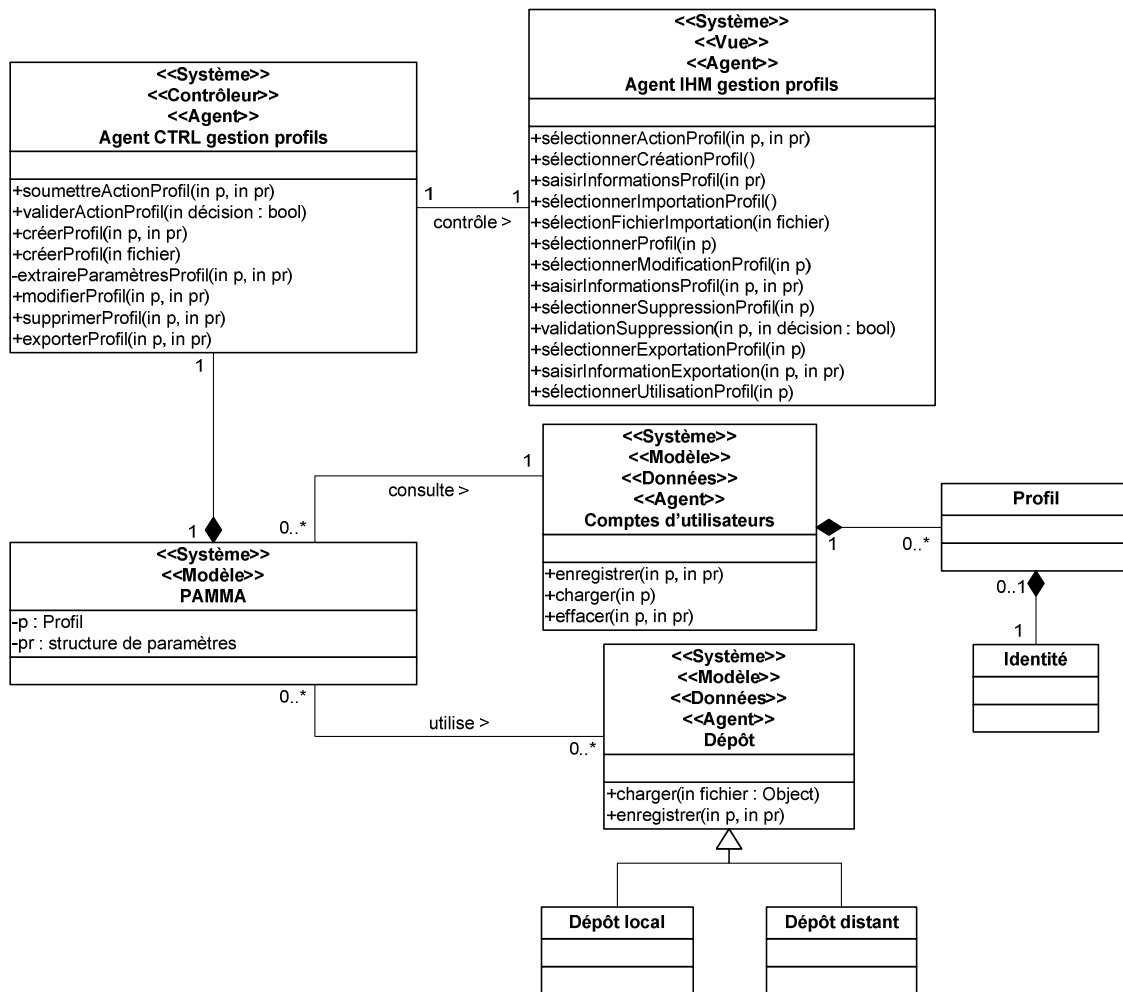


Figure 146 : Diagramme de classe technique de gestion et d'utilisation des profils (précondition aux cas d'utilisation « Valider une action sur un profil » et cas d'utilisation de 13 « Création d'un profil » à 17 « Connexion/Déconnexion d'un profil »)

Pour que notre modélisation englobe bien tous les éléments constitutifs de l'IHM, nous avons réalisé des diagrammes de classe techniques pour les écrans d'accueil du serveur (**Figure 147**) et du client (**Figure 148**), de la même manière que nous avons produit des diagrammes de séquence techniques pour ces deux écrans (**Figure 135** et **Figure 136**).

Le diagramme de classe technique suivant présente les cas d'utilisation de connexion/déconnexion entre PAMMA (**Figure 149**). On notera l'apparition de la classe des agents de connexion dont un agent sera instancié pour chaque connexion établie.

Vient ensuite le diagramme de classe technique de la consultation d'un dossier patient (**Figure 150**). Sur ce dernier, ont été intégrés les cas d'utilisation 6 « Calibrage de la simulation » et 12 « Recalibrage de la simulation » qui ne l'étaient pas précédemment (**Figure 122**). On notera que le modèle de simulation dispose de son propre agent.

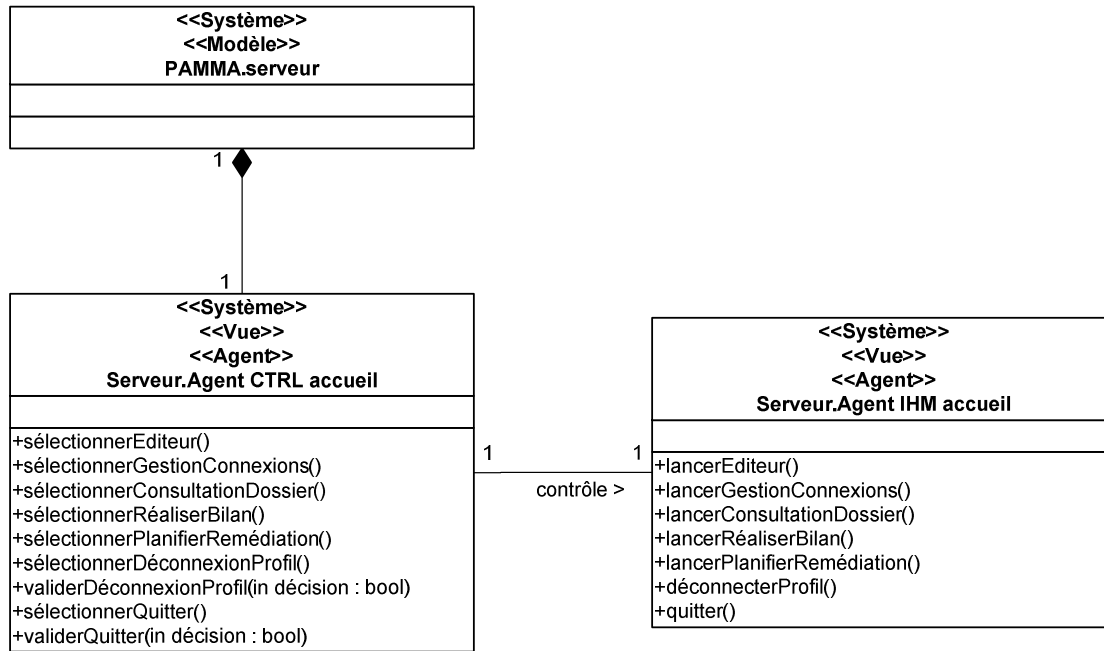


Figure 147 : Diagramme de classe technique de l'utilisation de l'accueil du serveur

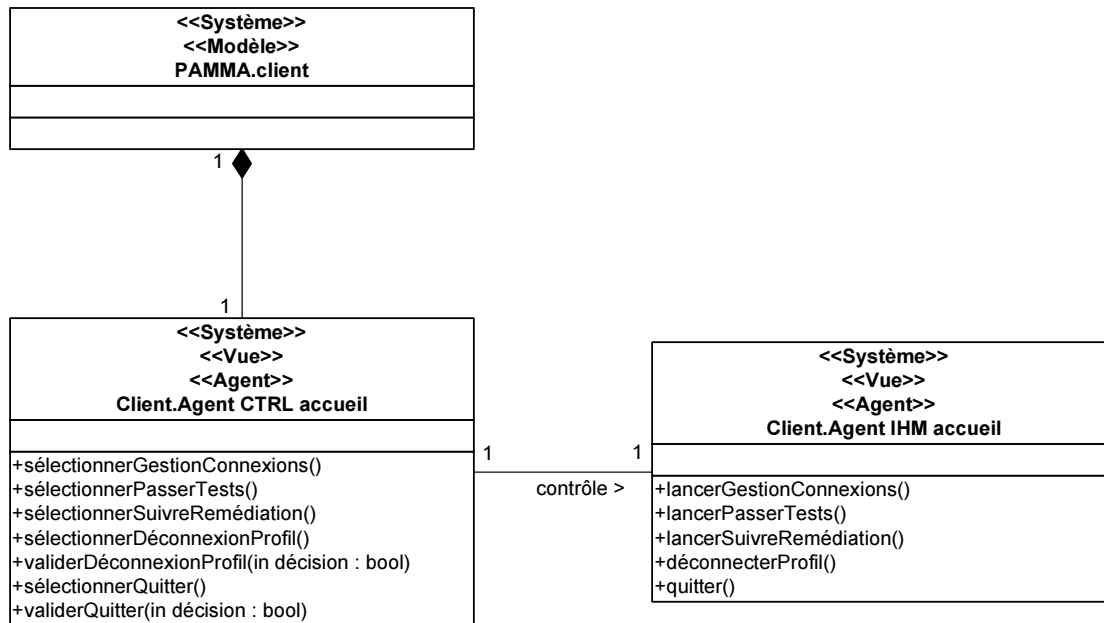


Figure 148 : Diagramme de classe technique de l'utilisation de l'accueil du client

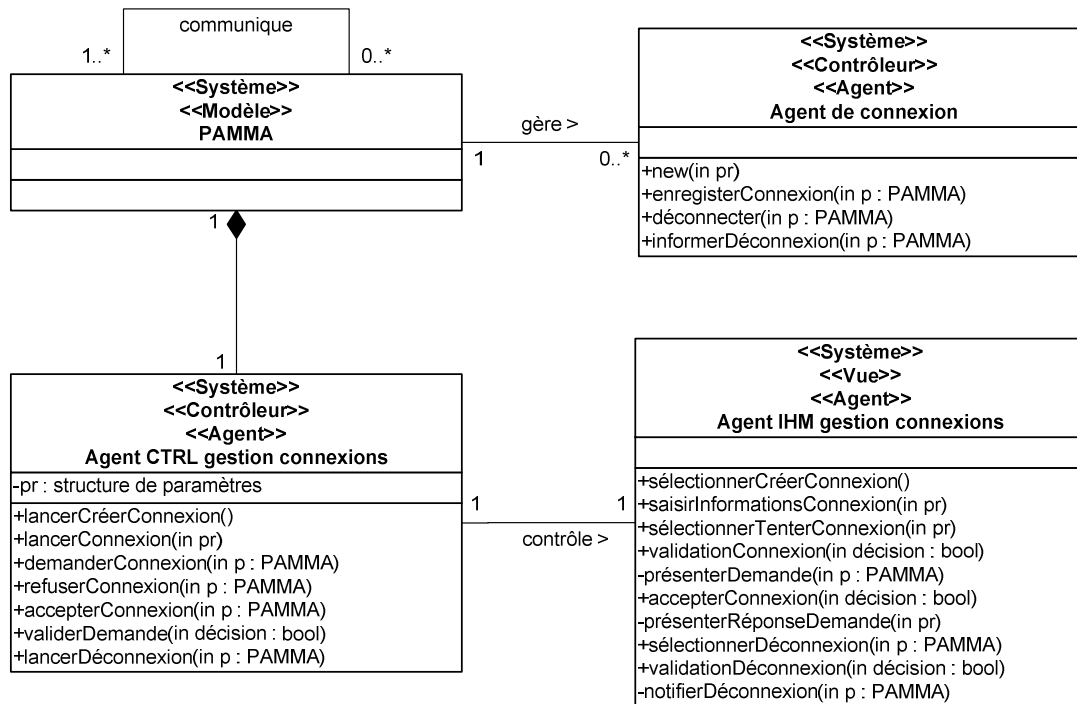


Figure 149 : Diagramme de classe technique de gestion des connexions (cas d'utilisation 19 « Connexion client-serveur » et 20 « Déconnexion client-serveur »)

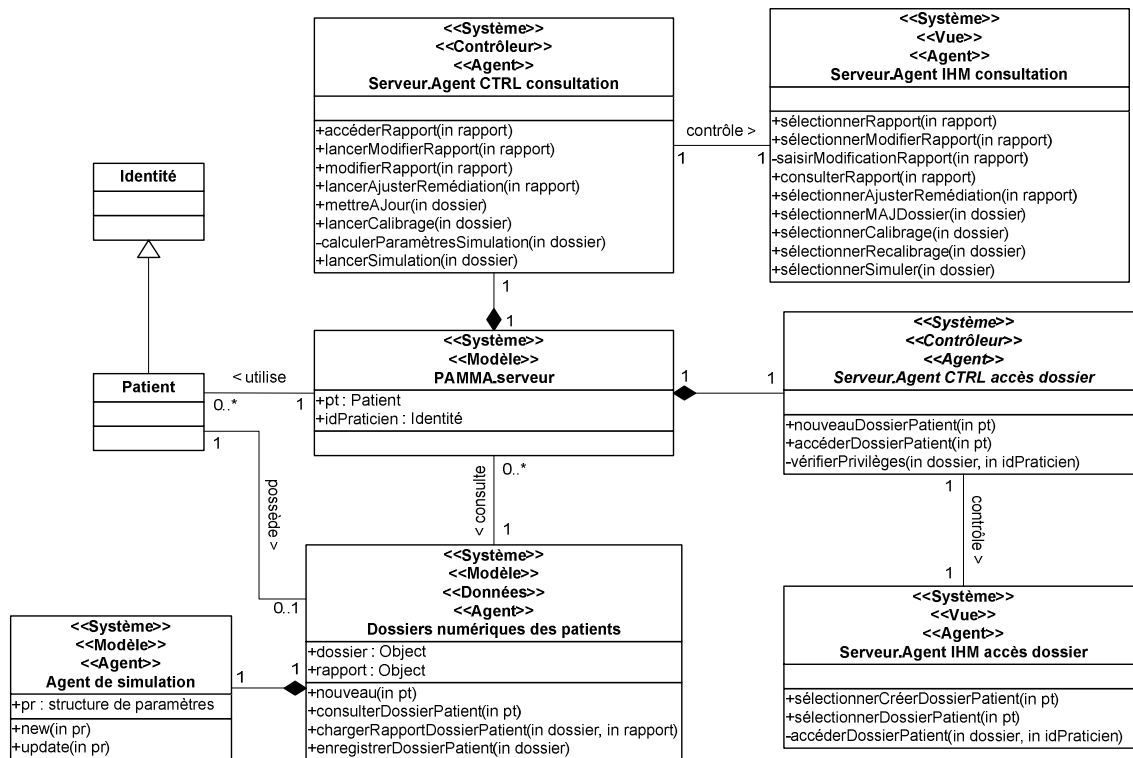


Figure 150 : Diagramme de classe technique de la consultation d'un dossier patient (précondition aux cas d'utilisation « Accéder à un dossier patient », cas d'utilisation 6)

« Calibrage de la simulation », 12 « Recalibrage de la simulation » et 18 « Consultation d'un dossier patient »)

Nos diagrammes suivants portent sur l'aide diagnostique (**Figure 151**) et le suivi de remédiation (**Figure 152**). Ils mettent, en particulier, en évidence l'utilité des agents de connexion pour la communication entre PAMMA.

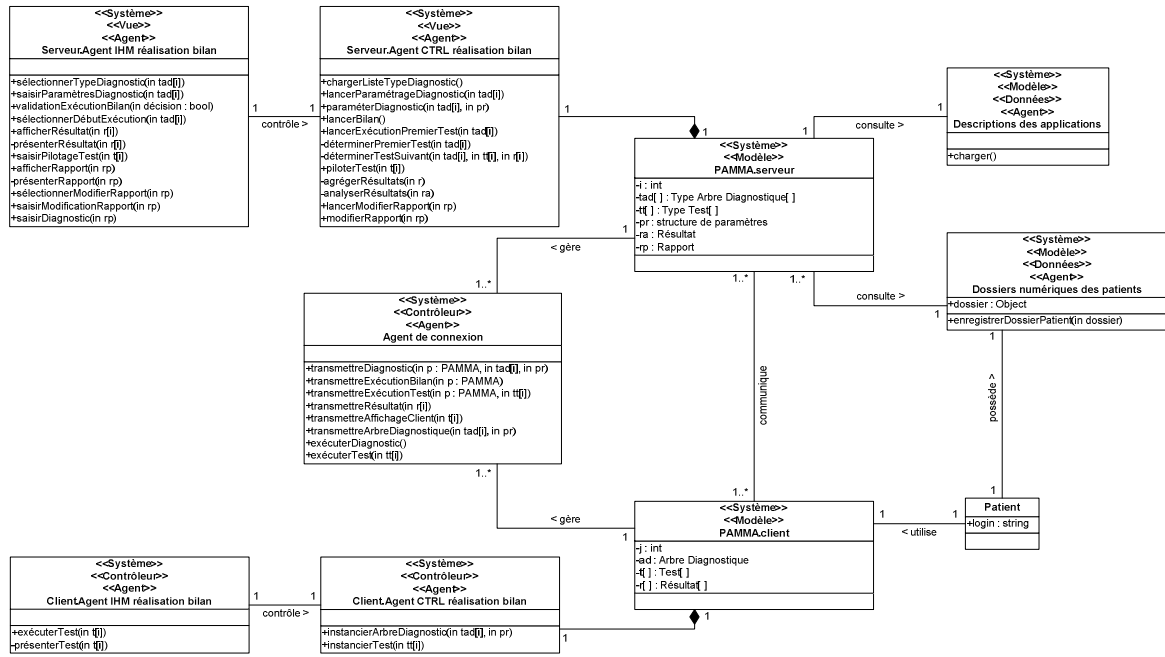


Figure 151 : Diagramme de classe technique d'utilisation de l'aide diagnostique cas d'utilisation de 3 « Réaliser un bilan » à 5 « Analyse et conclusion »)

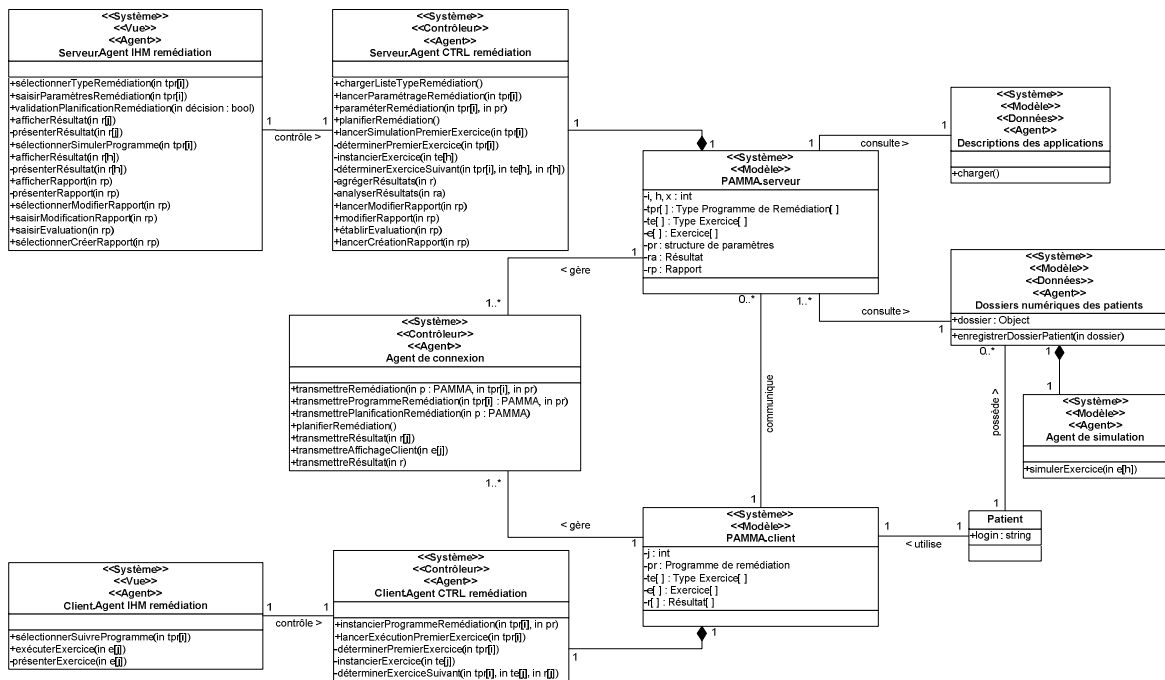


Figure 152 : Diagramme de classe technique d'utilisation du suivi de remédiation (cas d'utilisation de 7 « Planifier le programme de remédiation » à 11 « Ajuster la remédiation »)

Notre dernier diagramme de classe technique porte sur les services de pages blanches, de pages jaunes et de médiateur/facilitateur (**Figure 153**). Il s'agit des services indispensables à la réalisation ultérieure d'application sur la PAMMA dont nous avons présenté les principes de fonctionnement par des diagrammes de séquence techniques (**Figure 141**, **Figure 142** et **Figure 143**).

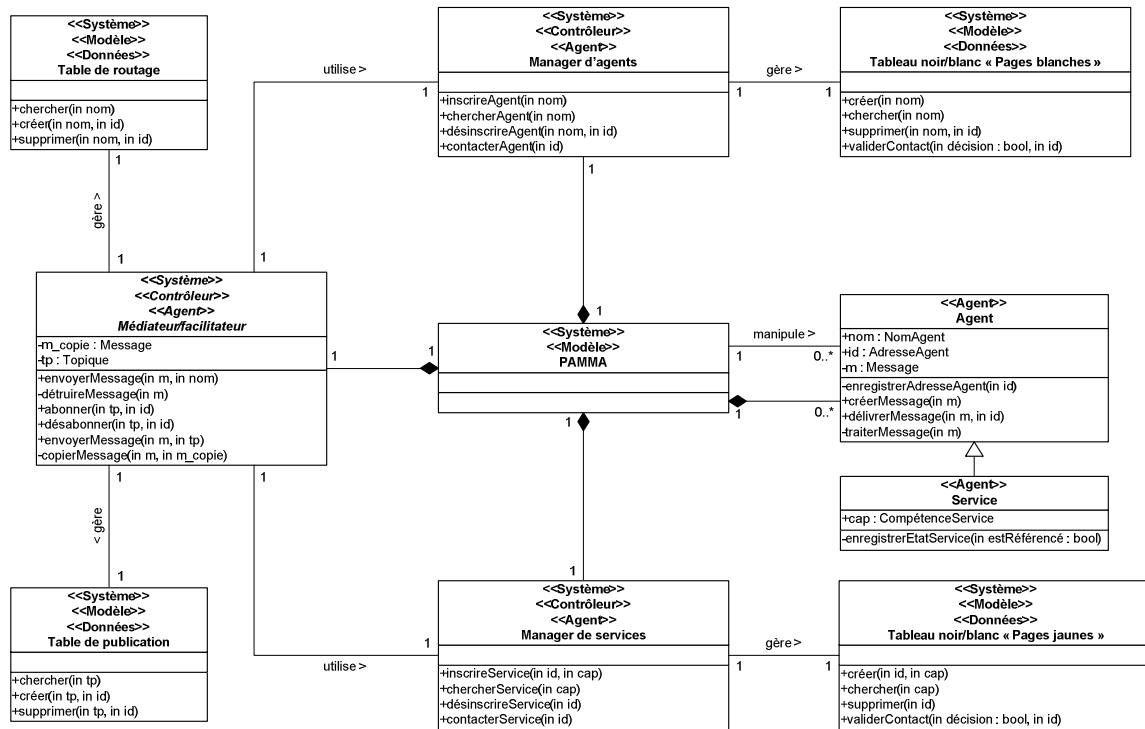


Figure 153 : Diagramme de classe technique des services de pages blanches, pages jaunes et médiateur/facilitateur

4) L'élaboration du diagramme de paquetage

FICHE GUIDE – FG18	
Activité 5 : Conception	
Sous-activité 5.4 : Elaboration du diagramme de paquetage	
Objectif	Regrouper l'ensemble des classes en paquetage pour avoir une vision globale et structurée du système étudié
Point de départ	Tous les diagrammes de classe par cas d'utilisation
Point d'arrivée	Regroupement de l'ensemble des classes dans un seul diagramme de paquetage
Démarche d'élaboration	
<p>1 - Regrouper l'ensemble des classes par ensembles homogènes. Chaque ensemble correspond à un paquetage (ex. Gestion Mail). L'ensemble peut correspondre à un découpage technique (architecture en couches) ou fonctionnel.</p> <p>2 - Déterminer les dépendances entre les paquetages :</p> <ul style="list-style-type: none"> - « import » - « access » - ... 	

Tableau 26 : Fiche Guide 18 pour UP7 (d'après (Gabay & Gabay, 2008))

Cette dernière sous-activité de conception vient conclure notre modélisation du système générique pour la prise en charge de la dyslexie. Elle consiste à proposer une vision structurée des classes qui composent le système. A ce titre, nous avons créé des paquetages nommés à partir des dénominations des cas d'utilisation qui adoptent une approche purement fonctionnelle, mais également des paquetages nommés à partir des noms des prototypes UML que nous avons précédemment définis et utilisés.

En plus des paquetages constitutifs du système, nous avons représenté, sur la **Figure 154**, les paquetages de notre framework. Conformément à ce que nous avons précédemment défini (voir la sous-section **La perspective du méta-système**), il apparaît une variété de vingt-sept différents paquetages possibles pour ce framework par combinaison des paquetages représentés sur la figure.

Finalement, on précisera, pour reboucler sur l'architecture globale du système (**Figure 129**), que les paquetages du noyau sont ceux inclus à celui de la PAMMA ou directement en relation avec lui. Dans le cas de notre application pour la prise en charge de la dyslexie, les paquetages des couches applicatives sont le serveur, le client et tous les paquetages reliés à l'un des deux.

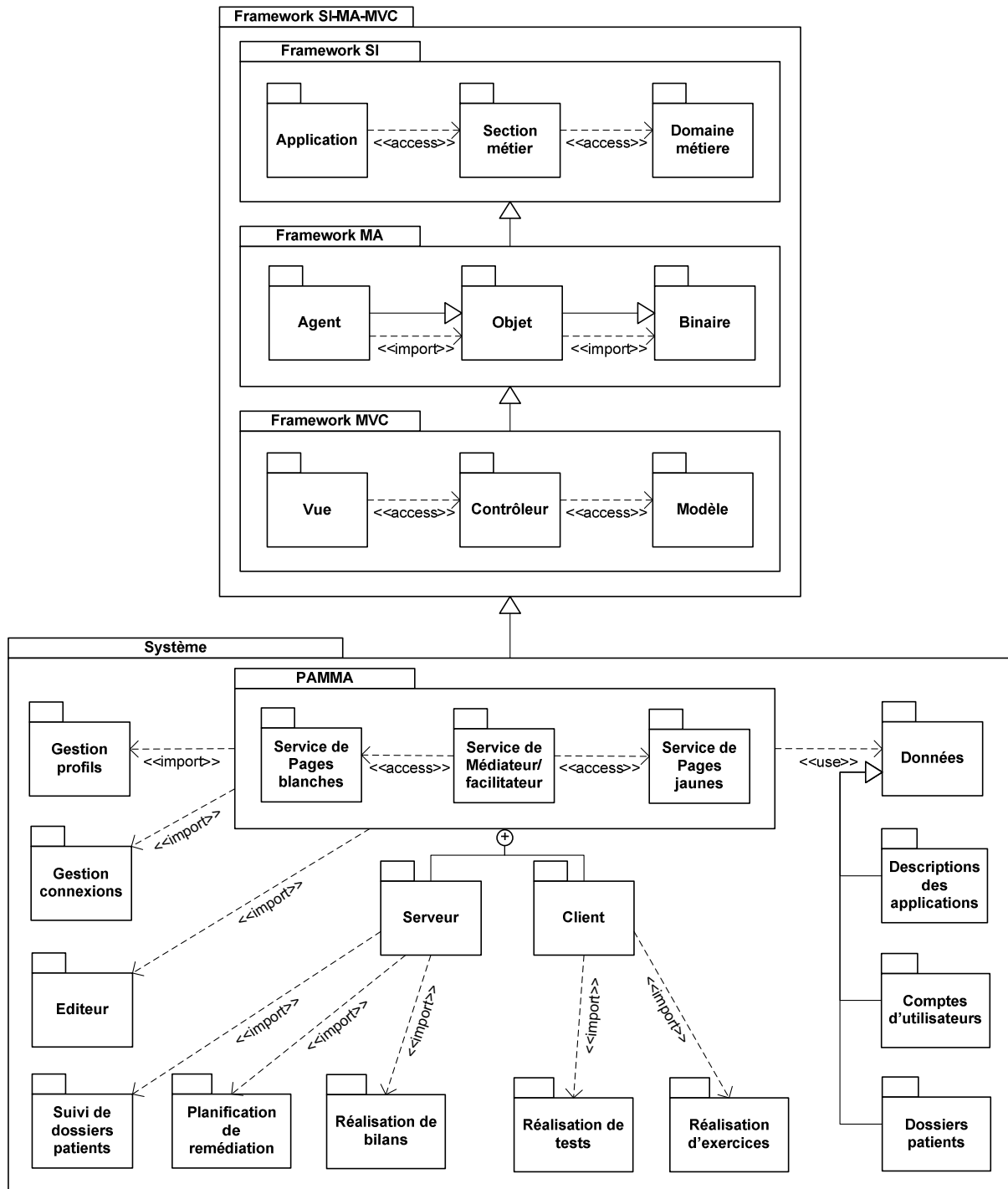


Figure 154 : Diagramme de paquetage

F) L'implémentation et les tests

Rappelons tout d'abord que notre objectif était seulement de modéliser le système (**MODEL1**). A ce titre, nous n'en produirons pas de version exécutable. Néanmoins, il nous paraît judicieux de présenter brièvement ce qui devrait être effectué lors des activités d'implémentation et de tests dans le cadre d'un processus de développement complet. En effet, la prise en compte entière de toutes les activités et/ou phases du processus de développement fait partie intégrante de notre démarche visant à maximiser la qualité finale d'un futur produit. On pourrait également préciser que, consécutivement à ces deux phases, nous pourrions aussi considérer des phases de déploiement et de maintenance (Rational Software, 2001). Cependant, de telles phases sortent largement du périmètre du développement de notre application.

Aucune méthode ni aucune norme actuelle, malgré toutes les mesures prises, ne peut absolument garantir la sûreté des logiciels à usage médical. Fox (2002) propose tout de même de bonnes pratiques pour maximiser la qualité, à la fois de la plate-forme logicielle et du contenu clinique spécifique qu'elle héberge :

- Le logiciel devrait être conçu, implémenté, testé et documenté en utilisant des méthodes généralement reconnues, notamment en ce qui concerne son processus de développement.
- Un plan de la qualité devrait être explicitement élaboré afin de couvrir toutes les phases d'implémentation, les tests et la maintenance du système.
- Les tests doivent être effectués en suivant des pratiques reconnues et faire l'objet d'une historisation.

La méthodologie de développement que nous avons adoptée nous met en conformité vis-à-vis du premier de ces trois points. Le deuxième point est assuré par l'implication continue d'experts métiers dans le développement, ainsi que par la mise à profit de connaissances scientifiques reconnues et documentées. Enfin, pour satisfaire au dernier point, chaque test va devoir être documenté afin de (Benington, 1987) :

- Détailler explicitement ses buts
- Spécifier les données entrantes
- Préciser la procédure de test
- Décrire les résultats et sorties attendus

L'idée derrière l'application de ces bonnes pratiques est la même que celle adoptée lors de toute instanciation d'un processus dérivant de l'UP : pouvoir retracer et contrôler les activités impliquées dans la réalisation du logiciel. A ce titre, nous allons maintenant formuler un certain nombre de recommandations qu'il faudrait idéalement suivre lors des phases d'implémentation et de test pour achever le développement de notre dispositif.

1) L'implémentation

L'implémentation correspond à la production réelle du logiciel, à sa programmation. Cette dernière est constituée de composants, d'instructions, de bibliothèques ou de fichiers. Gabay et Gabay (2008) précisent que sa réalisation représente une importante charge de

travail d'au moins 40% supérieur à n'importe laquelle de celles des autres phases. L'objet de cette phase d'implémentation est de (Rational Software, 2001) :

- Définir l'organisation en couche de code.
- Coder les structures de données sous forme d'objets, et dans notre cas d'agents.
- Éprouver les développements comme étant un tout.
- Intégrer ensemble les différents codes produits.

Une implémentation réussie du système doit être tolérante aux erreurs et aux pannes (Arney et al., 2014; Hatcliff et al., 2012). En effet, l'utilisation de notre système en milieu clinique, pour des considérations évidentes et relatives à la sécurité des personnes, se doit d'être souple en ce qui concerne les dysfonctionnements. Ainsi, le système doit, à tout prix, éviter de se trouver dans une situation de blocage/plantage menant à l'échec technique du processus de soins. Intégrer un système de journalisation des événements est prépondérant pour identifier, analyser et résoudre tout problème pouvant survenir au sein de l'architecture distribuée de notre dispositif (Vasserman & Hatcliff, 2014). A ce titre, la phase d'implémentation sera l'occasion de mettre en place une brique logicielle capable de journaliser chaque interaction entre les différents composants de la solution. Cependant, si l'on considère les interactions entre les utilisateurs et la machine, il peut devenir assez délicat de déterminer des notations adaptées pour journaliser les interactions, en particulier si ces dernières sont de nature multimodale (Weiss et al., 2012).

Pour finir, on signalera qu'il est délicat de passer de la conception d'un SMA à son implémentation (Noël, 2012). Ainsi, il est plus que recommandé d'avoir recours à des outils d'assistance à l'implémentation. Dans notre cas, il pourrait donc être intéressant d'implémenter certains composants de notre solution à l'aide d'un framework multi-agents existant (van Breemen, 2003) tel que JADE (Bellifemine et al., 2003, 2008; Telecom Italia SpA, 2014), MASK (Occello et al., 2003; Occello, 2003), MaDKit (Gutknecht et al., 2000, 2001; LIRMM, 2014), ZEUS (Nwana et al., 1999) ou encore JACK (AOS Group, 2008, 2014). Il y a d'ailleurs des retours d'expériences positifs sur l'utilisation de ces frameworks pour la réalisation de systèmes tels que le nôtre (pour un exemple, consulter (Han et al., 2006)). Cependant, on remarquera que ces frameworks n'offrent pas réellement de langage orienté agent (Giorgini & Henderson-Sellers, 2005) et que, dans le contexte de nos développements, il nous faut pouvoir disposer de plus d'outils que ceux communément offerts par ces plateformes. A ce titre, notre solution est de concevoir notre propre plate-forme d'implémentation (voir la section **L'intégration horizontale : support des processus métiers collaboratifs**). Cette plate-forme n'est autre que le noyau de la PAMMA dont nous avons déjà présenté de nombreux éléments (**Figure 9** et **Figure 32**).

2) Les tests

Notre système doit satisfaire un certain nombre d'exigences fonctionnelles et techniques. Pratiquer des tests sur ce système, et sur les prototypes menant à sa réalisation effective, va permettre de s'assurer que ces exigences sont correctement prises en compte. On peut d'ailleurs préciser que (Arney et al., 2014) :

- Les exigences des utilisateurs servent à valider que le système répond effectivement aux besoins des utilisateurs.
- Les exigences techniques servent à vérifier le fonctionnement correct du système.

Les batteries de tests permettent de confirmer que l'implémentation de toutes ces exigences, ainsi que l'intégration de tous les composants, sont correctes. Pour cela, il faut s'assurer que ces composants implémentent avec succès les cas d'utilisation (Khamis & Abdelmonem, 2002). Les tests conduits de la sorte peuvent être de différentes natures : tests unitaires, tests d'intégration, tests de réception, tests de performance ou encore tests de non-régression (Gabay & Gabay, 2008). Dans leur article de 2010, Mathur et Malik présentent d'ailleurs un résumé des différents tests qu'un logiciel doit subir afin d'améliorer sa maintenabilité et ses performances.

Les choix architecturaux conditionnent la manière dont vont être pratiqués les tests. On pourra noter que la séparation de l'application en couches, comme dans le cadre du design pattern MVC, permet de faciliter globalement la testabilité du système (Bragge, 2013). D'une manière générale, toute modélisation entraînant un cloisonnement de fonctionnalités sous forme de composants, et a fortiori la définition d'interfaces, définit des éléments pouvant et méritant d'être testés de façon indépendante. Dans le cas de notre PAMMA, garantir l'interopérabilité des composants développés est d'ailleurs primordial, voire critique. Ainsi, pour garantir cette interopérabilité, les tests pratiqués doivent permettre de s'assurer que (Hatcliff et al., 2012) :

- Chaque composant est conforme aux interfaces prédéfinies.
- Les assemblages de composants permettent effectivement de réaliser ce pour quoi ils sont prévus.

Pour résumer, l'objectif des tests est l'évaluation des aspects techniques et fonctionnels de chaque brique du système. Tant qu'ils ne permettent pas d'établir la conformité des composants implémentés, il faudra itérativement procéder à des corrections du code de ces composants et relancer les tests en permettant l'évaluation. En particulier, dans le contexte du développement d'applications médicales telles que la nôtre, il est important d'identifier les risques/dangers pour les patients sous la forme d'une liste/tableau (Fox, 2002). A ce titre, il faut faire apparaître et tester, pour chaque processus de soins qui sera développé, les moyens mis en œuvre à l'exécution pour mitiger et/ou prévenir les risques/dangers pour les patients. Ces dispositifs de surveillance de la bonne tenue de l'exécution du programme sont, en quelque sorte, des procédures de tests persistantes et continues, dont le cycle de vie s'étend largement au-delà de la seule phase de développement du dispositif.

Nous pouvons également apporter quelques précisions relatives aux tests spécifiques à pratiquer sur un SMA. Dans le contexte de la validation des protocoles d'interaction utilisés ainsi que de l'évaluation de leur conformité et de leur interopérabilité, il faut, par exemple, procéder à des tests pour vérifier qu'il n'y a pas de cycle bloquant ou d'interblocage et que l'implémentation du protocole exhibe un comportement identique à celui de son modèle formel (Huget, 2001). Ces tests peuvent être menés de manière automatique sur les graphes qui permettent la représentation des protocoles. L'étude de telles techniques de validation est néanmoins complexe ainsi qu'hors du périmètre d'étude de ce manuscrit.

Finalement on ajoutera que, dans un cadre de développement idéal, les tests et vérifications du code devraient être effectués par des individus différents de ceux qui ont participé à les implémenter (Royce, 1970).

II) L'intégration horizontale : support des processus métiers collaboratifs

Dans cette partie nous allons décrire, d'un point de vue un peu plus technique que celui adopté précédemment, les éléments de la plate-forme thérapeutique soutenant le travail collaboratif. Cette étude vient en complément de notre objectif **MODEL1** et a pour but de présenter des pistes concrètes concernant le support d'intégration horizontale afin de procéder ultérieurement à une implémentation du système. Rappelons que cette partie de notre système vise à permettre l'intégration des processus métiers intra et inter-organisationnels. Dans le contexte de notre développement, l'intégration va s'effectuer à différents niveaux techniques (Heckel & Engels, 2002; Wangler & Paheerathan, 2000) :

- L'intégration des données
- L'intégration d'applications
- L'intégration des processus métiers

On peut remarquer que ces niveaux d'intégration peuvent être impliqués dans une intégration aussi bien verticale qu'horizontale. D'ailleurs, la différence entre ces deux types d'intégration ne se situe qu'à un niveau conceptuel, induit par la structure hiérarchique et fonctionnelle de la ou des organisations concernées par l'action d'intégration. En effet, alors que l'intégration verticale s'adresse aux personnels directement affairant à un objectif de création de produits et/ou services, l'intégration horizontale vise à supporter la direction/organisation des interactions au sein de ces personnels. A ce titre, les moyens mis en œuvre pour réaliser ces deux types d'intégration sont similaires, voire identiques. On pourra ainsi signaler que ces types d'intégration s'inscrivent dans un même framework décrivant les relations entre organisation, processus métiers et données (**Figure 155**).

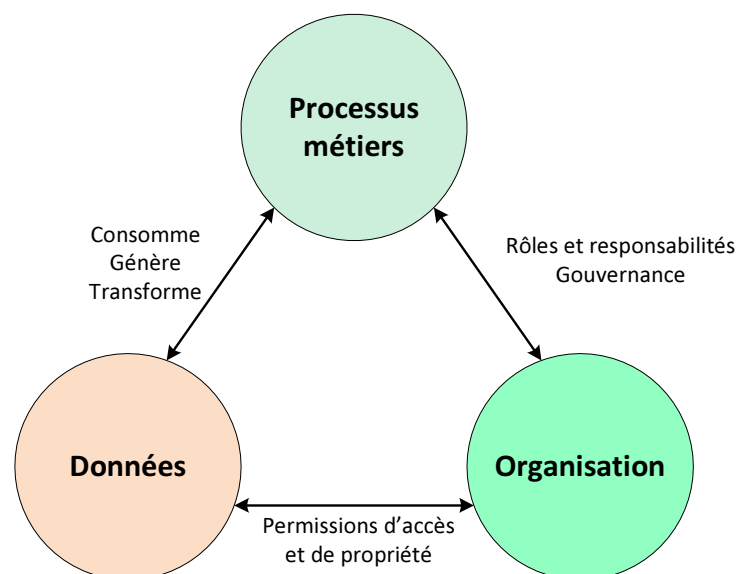


Figure 155 : Schématisation des relations entre organisation, processus métiers et informations (d'après (Hollingsworth, 2004))

Ce framework stipule que des acteurs rassemblés au sein d'une organisation collaborent autour de processus métiers. Chaque acteur, au travers de ces processus, utilise, produit et détruit des données conformément aux pouvoirs qui lui sont alloués au sein de cette organisation. L'intégration, que nous traiterons ici, consiste à assister informatiquement une restriction de telles relations survenant dans une entreprise (ou entre plusieurs entreprises)(Heckel & Engels, 2002). Dans le contexte médical, une entreprise peut, par exemple, prendre la forme d'un cabinet médical ou encore d'un service hospitalier.

Dans la présente partie, nous nous concentrerons sur l'intégration horizontale, et nous verrons comment les niveaux d'intégration, précédemment énoncés, peuvent la soutenir au travers de choix architecturaux et technologiques. En particulier, nous présenterons les exigences requises pour une telle intégration et nous verrons comment les SMA peuvent être un moyen de la réaliser (Kishore, Zhang, & Ramesh, 2006).

A) L'intégration des données

L'intégration des données passe classiquement soit par la mise en place de passerelles d'accès vers différentes bases de données, dont les schémas structurels sont connus, à l'aide de SQL (International Organization for Standardization, 2011), soit en procédant à un portage de données existantes dans un format qui convient mieux au nouvel usage souhaité. Notre modélisation des données alimentant le système, comme nous l'avons présenté précédemment (voir la sous-section **Modéliser la gestion des données**), nous permet de mettre en place l'une ou l'autre de ces approches de façon transparente pour le cœur du système. En effet, la modélisation MA retenue est propice au cloisonnement des fonctionnalités et à l'abstraction des technologies qui sous-tendent ces fonctionnalités. Dans le cas de notre PAMMA, ces données seront principalement de nature médicale. La littérature sur les SMA propose d'ailleurs plusieurs manières, à un niveau macroscopique, de les gérer en fonction de leurs types (Fox, Johns, Rahmzadeh, & Thomson, 1996; Isern et al., 2010; Tu & Musen, 2001) :

- Sous la forme de bibliothèques de connaissances, permettant de consulter et réviser l'état de savoirs médicaux ou encore d'informations relatives aux professionnels de santé et aux institutions auxquelles ils appartiennent (pour des exemples d'utilisation, consulter (Kostkova et al., 2003; Singh, Sachan, & Gangwar, 2012)). Il s'agit principalement de données de notoriété publique auxquelles un grand nombre d'utilisateurs différents peut être amené à avoir accès.
- Sous la forme de dossiers électroniques et virtuels des patients, permettant de recomposer les dossiers des dits patients à partir de sources hétérogènes et distribuées (pour des exemples d'utilisation, consulter (Cruz-Correia et al., 2005; Ferreira et al., 2009)). Par ailleurs, nous signalons l'existence d'un service nommé « **Master Patient Index** », destiné à identifier précisément un patient et de faire le lien avec les différentes données le concernant sur les différents systèmes de différents établissements de santé (Kilman & Forslund, 1997). Le MPI est un service distant jouant le rôle de médiateur, avec une gestion des droits d'accès, entre un système demandeur d'informations et ceux où elles sont stockées.
- Sous la forme d'une connaissance des procédures médicales, permettant la gestion de toutes données utiles afin de répondre en temps réel aux besoins des utilisateurs

du système (pour des exemples d'utilisation, consulter (Hashmi, Abidi, & Cheah, 2002; Rodríguez, Favela, Preciado, & Vizcaíno, 2005)). Les procédures sont aussi appelées « protocoles médicaux » ou encore « processus de soins ». Ces procédures peuvent ainsi être représentées sous la forme de workflows impliquant les patients et les professionnels de santé (Leonardi et al., 2007).

Les architectures qui sous-tendent ces diverses mécaniques d'intégration de données sont autant de façons de manipuler des données médicales de différents types (connaissances scientifiques, données des patients, état des ressources humaines et matérielles, etc.). En particulier, elles apportent des solutions à des problèmes que nous rencontrerons pour la réalisation de notre PAMMA et de ses applications. A ce titre, elles peuvent donc être considérées comme de véritables design patterns pour la gestion de données médicales. Nous ne rentrerons cependant pas dans les détails de ces différents patterns mais donnerons une vue d'ensemble de ce domaine de recherche. Ainsi, notre proposition d'architecture de la PAMMA disposera de composants paramétrables permettant n'importe lequel de ces niveaux d'intégration de données.

Le paradigme MA offre différents types d'agents pour accéder à ces données. La nature de ces agents dépend grandement des fonctionnalités, proposées à l'utilisateur, dans lesquelles ils sont impliqués. Par exemple, il existe des agents de recherche intelligents (par exemple, consulter (Faulkner, Kolp, Nguyen, Coyette, & Do, 2004; Hashmi et al., 2002)), des agents proactifs d'alertes ou encore des agents contrôlant l'état de mise à jour des données (Kostkova et al., 2003). La faculté de ces agents à être distribués va permettre l'ubiquité du système. Aussi, des mesures de sécurité particulières doivent être adoptées pour garantir la confidentialité des données échangées entre composants distribués, en particulier dans le cas de la mise en œuvre de technologies de mise en réseau sans fil (Ferreira et al., 2009).

1) La sécurité

L'utilisation d'agents pousse nécessairement à s'interroger sur les aspects liés à la sécurité de leur mise en œuvre au sein de notre système (Alsinet, Béjar, Fernández, & Manyà, 2000; Moreno, Sánchez, & Isern, 2003; Poggi, Rimassa, & Tomaiuolo, 2001; Poslad, Charlton, & Calisti, 2003). Effectivement, les agents présents dans le système agissant de façon distribuée et sans contrôle central, il faut s'assurer qu'aucun agent malveillant ne pénètre le système qui, une fois investi, serait très vulnérable. Ainsi, étant donné l'importance de la confidentialité et de la sûreté exigées pour un système tel que le nôtre, des mesures visant à assurer la sécurité doivent prendre place à différents niveaux (Choe & Yoo, 2008; Ferreira et al., 2009; Sycara et al., 2003; Vieira-Marques et al., 2006):

- Au niveau de l'accès au système, c'est-à-dire qu'il faut contrôler qu'un utilisateur, ou de manière plus générale un agent, a effectivement le droit de solliciter tout ou partie des ressources et/ou fonctionnalités du système (à l'aide, par exemple, d'un protocole LDAP (The Internet Engineering Task Force, 2014a) ou d'une approche RBAC (National Institute of Standards and Technology, 2014)).
- Au niveau des données, c'est-à-dire que les données transitant à travers le réseau doivent être protégées en lecture et en écriture afin d'en garantir la confidentialité et l'intégrité (pour un exemple de telles techniques, consulter (Karjoth, Asokan, & Gülcü, 1998)).

- Au niveau du code des agents mobiles, c'est-à-dire que les agents pouvant se déplacer entre différentes PAMMA doivent pouvoir être authentifiés et vérifiés avant exécution (pour un exemple de telles techniques, consulter (Ametller, Robles, & Ortega-Ruiz, 2004)).

A tous ces niveaux, des agents spécialisés, ainsi que des composants internes aux agents, peuvent assurer la sécurité du système (Shakshuki et al., 2004). En particulier, un module d'échange sécurisé de messages peut être intégré au sein de chaque agent (Novák, Rollo, Hodík, & Vlček, 2003). D'autre part, la FIPA (IEEE Computer Society, 2014) propose des mesures de sécurité qui peuvent être mises en place pour les SMA (Poslad et al., 2003).

Les problèmes d'accès peuvent être gérés à l'aide d'un mécanisme de privilèges (Poggi et al., 2001). A ce titre, les agents, qui ont la faculté d'agir au nom d'un autre agent, peuvent avoir recours à un système de délégation (Nagaratnam & Lea, 1998). Dans ce contexte, il existe trois approches pour gérer les privilèges d'un agent en représentant un autre :

- Pas de délégation, c'est-à-dire que chaque agent dispose uniquement de ses privilèges propres.
- Délégation simple, qui permet à un agent d'agir en utilisant les privilèges du premier agent à l'initiative d'une série de délégation de tâches. Cette transmission de privilèges peut se répéter de façon limitée ou illimitée.
- Délégation en cascade, où les privilèges des agents se cumulent à chaque fois qu'un agent est amené à en représenter un autre.

Les données transitant via le réseau peuvent, quant à elles, être échangées dans le cadre d'une interaction régie à l'aide d'un protocole sécurisé (Isern, 2008)(comme, par exemple, SSL (The Internet Engineering Task Force, 2014b)) et également faire l'objet d'un chiffrement, d'utilisation de certificats ou de hachage. Il est d'ailleurs possible, au sein d'un même système, de composer ces différents moyens pour sécuriser les données en fonction des capacités des matériels disponibles et du niveau de confidentialité souhaité (Sulaiman et al., 2009). En procédant ainsi, les données ne sont naturellement compréhensibles que par les interacteurs participant effectivement à l'échange au travers du protocole sécurisé retenu. Nous ne traiterons pas plus en profondeur cet aspect lié au chiffrement des données, qui est hors du périmètre de ce travail.

Pour finir, on pourra remarquer que la protection du code des agents peut se faire au niveau de la plate-forme ou des agents eux-mêmes. Ce deuxième cas est évidemment meilleur que le premier, d'un point de vue de l'interopérabilité, car il n'implique aucune modification de la plate-forme dans le cas, par exemple, de changement de la technique de chiffrement. A contrario, déléguer entièrement cette protection aux agents peut faciliter l'entrée dans le système d'agents malicieux. Un bon compromis peut ainsi être de baser cette protection de manière hybride entre la plate-forme et les agents (Ametller et al., 2004).

De cette analyse ressort la nécessité pour notre PAMMA de disposer d'un agent en charge de l'authentification des agents, de la gestion des certificats et des clefs de chiffrement, ainsi que pour chaque agent d'intégrer un module de sécurisation des messages entrants et sortants.

2) La standardisation

Pour qu'un système, ou un composant d'un système, soit intégrable dans un environnement dynamique, il se doit de disposer d'interfaces clairement définies afin d'être interopérable. Un moyen de faciliter cette interopérabilité est d'avoir recours à l'utilisation de standards. Dans le contexte de l'intégration de données à caractère médical, on peut identifier quatre niveaux de standardisation (Sachdeva & Bhalla, 2012) :

- Le contenu, qui aborde le problème du codage des informations. Une terminologie normalisée est une exigence essentielle des applications de santé afin d'assurer des diagnostics et des traitements précis.
- La construction, qui met l'accent sur la structuration des éléments du SI.
- La technologie, qui stipule des règles concernant les composants et interfaces du système.
- L'organisation, qui s'intéresse aux changements organisationnels induits par l'utilisation du système.

Une telle standardisation est un mécanisme favorisant la conformité du système vis-à-vis des usages du domaine, et éventuellement de la législation, au travers de l'utilisation de normes et de protocoles reconnus. C'est également un moyen, dans le cadre de l'intégration horizontale, de permettre l'interopérabilité et la réutilisabilité de tout ou partie des éléments du système (Laleci et al., 2008; Madni & Sievers, 2014; Wright & Sittig, 2008). En particulier, l'interopérabilité d'éléments du système doit être construite en s'appuyant sur les standards d'échanges de données (Arney et al., 2014; Bates et al., 2001; National Transitions of Care Coalition, 2010; Xu, Sauquet, Degoulet, & Jaulent, 2003). Effectivement, adopter une standardisation des échanges entre systèmes permet de disposer d'un référentiel commun qui aide à passer de définitions d'interfaces « plusieurs-à-plusieurs » à des définitions d'interfaces « un-à-plusieurs » (Johnson et al., 2001)(voir, par exemple, (Ferrara, 1996)). Un tel ensemble de systèmes devient alors plus facilement maintenable et extensible dans la mesure où les modifications d'interfaces affichent alors une complexité linéaire, au lieu d'une complexité exponentielle auparavant. La standardisation des données permet également de disposer d'informations partageables (Greenes, 2014) évitant ainsi, par exemple, les efforts de duplication de workflows (Pattison-Gordon et al., 1996). Il y a là la même volonté d'uniformisation, en vue de permettre l'interopérabilité des systèmes, que dans le passage d'une stratégie d'intégration point-à-point à une architecture centrée autour d'un broker (voir la section **L'intégration d'applications**).

Les standards de communication sont très importants car ils conditionnent l'architecture (Maier, 1998). En effet, l'architecture d'un système informatique s'articulant autour de la communication entre ses composants, l'aspect protocolaire, qui est lié à cette communication, est prépondérant. Il existe différents standards pour spécifier les modalités techniques d'échanges de données informatisées, c'est-à-dire les formats des messages, entre systèmes (Hatcliff et al., 2012; Kalra, 2006; Kilman & Forslund, 1997). L'utilisation de telles normes a pour but premier la réalisation d'économies substantielles à un niveau administratif (Sainfort et al., 2007). Dans le domaine médical, ces standards peuvent porter sur les données transmises par des capteurs, celles récupérées à partir d'un DMP, ou plus généralement pour tous les types d'informations manipulées par un SI à vocation médicale (Laleci et al., 2008). D'autre part, elles servent également à garantir la bonne interprétation

des données lors de leurs échanges entre différents systèmes. Pour illustrer notre propos, voici une énumération non exhaustive de consortia et des standards actuels qu'ils présentent :

- Les Nations Unies, qui ont publié le framework EDIFACT (United Nations, 2014) pour faciliter les échanges de données et dont l'application s'étend à plus de domaines qu'au domaine médical seul.
- CEN (Comité Européen de Normalisation, 2014), au travers de son Technical Committee 251, qui propose le modèle EN 13606 utilisé pour les DMP.
- HL7 (Health Level Seven International, 2014), qui développe son modèle HL7 RIM destiné à la représentation de données médicales (Schadow, Russler, Mead, & McDonald, 2000)(pour des exemples de systèmes utilisant ce standard, consulter (Czibula et al., 2008; Mandellos et al., 2009; Tu et al., 2007)).
- NEMA (National Electrical Manufacturers Association, 2015), qui maintient le standard destiné à la gestion et au partage d'images médicales DICOM (National Electrical Manufacturers Association, 2014).
- ACNS (American Clinical Neurophysiology Society, 2015), qui propose un format d'échange standard pour les données EEG appelé ACNS TS1 (American Clinical Neurophysiology Society, 2008)(anciennement ASTM E1467-94 (American Society for Testing and Materials, 2000)).
- IHTSDO (International Health Terminology Standards Development Organisation, 2015), qui maintient la terminologie d'actes médicaux SNOMED CT (International Health Terminology Standards Development Organisation, 2014).
- openEHR (openEHR Foundation, 2015), qui entretient des spécifications et implémentations open source pour DMP.
- SOAP (World Wide Web Consortium, 2007), qui repose sur XML et offre une solution générique d'échanges de données entre systèmes utilisant des objets. Ce standard est de plus haut niveau d'abstraction que les précédents que nous avons cités et son utilité ne se limite évidemment pas qu'au domaine médical.

Il existe naturellement d'autres initiatives similaires telles que IHE (IHE International, 2013) ou CDISC (Clinical Data Interchange Standards Consortium, 2015). On notera qu'il existe des outils de middleware qui permettent le support et l'intégration de messages relevant de ces différents standards (par exemple, Mirth Connect (Quality Systems Inc., 2014)). On signalera également que, dans le cadre de l'aide à la décision, aucun standard ne semble réellement se distinguer et être massivement adopté (Kawamoto & Greenes, 2014). Ceci peut s'expliquer par la complexité des standards existants ainsi que par le manque d'outils couvrant l'intégralité de leur processus de développement de la modélisation à l'exécution.

Dans le but de normaliser la terminologie médicale employée, c'est-à-dire les règles portant sur l'information contenue/échangée, des ontologies sont fréquemment utilisées (Elkin, Brown, & Chute, 2001). Cela revient à proposer un middleware pour faciliter la communication entre un service de workflow, des applications et des SI. Un tel service peut être assuré à l'aide d'un outil tel que Protégé (Stanford Center for Biomedical Informatics Research, 2014), GEODE-CM (Greenes et al., 1999; Stoufflet, Ohno-Machado, Deibel, Lee, & Greenes, 1996) ou encore Ontolingua (Gruber, 1993; Stanford University, 2008). Ce type de représentations des relations entre les concepts manipulés par le système pourra être exploité par les agents pendant leurs actes de communication (Chen, Hsu, Dayal, & Griss, 1999; Nealon & Moreno, 2002, 2003)(comme, par exemple, en utilisant KIF (Genesereth &

Fikes, 1992)). Parmi toutes les terminologies standardisées existantes, nous pouvons en citer les plus répandues :

- MeSH (U.S. National Library of Medicine, 2013) est utilisé pour PubMed (U.S. National Library of Medicine, 2015) pour l'indexation de publications médicales.
- SNOMED CT (International Health Terminology Standards Development Organisation, 2014) est la nomenclature la plus utilisée en milieu clinique.
- ICD-10 (World Health Organization, 2011) est une classification internationale et reconnue pour la classification des pathologies et des informations s'y rapportant.
- LOINC (McDonald, Huff, Deckard, Holck, & Vreeman, 2014) est une terminologie utilisée pour les informations relatives au DMP.
- UMLS-RxNorm (U.S. National Library of Medicine, 2014) est une nomenclature pour la gestion des médicaments et de leurs interactions.
- ICPM/ICHI (World Health Organization, 2015) est une méthode permettant la classification des actes médicaux.
- ICPC (WONCA International Classification Committee, 2005) est une classification internationale des soins de la médecine générale.
- CPT-4 (American Physical Therapy Association, 2014) est un codage employé pour la facturation des actes médicaux.

Un système informatique peut d'ailleurs en utiliser plusieurs, en fonction de ses besoins et des domaines de nommage proposés par chacune d'entre elles (pour un exemple, consulter (Wright & Sittig, 2008)). Il existe évidemment des mécanismes pour passer d'une terminologie à une autre d'un même domaine fonctionnel (comme, par exemple, entre SNOMED CT et MeSH (Jacobs, Quinn, & Nelson, 2006)). Il n'est cependant ni facile, ni garanti, de pouvoir ainsi exprimer tous les concepts et détails d'une terminologie dans une autre. Certains outils, tels que Apelon's TermWorks (Apelon Inc., 2014), permettent de procéder à une mise en correspondance entre les termes issus de multiples terminologies. Au sein d'un SMA, un médiateur/facilitateur (**Figure 143**) peut d'ailleurs faciliter le passage d'une terminologie à l'autre (par exemple, consulter (Andonoff, Bouaziz, Hanachi, & Bouzguenda, 2009)). De telles ontologies sont particulièrement précieuses lorsque l'on souhaite décrire des workflows indépendants de tout SI (Boxwala et al., 2001). En effet, elles vont permettre de faire le lien entre les notions manipulées par les workflows et leurs pendants au sein des SI et outils existants (Madni & Sievers, 2014). Ainsi, les ontologies permettent aux agents de disposer d'un référentiel sémantique commun lors de leurs coopérations à visée applicative (Huhns & Singh, 1998). Ceci nous conduit d'ailleurs logiquement à nous intéresser, plus en détail, à l'intégration d'applications.

B) L'intégration d'applications

Le second type d'intégration qu'il sera impératif de maîtriser est l'intégration d'applications, c'est-à-dire la nécessité de pouvoir intégrer notre système avec d'autres systèmes existants et indépendants (Genesereth & Ketchpel, 1994; Genesereth, 1997; Madni & Sievers, 2014; Maier, 1998). Par « applications », nous désignons les différentes technologies assurant techniquement la réalisation d'au moins une fonction impliquée dans un processus de travail. Dans le cas de notre PAMMA, ces applications prennent la forme de composants applicatifs (par exemple, OpenCV (Intel & Willow Garage, 2013) utilisé pour le traitement

d'images) ou de programmes/services autonomes (par exemple, Unity (Unity Technologies, 2010) utilisés pour créer des environnements graphiques dans lesquels l'utilisateur peut interagir, ou encore une base de données telle que MySQL (Oracle Corporation, 2015)). Ce genre d'intégration est assuré par la mise en place de frameworks proposant des connecteurs, des adaptateurs et fixant des règles de communication avec les technologies applicatives à intégrer (Xu, Sauquet, Degoulet, & Jaulent, 2003). Il faut également pouvoir synchroniser l'exécution de ces différentes technologies (Chan, Zhang, & Li, 2003). En l'occurrence, c'est exactement ce que nous voulons faire avec notre PAMMA.

Les agents ont la capacité d'encapsuler des technologies existantes (Nahm & Ishikawa, 2005; Peng et al., 1998), par exemple, par le biais de Web services (Purvis, Savarimuthu, & Purvis, 2004). Ils permettent ainsi, au travers de la constitution de réseaux d'agents, l'intégration d'organisations disposant déjà d'un certain nombre d'outils auxquels elles n'auront pas à renoncer (Huhns & Singh, 1998; Kishore et al., 2006). Dans le contexte médical, cette faculté est indispensable pour obtenir l'assentiment des praticiens en ce qui concerne l'opportunité d'introduction d'une nouvelle technologie (Cortés, Annicchiarico, & Urdiales, 2008; Nealon & Moreno, 2002, 2003). En effet, il n'est pas réaliste d'imaginer qu'une organisation déjà en place va intégralement remplacer son infrastructure d'information par la nôtre ou modifier ses pratiques pour s'adapter à notre système (Protti, 2009). De plus, les systèmes existants ainsi que les pratiques en vigueur vont à la fois contraindre et faciliter notre action (Hanseth & Lundberg, 2001; Hanseth & Lyytinen, 2004). L'intégration entre différentes technologies n'est, cependant, pas une chose triviale et mérite donc d'être prise en considération le plus tôt possible dans le développement (Ellingsen & Røed, 2010) afin de se conformer aux standards d'échanges de données (Kalra, 2006)(voir la section **L'intégration des données**). Nous nous proposons d'énumérer les différents types de stratégies architecturales qui permettent une telle intégration (Dayal et al., 2001; Johannesson & Perjons, 2000; Wangler et al., 2003) :

- La stratégie point-à-point, qui consiste à connecter les différents systèmes à intégrer deux à deux. Le nombre d'interfaces à implémenter explose donc exponentiellement alors que le nombre de systèmes différents augmente.
- L'architecture organisée autour d'un broker, consistant à introduire un médiateur/facilitateur entre les systèmes. Ainsi, chaque système doit juste se conformer à un format de messages ou d'objets tel que défini par le broker, et n'envoie des informations, ou n'en reçoit, qu'à travers ce dernier.
- Le gestionnaire de workflows est, quant à lui, un broker amélioré qui centralise toute la logique, toute la sémantique, qui est derrière les communications entre différents systèmes. Ce gestionnaire, qui s'appuie sur des modèles, va donc pouvoir piloter/automatiser les processus de travail, impliquant différents systèmes, et offrir la possibilité de les modifier à volonté simplement et rapidement. L'analyse sur le long terme, que permet l'utilisation d'une pareille technologie, facilite la validation et l'amélioration des processus métiers (National Transitions of Care Coalition, 2010).

L'intégration entre différents systèmes peut se faire à l'aide de brokers de messages ou/et d'objets (comme CORBA (Object Management Group, 2014d) qui est, par exemple, utilisé dans GEODE-CM (Greenes et al., 1999; Stoufflet et al., 1996) ainsi que dans une plate-forme d'intégration MA (Chan et al., 2003)), de portails en ligne (comme DDS (Object Management Group, 2014b)) ou encore de Web services (par exemple, transmettant des fichiers au format XML (World Wide Web Consortium, 2013) comme dans (Wright & Sittig, 2008; Yan,

Pidgeon, Krishna, & Yong, 2007; Yoo et al., 2008)). D'autre part, il existe un certain nombre de normes de sécurité et d'interopérabilité permettant de guider cette intégration. Dans le contexte médical, nous avons décidé d'évoquer les suivantes :

- ASTM F2761-09 (American Society for Testing and Materials, 2013), qui décrit l'architecture ICE (composée d'un superviseur, d'un contrôleur réseau et d'interfaces pour chaque matériel) utilisée pour les PAM (Hatcliff et al., 2012; Larson et al., 2012).
- ISO/IEEE 11073 (International Organization for Standardization, 2015b), qui présente des standards de communication pour les appareils médicaux.
- ISO 14971 (International Organization for Standardization, 2013), qui caractérise les mesures de gestion des risques des appareils médicaux afin de garantir la sécurité des patients.
- IEC 60601 (International Organization for Standardization, 2015a), qui est une collection de standards pour permettre l'efficacité et la sûreté des appareils médicaux.

La prise en compte de ces différentes normes est indispensable pour la mise sur le marché d'un quelconque produit. En effet, elles sont garantes du respect des préconisations communément admises par les experts techniques du domaine. Finalement, on pourra remarquer que ces différents moyens d'intégration sont autant de frameworks (Fayad & Schmidt, 1997). Le niveau le plus flexible de l'intégration d'applications est celui qui repose sur un gestionnaire de workflows. Une telle intégration revient à intégrer explicitement les processus de travail des utilisateurs du système.

C) L'intégration des processus métiers

L'intégration des processus métiers repose sur la capacité à modéliser ces processus et à incorporer au système les modèles ainsi générés. Ce niveau d'intégration fournit un haut niveau d'abstraction et d'adaptabilité qui permet la définition, le contrôle et la modification des processus de travail au sein d'une organisation (Morley et al., 2011). Ces processus permettent alors la conception formelle d'une organisation, c'est-à-dire la définition d'une structure organisationnelle définie précisément et reliée à un ensemble d'objectifs, de règles, d'outils et de technologies régis ensemble par les contraintes organisationnelles (Kirn & Gasser, 1998). Ainsi, un tel système permet, entre autres, d'explicitier les interdépendances entre les activités des processus de travail, d'automatiser ces processus et d'assurer la coordination et la collaboration, au travers de partages d'informations/données, entre les différentes ressources potentiellement distribuées du système (Kishore et al., 2006; Purvis, Purvis, Haidar, & Savarimuthu, 2005; Purvis et al., 2004). Il est, à ce titre, commode de constater que le formalisme MA s'avère adapté pour ce genre d'intégration dans la mesure où les agents peuvent représenter dynamiquement les acteurs du système, qui sont autonomes et qui poursuivent leurs objectifs propres, et peuvent contrôler le comportement des données par la mise en œuvre des règles de gestion (Griss, Chen, Bolcer, Kessler, & Osterweil, 1999; Guo, 2007; Hawryszkiewicz & Debenham, 1998; Jennings, 1996; Sánchez et al., 2010; Taveter & Tamm, 2001; Yu & Schmid, 1999)(pour des exemples d'applications de gestion de workflows à l'aide d'agents, consulter (Hsieh & Lin, 2014; Savarimuthu & Purvis, 2004; Zeng, Benatallah, Nguyen, & Ngu, 2001)). En particulier, la gestion de workflows à

l'aide de SMA implique une gestion décentralisée et cohérente avec les structures organisationnelles distribuées que nous voulons modéliser (Dayal et al., 2001).

D'autre part, dans le domaine médical, il est primordial de tenir compte des directives/recommandations médicales qui se présentent comme garantes de la mise en œuvre de bonnes pratiques (Moser et al., 2001; Wang, Peleg, Tu, Shortliffe, & Greenes, 2001). Dans le but de spécifier de telles directives/recommandations cliniques, il est nécessaire de procéder à une décomposition cartésienne des problèmes à résoudre (de Clercq, Hasman, Blom, & Korsten, 2001; Tu, Eriksson, Gennari, Shahar, & Musen, 1995) pour lequel l'utilisation de SMA est, par nature, appropriée. Historiquement ces directives/recommandations cliniques se présentent sous la forme d'énoncés élaborés pour assister les praticiens dans leurs prises de décisions en ce qui concerne les dispositions appropriées à des circonstances cliniques précises (Peleg & González-Ferrer, 2014). A ce titre, un modèle visant à les décrire doit permettre (de Clercq et al., 2004) :

- De comprendre en profondeur les procédures cliniques visées.
- De les décrire de manière précise et non-ambigüe.
- D'offrir un moyen, facilitant l'aide à la décision, pour les analyser et les effectuer automatiquement.

Malheureusement, ces énoncés sont fréquemment décrits d'une manière textuelle et narrative assez peu formelle et, de ce fait, ne sont pas directement exécutables par un ordinateur (Lobach & Kerner, 2000; Peleg & González-Ferrer, 2014; Votruba, Miksch, & Kosara, 2004). De telles représentations n'influencent donc que modérément les pratiques médicales (Boxwala et al., 2001), contrairement à celles implémentées informatiquement qui normalisent positivement les pratiques cliniques (Isern, 2008; Lobach & Hammond, 1997; Ohno-Machado et al., 1998). Pour en réaliser des versions exécutables, il est possible de disposer d'une représentation formelle apte à la description des pratiques médicales sous la forme de workflows. En effet, les objectifs et exigences des directives/recommandations cliniques sont naturellement similaires à ceux d'autres processus métiers. On remarquera cependant que leur passage d'une représentation textuelle, en langage naturel, à une représentation formelle, est une activité critique pour la réalisation d'une application répondant réellement aux besoins des utilisateurs (Yu, 1995). A ce titre, et afin de capturer les différents aspects précités, nous proposons de découper l'implémentation des workflows en milieu clinique en trois activités et traitements (**Figure 156**) :

- Spécifier, qui consiste à produire une représentation graphique des workflows afin d'être compréhensible par des humains et plus facilement adoptée.
- Encoder, c'est-à-dire que l'on produit des versions informatiques et exécutables des workflows à partir de leurs représentations graphiques.
- Exécuter, qui est l'étape qui correspond à l'adoption des workflows en condition réelle tels qu'ils sont décrits au travers de leurs versions encodés.

La frontière entre ces deux premières activités est assez floue, car elles consistent toutes les deux à définir des représentations formelles ou semi-formelles des workflows (Terenziani, Molino, & Torchio, 2001). Dans le cas où serait utilisé un formalisme sans représentation graphique, la spécification et l'encodage sont confondus. Dans celui où les workflows sont spécifiés graphiquement, l'utilisateur ne peut influencer l'encodage qu'en sélectionnant le formalisme utilisé par celui-ci. La **Figure 156** représente ce processus d'implémentation (pour une description plus détaillée, consulter (Gooch & Roudsari, 2011)). Pour respecter les

contraintes d'interopérabilité, dont nous avons déjà plusieurs fois traité, chacune de ces étapes doit reposer sur l'utilisation d'une norme, d'un standard. Ainsi, il devient possible d'obtenir des représentations formelles, ou semi-formelles, de connaissances médicales qui peuvent être aisément disséminées au travers de workflows capturant une certaine expertise clinique (Fox, Johns, & Rahmanzadeh, 1998). L'exécution de ces workflows va de cette manière permettre d'apporter une aide à la décision aux praticiens au cours de leur exercice clinique quotidien (Terenziani, Montani, Bottrighi, Molino, & Torchio, 2008).

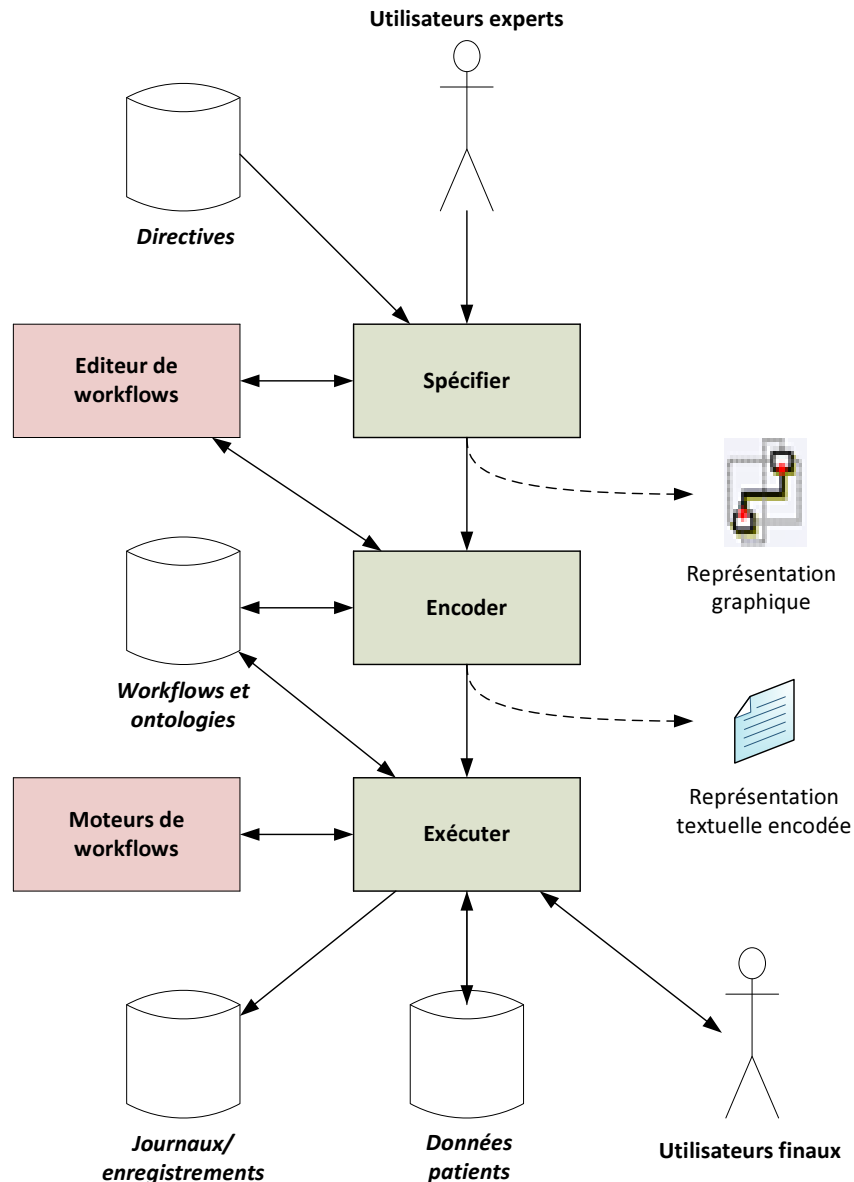


Figure 156 : Représentation d'un modèle d'implémentation de workflows en milieu clinique

Dans notre contexte applicatif, les agents vont à la fois être utilisés pour décrire les ressources du système mais également pour assurer l'infrastructure associée à l'exécution des workflows (Griss et al., 1999; Kishore et al., 2006; Leymann & Roller, 1997)(comme, par exemple, cela est effectué dans (Alsinet et al., 2000; Chen et al., 1999)). Cette infrastructure va, en particulier, gérer l'entrée/apparition et la sortie/disparition d'agents dans le système, et garantir que chacun d'entre eux respecte des règles communes et indépendantes de leurs implémentations respectives (Esteva, Rosell, Rodriguez-Aguilar, & Arcos, 2004).

Les outils que nous souhaitons proposer doivent constituer le véritable support d'une méthode d'ingénierie pour la réalisation de SMA conformes à notre framework (voir la sous-section **La perspective du méta-système**). Ces SMA sont simplement les applications destinées à fonctionner sur la PAMMA (pour des exemples de telles applications ainsi que des développements concernant la méthode d'ingénierie, voir la partie **L'analyse des résultats et les perspectives**). Nous allons maintenant détailler les outils de la PAMMA permettant d'accomplir les points clef de la méthode proposée sur la **Figure 156**.

1) La spécification

La spécification va être réalisée grâce à l'outil que nous nommerons simplement « l'éditeur ». Cet éditeur est un outil de génie logiciel assisté par ordinateur : son but est de proposer un environnement de développement rapide d'applications au travers d'une conception principalement graphique (d'une manière proche, par exemple, de l'outil présenté dans (Adam, 2000)). Les grandes lignes de son utilisation ont été présentées sur la **Figure 68** et la **Figure 69**. En particulier, ce type d'outil permet à son utilisateur (Nwana et al., 1999) :

- De se concentrer sur l'implémentation de la résolution de problèmes du domaine.
- De bénéficier d'une interface graphique de manipulation directe.
- D'étendre facilement les fonctionnalités du système.
- De se reposer sur des composants standardisés favorisant l'adoption du système.

Par ailleurs, dans le contexte des applications médicales, il est montré que l'utilisation de représentations graphiques facilite la compréhension de données complexes (Aigner, Kaiser, & Miksch, 2008; Miksch, Kosara, Shahar, & Johnson, 1998; Müller et al., 2011, 2010). Ainsi, nous voulons pouvoir proposer un environnement de conception dans lequel l'utilisateur pourra décrire le fonctionnement de systèmes sous des formes similaires à des schémas entité-association (Chen, 2002)(la **Figure 160** est un exemple de schéma entité-association) et des ordinogrammes (Ecma International, 1966; IBM, 1970)(la **Figure 166** est un exemple d'ordinogramme) afin de, respectivement, décrire la structure et la dynamique du système. Un avantage de l'utilisation des ordinogrammes est qu'ils sont bien connus des médecins (Hadorn, 1995; Society for Medical Decision Making Committee on Standardization of Clinical Algorithms, 1992) et qu'ils ne requièrent qu'un apprentissage minimal. L'éditeur va permettre de faciliter la programmation en proposant des mécanismes d'assistance au développement et des éléments réutilisables (Klinker et al., 1991). Il se place alors comme étant l'outil qui assure une utilisation correcte et efficace de notre framework MA (voir la sous-section **La perspective du méta-système**).

Comme nous l'avons précédemment indiqué, le point de départ du développement d'une application sur notre PAMMA est une directive/recommandation clinique sous une forme narrative. Cette dernière va être convertie en un algorithme représenté graphiquement (pour un exemple de telle conversion, consulter (Lobach & Kerner, 2000)) à l'aide de l'interface graphique de l'éditeur et va prendre la forme d'un workflow (Isern, 2008). Ce dernier va combiner et paramétrer différentes applications ou composants applicatifs, c'est-à-dire les intégrer (comme, par exemple, dans (Huser et al., 2011)). D'autre part, on notera qu'une telle représentation graphique va faciliter la coopération entre praticiens experts et analystes informatiques (Huser, Rocha, & James, 2006). La modélisation des processus doit

d'ailleurs se faire au travers de quatre activités successives (Kirn & Gasser, 1998; Nurcan, 1996) :

- La spécification et décomposition du travail en tâches élémentaires, c'est-à-dire en tâches qui peuvent être effectuées à l'aide d'une seule fonction.
- La sélection et l'intégration des tâches en groupes cohérents exhibant des traits similaires comme leur domaine fonctionnel ou les compétences qu'elles requièrent.
- La sélection et le regroupement des postes, c'est-à-dire la définition de groupes de rôles basés sur leurs interdépendances à la fois hiérarchiques et en termes de compétences nécessaires à l'accomplissement de tâches.
- Le raccordement des acteurs à des postes, c'est-à-dire la désignation d'individus à des postes correspondant à leurs compétences.

C'est pourquoi notre éditeur devra proposer des écrans et des fonctionnalités adaptées à chacune de ces activités et chaque type d'utilisateurs. Il est notable que cette capacité de notre outil ouvre également la voie à l'amélioration et l'optimisation de ces processus (Christov et al., 2008; Ehrler, Fleurke, Purvis, & Savarimuthu, 2005; Morley et al., 2011; Yu, 1995), voire à la proposition de standards (Framinan et al., 2005; Greenes et al., 1999).

L'éditeur doit disposer de trois niveaux permettant de décrire les éléments appartenant aux différentes couches fonctionnelles du méta-modèle que nous avons précédemment défini (voir la sous-section **La perspective du méta-système**). Cet éditeur facilite ainsi le développement d'applications conformes au framework MDA (Object Management Group, 2014c) (comme d'autres méthodes de développement de SMA le préconisent, telles que, par exemple, ASEME (Spanoudakis & Moraitis, 2008)), ce qui augmente leur portabilité et leur interopérabilité. Le **Tableau 27** résume la relation entre les couches fonctionnelles du méta-modèle, les niveaux de l'éditeur et les modèles du framework MDA.

Couches fonctionnelles du méta-modèle	Niveaux de l'éditeur	Modèles MDA
Les couches du domaine métier	Niveau 1	<i>Platform-Specific Model</i>
Les couches de la section métier	Niveau 2	<i>Platform-Independent Model</i>
Les couches applicatives	Niveau 3	<i>Computation-Independent Model</i>

Tableau 27 : Relation entre les couches fonctionnelles du méta-modèle, les niveaux de l'éditeur et les modèles du framework MDA

Dans le reste de cette section, nous allons expliquer plus en détail le fonctionnement de ces niveaux de l'éditeur, et donner des exemples concrets de formalismes qu'il pourrait être intéressant que chaque niveau supporte. On pourra d'ailleurs préciser que chacun des trois niveaux constitue un véritable framework. Nous évoquerons également l'intérêt qu'ont ces niveaux à se conformer aux modèles PSM, PIM et CIM du framework MDA.

a) *Le niveau 1 de l'éditeur*

Le niveau 1 de l'éditeur sert à implémenter les éléments des couches du domaine métier, c'est-à-dire les agents, ainsi que les éléments permettant de les constituer, mais également

les protocoles d'interaction décrivant leur collaboration. Dans le cas des agents, cette implémentation va réellement correspondre au codage de leurs composants (par exemple, leurs comportements internes ou encore leurs modules syntaxiques (voir la sous-section **Les systèmes multi-agents**) à l'aide de langages qui manipulent les concepts d'objets et/ou d'agents. Ce niveau est ainsi dépendant de langages de programmation (comme c'est, par exemple, également le cas pour les agents PAC (Nigay & Coutaz, 1995) ou ceux du framework SARL (Galland et al., 2014; Rodriguez, Gaud, & Galland, 2014)) et de plates-formes d'exécution spécifiques, c'est-à-dire de choix technologiques, ce qui coïncide bien au PSM du framework MDA (Object Management Group, 2014c). On remarquera que le choix des langages de programmation et celui des plates-formes d'exécution sont liés, cependant n'importe quel langage (ou plate-forme) peut potentiellement faire l'affaire. Ainsi, il nous paraît inutile d'énumérer ou de détailler les langages et/ou plates-formes qu'il sera possible d'utiliser pour l'implémentation des agents. La seule contrainte est qu'ils se conforment à notre framework (voir la sous-section **La perspective du méta-système**), tel que basiquement représenté sur la **Figure 157** (qui reprend le code couleur utilisé sur la **Figure 22**), afin de proposer une implémentation sous inversion de contrôle.

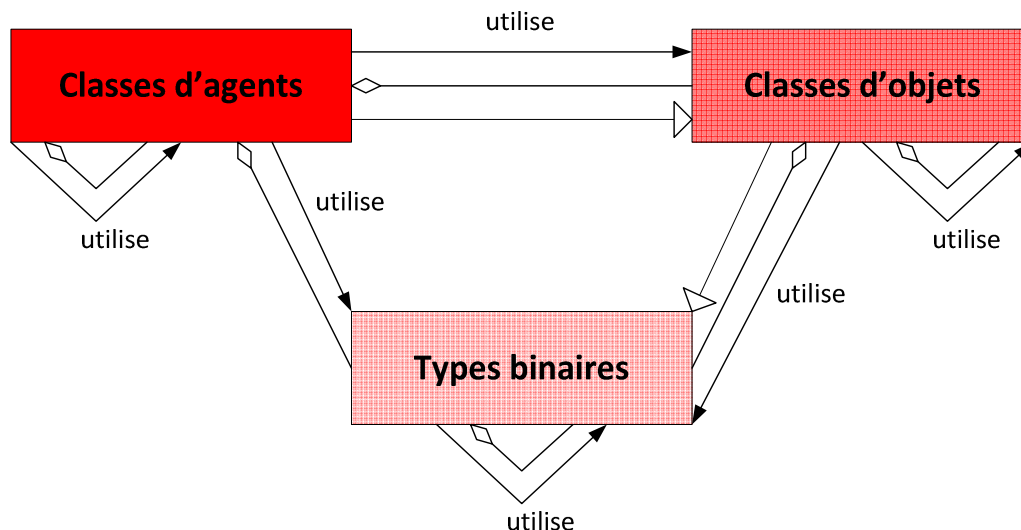


Figure 157 : Diagramme des types de classes des couches du domaine métier du modèle MVC composé d'agents

Cette relation entre les différents types de classes de notre framework vaut d'ailleurs pour trois parties du modèle MVC. L'utilisateur de l'éditeur utilisant le niveau 1 va donc pouvoir intervenir sur le Modèle, la Vue ou le Contrôleur. En ce qui concerne la partie de l'éditeur dédiée à la conception d'interface utilisateur, il est important de signaler qu'un tel outil de conception d'interfaces doit à la fois être accessible/compréhensible et offrir beaucoup de possibilités différentes (Myers et al., 2000). A ce titre, les modalités d'entrées peuvent être, par exemple, définies à l'aide d'une approche similaire à celle d'ICARE (Bouchet, 2006). Cette approche peut d'ailleurs également être appliquée aux modalités de sorties (Mansoux et al., 2006), mais elle relève du niveau 2 de notre éditeur, c'est-à-dire de la définition d'organisation/groupes d'agents. Au niveau 1, il est au mieux possible de proposer un composant WYSIWYG de l'éditeur qui facilite la définition rapide, sous la forme d'holarchies d'agents, des parties de l'interface directement perceptibles par l'utilisateur final. Finalement, nous rappelons que la PAMMA propose différents services qui sont utilisables

par les agents tels que les pages jaunes (comme, par exemple, dans (Kirn et al., 2003))(Figure 142) et les pages blanches (Figure 141). Ces services sont fournis par des gestionnaires, des « managers », et vont pouvoir être appelés explicitement à l'implémentation des agents.

Dans le cas des protocoles d'interaction, l'implémentation va consister à les décrire graphiquement à l'aide d'ordinogrammes. Pour chaque protocole, il faut préciser :

- Le nombre d'agents impliqués.
- La liste des rôles que ces agents peuvent prendre.
- L'agent qui initie l'interaction.

Nous rappelons qu'avoir recours à des protocoles entre agents permet, à l'exécution, la coordination dynamique des comportements coopératifs de ces derniers. Ces comportements permettent la réalisation de leurs objectifs individuels et de celui global du système (Frank, 2001; Nwana et al., 1996). Pour garantir la qualité de ces protocoles, leur conception devra également reposer sur une méthode formelle ou semi-formelle (comme, par exemple, celle proposée dans (Huget & Koning, 2003)). En plus des langages d'interaction entre agents, qui permettent également la description de protocoles (voir la sous-sous-section **La communication entre agents**), nous indiquons brièvement l'existence des langages suivants, qui permettent spécifiquement la description de protocoles :

- AgentTalk (Kuwabara, Ishida, & Osato, 1995a, 1995b).
- COOL (Barbuceanu & Fox, 1995; Barbuceanu & Lo, 1999).
- CPDL/GrCPDL/UAMLe (Koning, Huget, Wei, & Wang, 2002; Koning & Huget, 2001) pour lequel il existe un outil de conception graphique nommé DIP (Huget, 2001).
- LCC (Robertson, 2004, 2005, 2012).
- PDL (Goldblatt, 1992) ou ses extensions (Paurobally & Cunningham, 2003).
- Les réseaux de Petri colorés (Cheung, 1996; Coriat, 1993)(pour un exemple d'utilisation, consulter (Purvis et al., 2005)).
- SDL (Fonseca i Casas, 2014; Lai & Jirachiefpattana, 1998; Sandhu, 1992).
- UML et ses extensions (Bauer & Odell, 2005; Koning et al., 2002; Lind, 2002).

La plupart de ces langages dispose d'une représentation graphique et d'une représentation textuelle formelle ou semi-formelle. Le formatage de cette représentation textuelle peut d'ailleurs correspondre au processus d'encodage (voir la sous-section **L'encodage**). Nous remarquerons, d'autre part, que le point de vue adopté par ces différents langages peut être de décrire soit les états de chaque agent au cours de la conversation soit les états mêmes de la conversation, qui peuvent aussi être appelés « scènes » (Esteva et al., 2004).

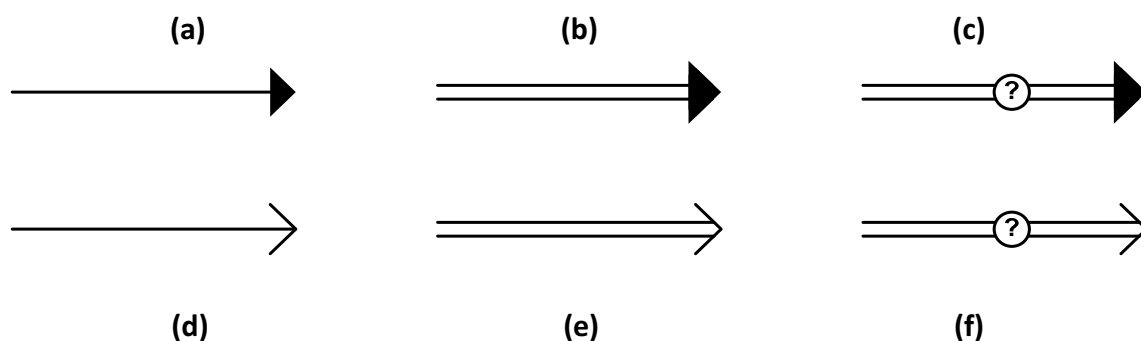


Figure 158 : Notation graphique des différents types d'envoi de messages

Par ailleurs, et afin de tenir compte des capacités de la plate-forme en termes de gestion des messages (voir la sous-sous-section **La communication entre agents**), il peut être judicieux d'introduire des notations graphiques pour :

- Représenter les messages synchrones (**Figure 158 (a)**) et asynchrones (**Figure 158 (d)**). Contrairement à ceux asynchrones, les messages synchrones provoquent chez l'expéditeur un état bloquant d'attente d'une réponse de la part du destinataire.
- Préciser quels messages passent par un médiateur/facilitateur (**Figure 158 (b)** et **(e)**).
- Signaler qu'un message n'est adressé à aucun agent en particulier (**Figure 158 (c)** et **(f)**). Un tel message est cependant destiné à tous les agents remplissant un ensemble de critères, identifié par un rôle, et est délivré grâce à un médiateur/facilitateur.

Un protocole doit également avoir un début et une fin. La **Figure 159** illustre cela par le biais d'un diagramme d'états-transitions en UML.

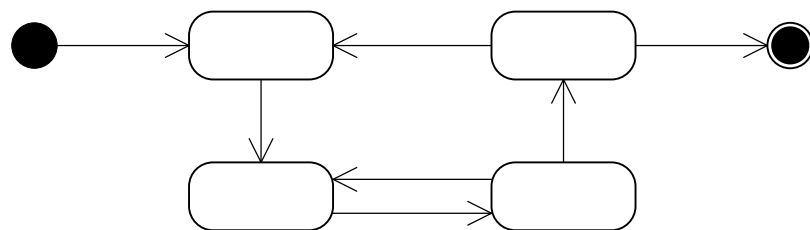


Figure 159 : Exemple de protocole d'interaction

On signalera également qu'afin de garantir la réutilisabilité de toute ou partie des protocoles définis dans le cadre de notre système, les protocoles peuvent être composés d'un ou plusieurs sous-protocoles (Huget, 2001). Pour permettre encore plus de réutilisabilité des protocoles d'interaction, il doit également être possible d'établir une sorte de filiation des protocoles à l'aide d'une propriété d'héritage (comme, par exemple, dans (Kuwabara et al., 1995a)). Le niveau 2 de l'éditeur va nous permettre d'utiliser les agents et protocoles d'interaction définis au niveau 1 afin de constituer des réseaux/organisations d'agents.

b) Le niveau 2 de l'éditeur

Le niveau 2 de l'éditeur sert à concevoir les réseaux/organisations d'agents, qui peuvent être aussi assimilés à des agents (au sens holarchique que l'on retrouve dans (Ferber et al., 2004; Giret & Botti, 2004; Rodriguez et al., 2014)). Pour cela, nous préconisons l'utilisation d'une représentation graphique à l'aide de schémas d'entité-association (éventuellement sur plusieurs niveaux comme une carte conceptuelle)(comme, par exemple, c'est le cas dans le framework *i** (Yu, 1995) qui est utilisé dans (Faulkner et al., 2004)). Ainsi, il est possible de représenter les agents et les relations qu'ils entretiennent, c'est-à-dire de représenter la structure de SMA, par des schémas entité-association (Chen, 2002) où les entités sont les agents et les associations sont les protocoles. Sur les arêtes de ces schémas apparaissent les rôles et cardinalités d'agents impliqués dans un protocole.

Ces relations entre agents, supportées par des protocoles du niveau 1 de l'éditeur, expriment des échanges de flux de données ou de contrôle. Les agents ont également été définis au niveau 1, l'utilisation d'une technologie ou d'une plate-forme particulière est donc

transparente à ce niveau. Ce niveau est uniquement dépendant des techniques retenues mais d'aucune technologie : il est donc effectivement conforme au PIM du framework MDA (Object Management Group, 2014c). Ce niveau rend ainsi possible la transformation des implémentations d'agents, des descriptions de protocoles d'interaction et des spécifications de leur organisation en véritable SMA (Wooldridge, 1997)(comme, par exemple, cela se fait avec MAS-ML (da Silva et al., 2004)). La **Figure 160** présente un exemple d'organisation définie à ce niveau 2 de l'éditeur.

Dans l'exemple, on distingue deux groupes d'agents : le groupe A et le groupe B. Ces agents interagissent au travers de deux protocoles distincts :

- Le protocole 1 qui prend place entre un agent du groupe A, jouant le rôle a, et un du groupe B, jouant le rôle b.
- Le protocole 2 qui prend place entre l'agent A2, jouant le rôle c, et un agent du groupe B, jouant le rôle d.

Cet exemple montre qu'un protocole peut être destiné à un agent spécifique ou bien à un ensemble d'agents remplissant un ensemble de critères. Un tel type de représentation ne représente pas la dynamique de l'organisation d'agents mais seulement sa structure. D'autre part, on notera, pour faciliter la conception de ces SMA, qu'il est possible, voir préférable, d'utiliser les patterns méso-architecturaux de notre framework MA (voir la sous-section **La perspective du méta-système**).

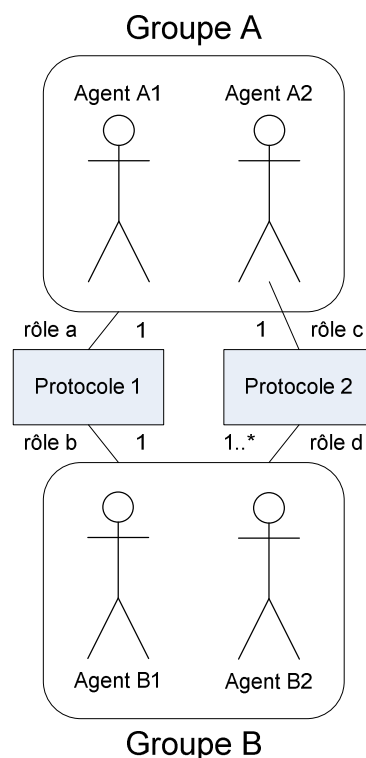


Figure 160 : Exemple d'une organisation d'agents

Il est également important de préciser que ces réseaux/organisations d'agents ne sont pas nécessairement composés uniquement d'agents locaux ni exclusivement d'agents disposant de leur propre thread de contrôle. Notre système doit ainsi offrir la possibilité d'intégrer aux SMA des agents locaux et/ou distants, qu'ils soient threadés ou non. A ce titre, nous souhaitons proposer une notation graphique apportant ces précisions à la représentation

d'un agent au niveau 2 de l'éditeur (**Figure 161 (a)**) tenant compte des propriétés de son implémentation au niveau 1. En effet, pour chaque agent, il nous faut pouvoir préciser :

- S'il dispose de son propre thread de contrôle, c'est-à-dire s'il est autonome du point de vue de son exécution (ce qui est obligatoire pour les agents proactifs et pour l'agent de plus haut niveau d'une holararchie de la Vue)(**Figure 161 (b)**).
- S'il exhibe un service qui doit apparaître dans les pages jaunes (**Figure 161 (c)**). L'implémentation de cet agent au niveau 1 de l'éditeur doit préciser la portée de visibilité du service (par exemple : « locale », « PAMMA voisines uniquement » ou encore « tout le réseau de PAMMA »).
- S'il s'agit d'un agent/service qui existe quelque part dans le réseau de PAMMA (qui peut donc être local ou distant) et que l'on souhaite utiliser (**Figure 161 (d)**).

Ces différentes notations ont pour but de faciliter le travail de conception au niveau 2 de l'éditeur. Le concepteur accède ainsi directement à des informations primordiales pour la construction d'un SMA sans avoir à se plonger dans l'implémentation des agents.

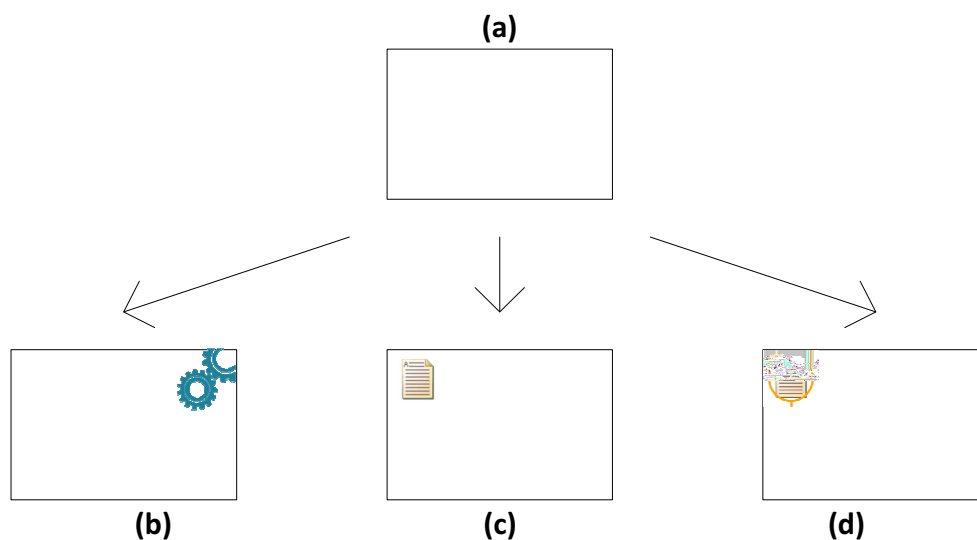


Figure 161 : Notations graphiques des agents au niveau 2 de l'éditeur

On notera que les notations des **Figure 161 (b)** et **Figure 161 (c)** peuvent être combinées afin de représenter un agent threadé qui propose au moins un service. De plus, les agents ne disposant pas de leur propre thread doivent être rattachés à un autre agent via un lien de filiation (traduisant également une relation holararchique entre ces agents)(**Figure 162**). Ce thread peut alors être successivement utilisé par tous les agents ayant des liens de filiation directe ou indirecte et qui forment une généalogie. Il existe des restrictions à ce type de généalogie :

- Tous les agents d'une même généalogie doivent être locaux.
- Au moins l'agent au sommet de cette généalogie doit posséder son propre thread de contrôle.

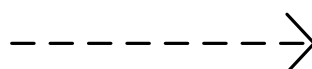


Figure 162 : Notation graphique d'un lien de filiation entre agents

Pour finir, nous signalons que l'utilisation d'agents pour la réalisation de workflows permet de donner une plus grande robustesse, flexibilité et évolutivité à ces derniers (Purvis et al., 2004; Yu & Schmid, 1999). Effectivement, au travers d'organisations d'agents, il devient possible de prendre en compte des activités qui ne sont pas entièrement spécifiées a priori comme cela est fréquemment le cas au sein des organisations réelles que l'on tente d'intégrer (Hawryszkiewicz & Debenham, 1998). Le niveau 3 de l'éditeur va nous permettre d'utiliser les SMA définis au niveau 2 à des fins d'intégration explicite et dynamique des processus de travail.

c) Le niveau 3 de l'éditeur

Le niveau 3 a pour but la modélisation des processus métiers et permet donc de représenter le fonctionnement de l'organisation humaine que doit intégrer le système informatique. A ce titre, il permet de représenter les acteurs, les activités et les artefacts impliqués dans cette organisation (Adam, 2000). Pour cela, il serait confortable de permettre à notre éditeur d'utiliser un langage graphique de description de processus. En effet, il est nécessaire de disposer d'un langage formel, ou semi-formel, pour décrire les processus cliniques et d'un outil de développement pour les spécifier (Fox et al., 2003)(pour un exemple de mise en pratique, consulter (Christov et al., 2008)). Il existe de nombreux formalismes et techniques de modélisation permettant la représentation de processus métiers en fonction des usages (Giaglis, 2001; Hollingsworth, 2004; Isern, 2008; Roser & Bauer, 2005). Dans ce contexte d'intégration de processus cliniques, nous signalons également qu'il existe de nombreux systèmes, qui disposent de capacités et performances variables, exploitant ces différents langages (Isern & Moreno, 2008; Peleg & González-Ferrer, 2014; Shiffman, Liaw, Brandt, & Corb, 1999). Il n'est donc pas trivial d'en choisir un sans connaître exactement le contexte de l'intégration qu'il doit réaliser. Dans tous les cas, ces langages de workflows permettent la spécification de vues fonctionnelles du système qui ne tiennent pas compte de choix techniques ou technologiques. Le niveau 3 est donc conforme au CIM au framework MDA (Object Management Group, 2014c). Les workflows ainsi définis sont a fortiori hautement portables/partageables.

Comme il est important d'avoir recours à des standards de données préétablis dans le but de bénéficier d'une forte interopérabilité, il est aussi important d'utiliser des formalismes reconnus pour décrire les workflows dans des formats partageables. De telles descriptions facilitent la communication entre les différents acteurs participant à la réalisation des applications (Fonseca i Casas, 2014). Il est important que les langages de description des processus (graphique ou non) soient des standards afin de permettre l'interopérabilité des processus décrits avec d'autres systèmes. La couche de code est d'ailleurs cachée derrière la représentation graphique, sous forme de diagramme, et fait appel à des applications modulaires et externes au gestionnaire de processus métiers (Huser et al., 2011). Il existe trois grands types de frameworks pour décrire les processus métiers (Morley et al., 2011). Ils diffèrent selon qu'ils adoptent des représentations orientées activité, orientées flux ou orientées états. Pour représenter les processus métiers sous forme de workflows, il faut avoir recours à des langages de modélisation adaptés, c'est pourquoi nous allons maintenant proposer un panorama, non-exhaustif, de langages de modélisation de workflows. Parmi ces langages, nous avons retenu des langages généralistes et d'autres plus spécifiques au

domaine médical (Isern & Moreno, 2008; Lenz et al., 2012). Il nous paraît ainsi pertinent de parler brièvement des formalismes suivants :

- Arden Syntax (Clayton, Pryor, Wigertz, & Hripcsak, 1989; Hripcsak et al., 1994; Hripcsak, 1994) est le plus ancien formalisme permettant la description de pratiques médicales sous la forme de modules, afin d'en permettre l'exécution informatisée et ainsi de les rendre reproductibles et partageables. Ce langage a cependant des possibilités assez limitées (Fox et al., 1997), comme celle de ne pas supporter la modélisation de données patients (Boxwala et al., 2001; Wang et al., 2001).
- ARIS (Davis & Brabänder, 2007; Davis, 2008), à l'aide du langage EPC, est une méthode d'analyse holistique des processus d'entreprise accompagnée de nombreux outils. Il est, d'autre part, maintenant possible de convertir les modèles EPC, qu'il produit, en modèles BPMN.
- Asbru (Hammermüller & Miksch, 1999; Miksch, Shahar, & Johnson, 1997; Shahar, Miksch, & Johnson, 1996, 1998) est une représentation de workflows médicaux axée sur la temporalité des processus qu'il est possible de subdiviser hiérarchiquement. Cette représentation repose sur un système de gestion de contraintes temporelles absolues et relatives ainsi que sur la poursuite de buts, plus que sur la réalisation de tâches particulières.
- BPMN (Object Management Group, 2014a; White, 2004) est une notation graphique inspirée d'UML. Elle a le mérite d'être assez polyvalente pour permettre la modélisation des processus métiers de tout type d'organisations d'une manière intuitive (Owen & Raj, 2003).
- DILEMMA/CPGM (Gordon, Herbert, & Johnson, 1996; Herbert, Gordon, Jackson-Smale, & Salis, 1995) repose sur des diagrammes d'états-transitions et un modèle générique de la prise en charge des patients. Ce formalisme s'avère insuffisant pour spécifier des workflows complexes (Johnson & Musen, 1997).
- EON (Musen, Tu, Das, & Shahar, 1996; Tu & Musen, 1996, 2000, 2001) est un système à visée médicale reposant sur une architecture basée sur des composants orientés objet et des ontologies (Tu et al., 1995). Ce formalisme permet la résolution de requêtes temporelles complexes à l'aide d'un module nommé Tzolkin (Nguyen, Shahar, Tu, Das, & Musen, 1997, 1999)(pour un exemple d'utilisation, consulter (Shahar et al., 1998)).
- GLARE (Bottrighi, Molino, Montani, Terenziani, & Torchio, 2013; Terenziani et al., 2001, 2008; Terenziani, Montani, Torchio, Molino, & Anselma, 2003) est un formalisme caractérisé par l'utilisation de techniques d'intelligence artificielle pour la définition et l'implémentation de directives/recommandations cliniques. Effectivement, les workflows y sont représentés sous la forme de chaînes de Markov.
- GLIF (Boxwala et al., 2004; Ohno-Machado et al., 1998; Wang & Shortliffe, 2002) est une modélisation orientée objet, dont la syntaxe est basée sur XML (World Wide Web Consortium, 2013), permettant la description de processus médicaux (pour un exemple d'utilisation, consulter (Gillois et al., 2001)). Ce formalisme a souffert de nombreux manques dans ses premières versions (de Clercq et al., 2004), mais depuis il a été consolidé à l'aide de la grammaire d'Arden Syntax (Peleg et al., 2001).
- Guide/NewGuide (Ciccarese et al., 2003, 2005) est un formalisme dans lequel les workflows médicaux sont décrits à l'aide d'ordinogrammes présentant des analogies avec les réseaux de Petri. NewGuide est simplement une amélioration/évolution de

Guide dans laquelle l'accent est mis sur l'interopérabilité avec les SI médicaux existants (Ciccarese et al., 2003; Ciccarese, Kumar, & Quaglini, 2002).

- IDEF0/IDEF3 (Dorador & Young, 2000; IEEE Computer Society, 1998; Mayer et al., 1995) sont des langages relativement contraints et qui ne permettent pas nécessairement de représenter un niveau de détail suffisant. Ils sont probablement plus adaptés à la représentation de processus d'unités de production qu'à celle de processus de soins (pour un exemple d'utilisation en milieu clinique, consulter (Staccini, Joubert, Quaranta, Fieschib, & Fieschib, 2000)).
- Little-JIL (LASER Process Working Group, 2006; Lerner, Osterweil, Sutton, & Wise, 1998) est un formalisme qui s'attache à représenter la coordination des activités et des agents y participant. Son utilisabilité en contexte clinique a d'ailleurs été montrée (Christov et al., 2008).
- Merise (pour un guide pratique, par exemple, consulter (Baptiste, 2012)), au travers de diagrammes de flux, est un formalisme fortement orienté vers la modélisation et le développement de solutions informatiques. Il semble ainsi un peu trop focalisé sur ce versant technique pour être parfaitement compris et maîtrisé par des représentants du milieu médical.
- OSSAD/CISAD (Dumas & Charbonnel, 1990; Nurcan, 1996) est une méthode d'analyse ainsi que de spécification de SI centrée sur l'organisation du travail. Les processus de travail peuvent y être représentés à l'aide de diagrammes relevant du modèle descriptif de la méthode OSSAD, pour lesquels CISAD propose des améliorations.
- PRODIGY (Johnson, Tu, Booth, Sugden, & Purves, 2000; Purves, Sugden, Booth, & Sowerby, 1999; Sugden, 1999) est une approche dans laquelle les processus médicaux sont mis à disposition sous la forme de services centralisés. Ainsi, différentes institutions peuvent procéder à l'exécution de ces processus, via un même portail (Clarity Informatics, 2014).
- PROforma (Fox et al., 1997, 1998, 1996), qui poursuit les efforts entrepris avec le langage R^2L (Das et al., 1997). Ce langage a l'avantage d'être constitué d'une représentation graphique, simple et non-ambigüe, et d'un encodage qui sont homomorphes et ouverts (Sutton & Fox, 2003).
- UML (Object Management Group, 2014e) permet de modéliser des processus, particulièrement au travers de ses diagrammes de cas d'utilisation, de séquences, d'activités et d'états-transitions. Ce langage est cependant très orienté vers le développement logiciel.
- WF-net (van der Aalst & van Hee, 2000; van der Aalst, 2000) est une extension des réseaux de Petri destinée à la modélisation de processus métiers (pour un exemple d'utilisation, consulter (Casalicchio & Tucci, 2013)). Ce formalisme, qui repose sur un solide modèle formel, supporte nativement les analyses mathématiques théoriques.
- WS-BPEL (Organization for the Advancement of Structured Information Standards, 2014), anciennement BPEL4WS, est un formalisme basé sur l'utilisation de web services pour spécifier le comportement des processus métiers. Cette approche permet de modéliser le comportement des participants des processus de travail en s'appuyant sur WSDL (World Wide Web Consortium, 2001)(pour des exemples d'utilisation, consulter (Guo, 2007; Yan et al., 2007)).
- XPDL (Workflow Management Coalition, 2012), anciennement WPD, est un standard pour la définition de workflows à l'aide de XML. Il permet de représenter tous les concepts de BPMN et les informations de la représentation graphique des workflows.

Chacun des formalismes précités bénéficie d'outils qui en facilitent l'utilisation dans le contexte de la définition de workflows. Le **Tableau 28** donne un bref aperçu de certains de ces éditeurs. On notera que certains d'entre eux permettent également l'exécution des workflows (par exemple, consulter (Ciccarese et al., 2005)).

Langages	Exemples d'éditeurs
Arden Syntax	MÉDAILLE (Bång & Eriksson, 1997) construit à l'aide de Protégé (Stanford Center for Biomedical Informatics Research, 2014)
ARIS	ARIS Express (Software AG, 2012)
Asbru	AsbruView (Kosara & Miksch, 2001; Miksch et al., 1998) CareVis (Aigner & Miksch, 2006) Uruz (Ben Gurion University, 2004)
BPMN	ActiveModeler Advantage BPM Foundation (KAISHA-Tec Co. Ltd. Japan, 2014) Bonita BPM Studio (Bonitasoft Inc., 2014)
DILEMMA	Un éditeur construit à l'aide de Protégé (Johnson & Musen, 1997)
EON	Un éditeur construit à l'aide de Protégé (Musen, 1998; Tu et al., 1995)
GLARE	GLARE's acquisition tool (Terenziani et al., 2001)
GLIF	GASTON KA-Tool (de Clercq et al., 2001; de Clercq & Hasman, 2004) P-CAPE (Zielstorff et al., 1998)
Guide/NewGuide	GUIDE (Quaglini et al., 2000) NEW-GUIDE (Ciccarese et al., 2002)
IDEFO/IDEF3	AIØ WIN® Lite (Knowledge Based Systems Inc., 2014) Edraw Max (EdrawSoft, 2014)
Little-JIL	Un éditeur construit sous la forme d'un plugin Eclipse (LASER Process Working Group, 2011; The Eclipse Foundation, 2014b)
Merise	AnalyseSI (Canonical Ltd., 2014)
OSSAD/CISAD	Workey Designer (C-Log International, 2014)
PRODIGY	Un éditeur construit à l'aide de Protégé (Hall, Jones, & Purves, 2002; Smart & Purves, 2001; Sugden, 1999)
PROforma	Arezzo (InferMed Limited, 2014) Tallis (COSSAC, 2012)
UML	PowerAMC/PowerDesigner (Sybase Inc., 2013) Microsoft Office Visio 2007 (Microsoft, 2007c)
WF-net	WoPeD (Cooperative State University Karlsruhe, 2014)
WS-BPEL	BPEL Designer (The Eclipse Foundation, 2014a) qui est un plugin Eclipse
XPDL	Enhydra JaWE (Together Teamsolutions Co. Ltd., 2013a)

Tableau 28 : Exemples d'éditeurs pour les langages de description de workflows présentés

Parmi les formalismes que nous avons listés, un grand nombre d'entre eux permettent la définition d'objectifs et d'intentions pour spécifier les workflows à un haut niveau d'abstraction (de Clercq et al., 2004). Une telle logique rejoint avantageusement celle adoptée par la modélisation MA, c'est-à-dire que le niveau 3 de notre framework permet la spécification d'un réseau de buts de haut niveau qui seront poursuivis par les SMA définis au niveau 2. On retrouve d'ailleurs cette utilisation de réseaux d'objectifs dans certains SMA comme, par exemple, RETSINA (Sycara et al., 2002, 2003). C'est pourquoi, certains de ces formalismes sont couramment utilisés pour coordonner des réseaux d'agents (Guo, 2007)(comme, par exemple, Little-JIL ou PROforma). Ces derniers ouvrent la voie à des systèmes de gestion décentralisée et dynamique des workflows (Andonoff et al., 2009). Ceci est un point important car les praticiens déplorent souvent la rigidité de l'approche algorithmique des procédures médicales (Hadorn, 1995; Lenz et al., 2012).

La multiplication de ces formats de représentation de workflows peut cependant être un véritable frein à leur dissémination entre différentes institutions (Boxwala et al., 2001). Il en existe d'ailleurs de nombreux autres dont nous n'avons pas parlé, tels que PLAN (Dube, Mansour, & Wu, 2005; Dube & Wu, 2009), SAGE (Ram et al., 2004; Tu et al., 2007), YAWL (van der Aalst, Aldred, Dumas, & ter Hofstede, 2004), SDA* (Riaño, 2007) ou encore GEM (Shiffman et al., 2000). D'autre part, tous ces formalismes ne sont pas destinés à représenter les mêmes types de processus avec le même niveau de détail, et certains d'entre eux peuvent même s'avérer complémentaires pour décrire efficacement tous les éléments qui constituent un processus de travail (Dorador & Young, 2000; Peleg et al., 2001; Terenziani et al., 2001). Néanmoins, la majorité de ces approches dispose d'une base commune d'éléments couramment appelés « primitives » permettant de décrire les étapes élémentaires des workflows (de Clercq et al., 2001; Wang et al., 2002). Il est ainsi possible d'envisager l'utilisation d'un unique langage, pour représenter les processus médicaux, qui pourra être converti vers n'importe lequel de ces formats (comme, par exemple, GESDOR (Wang et al., 2003) ou HeCaSe2 (Isern, 2008)).

Notre éditeur, pour chacun de ces niveaux utilisant des représentations graphiques, doit disposer d'une représentation hiérarchique permettant de développer les éléments d'un niveau donné. Ceci permet de détailler plus spécifiquement certains éléments dans le but de formaliser plus facilement des processus qui peuvent être relativement complexes (comme, par exemple, dans (Ciccarese et al., 2005)). Différents niveaux de détails au sein d'un même niveau vont également pouvoir se retrouver au niveau 2, au travers des structures d'organisation des agents telles que les hiérarchies ou les holarchies, et au niveau 1, à l'aide de protocoles d'interaction décrivant la dynamique de telles structures organisationnelles. On retrouve ces deux niveaux de représentation, respectivement de modélisation d'organisations d'agents et de modélisation d'agents, par exemple, dans AORML (Wagner, 2003a, 2003b) ou SKwyRL (Faulkner et al., 2004). D'autre part, les formalismes pouvant être utilisés pour décrire les protocoles au niveau 1 de l'éditeur peuvent aussi être utilisés pour les workflows au niveau 3. Il est ainsi possible de décomposer les protocoles en « micro-protocoles » afin d'augmenter leur réutilisabilité (Koning et al., 2002). Effectivement, dans l'optique de procéder à l'intégration organisationnelle et opérationnelle de procédures médicales, il est important de pouvoir modéliser de manière récursive à différents niveaux de granularité gigogne (Adam, 2000; Musen et al., 1996). Ceci permet la modélisation de « processus majeurs » et de « processus élémentaires » qui peuvent être impliqués dans plusieurs processus majeurs (Morley et al., 2011)(**Figure 163**). Cette décomposition des

workflows, en plusieurs sous-workflows, présente un intérêt pour simplifier leur modélisation par les experts ainsi que pour leur vérification à l'aide d'algorithmes (van der Aalst, 2000). Il est alors possible de mutualiser certains sous-processus impliqués dans le traitement de différentes pathologies (Gillois et al., 2001). On pourra également parler de « fragments » pour désigner les processus provenant d'une décomposition d'un processus majeur (Hollingsworth, 2004). Ainsi, ces modèles sont encapsulés les uns dans les autres dans le sens croissant des niveaux (**Figure 164**). D'une manière similaire, un même niveau peut aussi faire l'objet de descriptions encapsulées, lorsqu'il s'agit d'apporter seulement plus de détails aux éléments existants, sans pour autant nécessiter de changement de niveau d'abstraction (voir l'exemple présenté au niveau 3 sur la **Figure 164**). Cette encapsulation, au niveau des workflows, va d'ailleurs se poursuivre jusqu'à ce qu'il ne reste que des activités atomiques, qu'on appelle des « tâches » et qui sont connectées les unes aux autres à l'aide de contraintes temporelles en permettant la coordination et la synchronisation (de Clercq et al., 2001; Terenziani et al., 2003). Il devient alors possible d'effectuer ces tâches à l'aide de primitives précédemment évoquées. Dans le domaine médical, les primitives suivantes sont fréquemment identifiées (Bottrighi et al., 2013; Isern, 2008; Wang et al., 2002)(**Figure 163**) :

- Une obtention/consultation de données, impliquant directement ou non le patient.
- Une prise de décision, pouvant nécessiter ou non une intervention humaine.
- Une définition d'état/contexte, soit du patient soit de l'exécution de la procédure.
- Un passage à l'action/acte, consistant à la réalisation d'une intervention clinique.

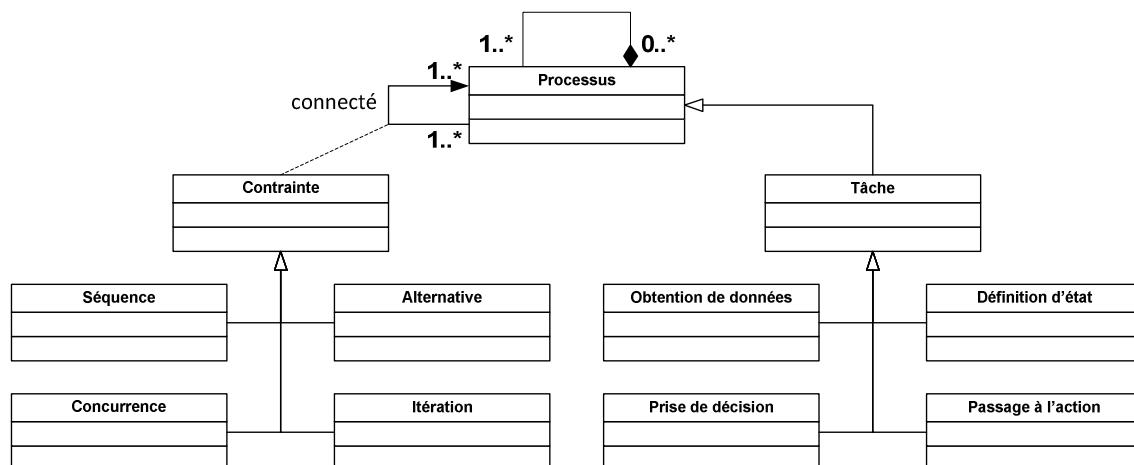


Figure 163 : Diagramme des classes composant les processus

Les prises de décisions peuvent être vues comme des activités menant à l'instanciation d'activités sous certaines contraintes. Elles sont automatiques ou semi-automatiques selon qu'elles nécessitent ou non l'intervention d'un humain (Guo, 2007). Les contraintes d'ordonnancement, basées sur la précédence d'activités ou déclenchées par une horloge ou un minuteur (Wang et al., 2001), peuvent prendre la forme de (**Figure 163**) : séquences, concurrences, alternatives et itérations. On signalera également que chaque workflow doit disposer d'un point d'entrée et d'au moins un point de sortie.

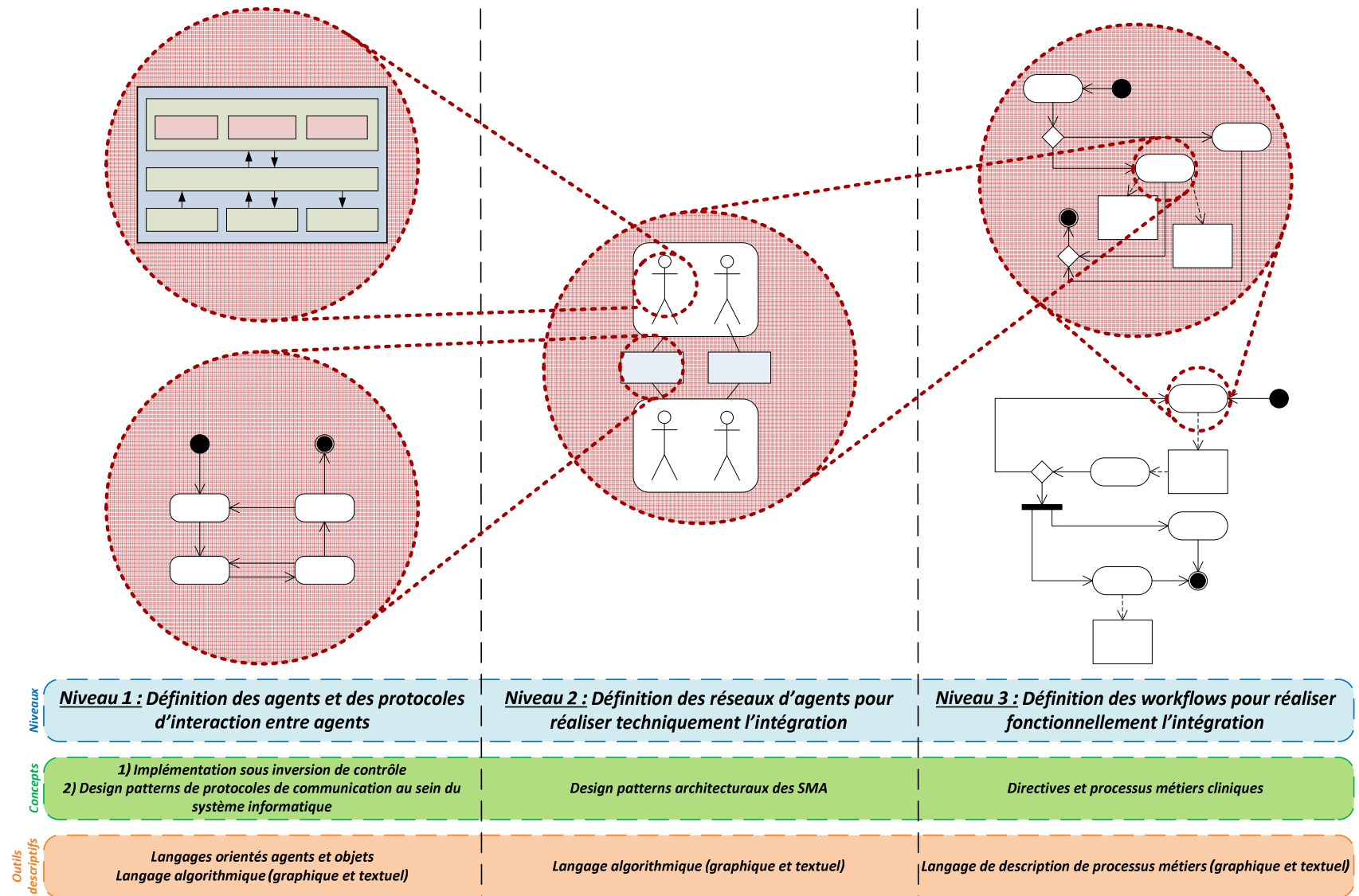


Figure 164 : Schéma des relations entre les trois niveaux conceptuels de l'éditeur

d) Les relations entre les niveaux de l'éditeur

L'idée sous-jacente aux niveaux 3 et 2 est similaire à celle explorée par Klinker et al. (1991) à l'aide, respectivement, des outils Spark et Burn. Effectivement, l'idée justifiant l'existence de ces niveaux est d'aider un utilisateur, n'ayant pas nécessairement de connaissances en programmation, à réaliser une application. Pour cela, notre outil va l'assister et lui proposer des éléments préconstruits qu'il lui faudra connecter et configurer pour son usage spécifique. Le niveau 1, quant à lui, permet d'aller plus loin en implémentant de nouveaux éléments, utilisables dans les niveaux supérieurs, ou en modifiant les comportements d'éléments existants. A ce niveau, on précise aussi quels sont les protocoles utilisés lors des interactions entre agents : on s'assure ainsi qu'à l'exécution chaque agent aura accès à une copie de ce protocole (ou de protocoles compatibles), ce qui est indispensable pour supporter correctement leurs échanges (Koning & Pesty, 2001).

De la sorte, notre éditeur, qui permet une telle conception assistée de workflows, doit lui-même intégrer intrinsèquement un workflow de conception de workflows. En clair, l'éditeur doit proposer une série d'étapes à travers les différents niveaux que nous avons évoqués pour arriver à l'application telle qu'elle sera exécutée par la PAMMA (voir la section **L'utilisation du framework au travers de l'éditeur**). Pour cela, les différents niveaux de l'éditeur doivent partager des références communes pour que le passage de l'un à l'autre s'inscrive dans une logique commune. A ce titre, nous avons identifié les activités de transition d'un niveau à l'autre qui suivent (par exemple, consulter également (Wilk et al., 2008))(Figure 164) :

- Du niveau 3 au niveau 2 : identifier le(s) rôle(s) qu'occupe chaque agent. Ces rôles sous-tendent les buts poursuivis par les agents et visent à satisfaire les besoins fonctionnels décrits par les experts métiers. C'est au niveau 2 que l'on précise où et comment vont être récupérées les données, qui sont fréquemment spécifiques à l'organisation à intégrer. Ceci revient à faire exécuter les activités du workflow par des organisations d'agents (Chen et al., 1999).
- Du niveau 2 au niveau 1 (comme, par exemple, dans (Sayyad Shirabad et al., 2012)) :
 - o Implémenter les agents : identifier et hiérarchiser les comportements à implémenter pour assurer les rôles. Ces comportements devront être programmés.
 - o Spécifier les protocoles : identifier les protocoles d'interaction entre agents. Ces protocoles peuvent impliquer deux agents ou plus. Ceci revient à organiser la coopération entre agents à l'aide de protocoles, c'est-à-dire de workflows (Chen et al., 1999). La distinction qui peut être faite avec ceux du niveau 3 est qu'ils modélisent des processus de travail alors que ceux du niveau 1 modélisent des processus de coordination (Andonoff et al., 2009).

Pour finir, on fera remarquer qu'il peut être judicieux de pouvoir associer à un modèle de workflows, constitué d'éléments des trois niveaux conceptuels présentés dans cette section, un certain nombre de métadonnées, tel qu'un nom pour le workflow, ses auteurs ou encore un numéro de version. En effet, ce type d'informations facilite la maintenabilité, la spécialisation et l'évolution incrémentale des modèles ainsi produits (de Clercq et al., 2004). La mise en place d'un tel système doit d'ailleurs bénéficier à tous les modèles issus des

différents niveaux de l'éditeur, et pas seulement à ceux du niveau 3 (pour un exemple de l'intérêt de gérer la filiation dans la conception de protocoles, c'est-à-dire au niveau 1 de l'éditeur, consulter (Kuwabara et al., 1995b)).

2) L'encodage

Le but de l'encodage est de passer d'une représentation graphique, spatiale et temporelle donc établie selon plusieurs dimensions, en une description textuelle, linéaire et formelle, apte à être informatiquement exécutée (de Clercq et al., 2004; Isern, 2008). De telles descriptions reposent sur des « langages d'exécution » (Owen & Raj, 2003), tels que, par exemple, RDF (World Wide Web Consortium, 2014) utilisé dans le cas de GLIF (Wang et al., 2004), ou encore PNML (Billington et al., 2014) pouvant être utilisé par WF-net (Casalicchio & Tucci, 2013), et sont fréquemment formatées à l'aide d'XML (Dubey & Chueh, 1998; World Wide Web Consortium, 2013) afin d'être facilement compréhensibles par des êtres humains ainsi que des machines, mais aussi d'être aisément extensibles (Xu et al., 2003)(pour des exemples d'application, consulter (Dart et al., 2001; Dube et al., 2005; Shiffman et al., 2000)). Hormis la possibilité d'exécuter directement des représentations encodées, ces dernières ont l'indéniable avantage de pouvoir être partagées entre différentes institutions. D'autre part, on pourra également remarquer qu'XML est adapté pour la modélisation entité-association (Chen, 2002), c'est-à-dire pour formater efficacement les représentations graphiques des SMA telles que précédemment évoquées.

Notre outil va devoir faciliter l'exportation de descriptifs des processus (par exemple, au format XML (World Wide Web Consortium, 2013) comme le permet XPD 2.2 (Workflow Management Coalition, 2012) ou encore SDM-XML 1.0 (Clinical Data Interchange Standards Consortium, 2011)). A ce titre, on signalera également que, lorsque Huser et al. (2011) ont développé leur moteur de workflows, ils ont décidé d'utiliser le langage XPD (Workflow Management Coalition, 2012) car ce dernier est majoritairement utilisé par l'industrie (Huser, Rasmussen, & Starren, 2009) et qu'il offre une correspondance entre représentations graphique et textuelle (pour un exemple d'utilisation, consulter (Huser et al., 2006)). Une telle correspondance est importante car elle facilite d'autant l'encodage. Ceci illustre le fait que tous les langages précédemment listés ne sont pas égaux quand il s'agit de passer à l'encodage. En effet, plus ils sont formels, plus il sera aisé de les encoder.

Finalement, nous signalerons que les workflows, les SMA et les protocoles doivent faire l'objet de méthodes systématiques de vérification, qui peuvent être empiriques ou formelles, avant d'être exécutés (de Clercq et al., 2004). Nous ne détaillerons pas ici ce type de procédures, néanmoins nous invitons le lecteur à se référer aux différentes publications relatives à ce sujet dans le cas des workflows (consulter, par exemple, (Lu, Bernstein, & Lewis, 2006; van der Aalst, 2000)), dans celui des SMA (consulter, par exemple, (Dastani, Hindriks, & Meyer, 2010; Hutzler, Kludel, & Wang, 2006)) ainsi que dans celui des échanges de messages et des protocoles (consulter, par exemple, (Bakar & Selamat, 2013; Huget & Koning, 2003)). Ainsi, notre produit devra disposer d'un outil de débogage et de simulation, permettant d'assurer la sûreté des processus destinés à être exécutés ainsi qu'à analyser leurs performances (comme, par exemple, ceci est réalisé dans (Casalicchio & Tucci, 2013)). Effectivement, l'absence de preuve de validité clinique est un reproche couramment fait par les praticiens à l'encontre de ce genre d'outils (Hadorn, 1995).

3) L'exécution

L'exécution consiste à accomplir informatiquement des tâches décrites dans un formalisme correctement encodé. On remarquera que les différents formalismes de description de workflows, que nous avons précédemment cités, disposent tous de moteurs ou de modules d'exécution (par exemple, dans le cas d'Arden Syntax, consulter (Kuhn & Reider, 1994), dans celui de GLIF, consulter (Wang & Shortliffe, 2002; Wang et al., 2004), dans celui d'EON, consulter (Tu & Musen, 2000) ou encore Enhydra Shark (Together Teamsolutions Co. Ltd., 2013b) dans celui de XDPL (pour un exemple d'application, consulter (Huser et al., 2006))). Quel que soit le formalisme choisi, les activités d'une application en exécution doivent pouvoir exhiber différents états représentatifs de l'avancement des traitements informatiques s'y rapportant (comme on peut en retrouver, par exemple, dans (Dadam et al., 2000; Herbert et al., 1995; Shahar et al., 1996; Sutton & Fox, 2003; Tu & Musen, 1996; Wang & Shortliffe, 2002)). Nous avons choisi de retenir les états des activités instanciées suivants :

- Active, qui indique que l'activité est en train d'être effectuée. Les SMA et agents utilisés pour cette activité sont a fortiori instanciés.
- Terminée, qui signifie que l'activité a été correctement effectuée. Le SMA et les agents impliqués de façon exclusive dans cette activité sont alors détruits.
- Abandonnée, qui signale que l'activité a été interrompue avant son terme. Cet état relève d'un abandon provoqué volontairement par l'utilisateur ou d'une erreur du système.
- En attente, qui précise que l'activité n'a pas encore été effectuée. Ce qui ne veut d'ailleurs pas dire qu'elle le sera systématiquement.
- Suspendue, qui annonce que l'activité est remise à plus tard. Cet état permet de mettre en attente une activité déjà débutée. L'instance d'une telle activité peut, à ce titre, potentiellement être retirée de l'environnement courant d'exécution.

Ces multiples états fournissent des informations utiles au monitoring des workflows ainsi qu'à la gestion de la mémoire par le service de workflows (Alsinet et al., 2000)(**Figure 8**). La **Figure 165** illustre les relations entre ces différents états possibles des activités.

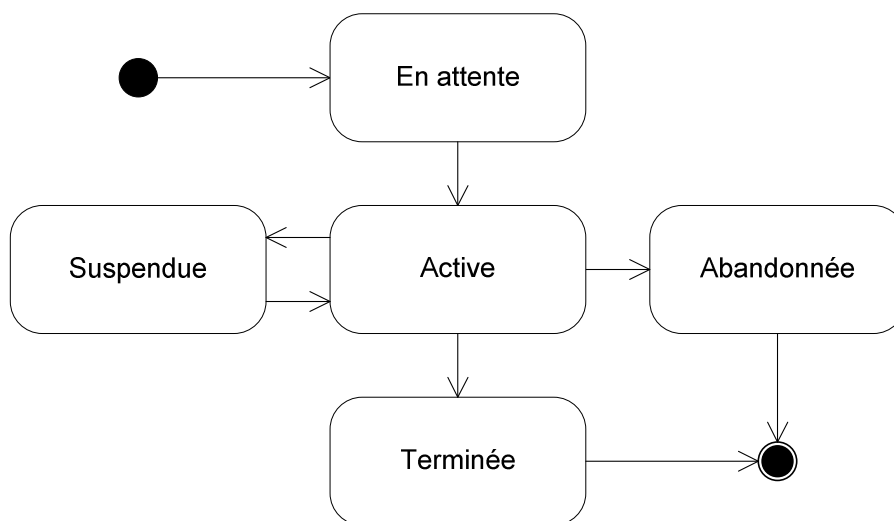


Figure 165 : Diagramme d'états-transitions des activités

D'autre part, ces différents états sous-tendent qu'une PAMMA est capable, lors d'une exécution, d'assurer différents types de fonctionnalités de gestion des cycles de vie des activités, des SMA et des agents. Dans la suite de cette sous-section, nous allons ainsi discuter des mécaniques suivantes :

- Le chargement/instanciation d'une application.
- L'exécution de l'application par les agents.
- Le fonctionnement des services indispensables de la PAMMA.
- La destruction des agents/applications.

Fonctionnellement, il est important, pour la bonne intégration du système informatique dans les processus de travail, que les professionnels de santé puissent consulter n'importe quand l'état d'avancement de chaque activité (Chen, 2010). En outre, nous rappelons que de telles activités peuvent être réalisées dans des endroits différents par des professionnels différents, c'est pourquoi notre système doit pouvoir supporter une exécution distribuée (Bottrighi et al., 2013). Il est également notable que le service de workflows permet l'exécution des modèles dynamiques que sont les workflows. A ce titre, il pourrait très bien être utilisé comme simulateur pour évaluer/vérifier les processus métiers, permettre leur visualisation et leur compréhension ou encore effectuer des prévisions concernant leurs différents comportements selon les situations (Treuil et al., 2008).

a) Le chargement/instanciation d'une application

Dans notre système, l'instanciation d'une application revient à l'exécution d'un workflow, c'est-à-dire l'exécution d'un modèle du niveau 3 de notre éditeur. Nous rappelons que cette exécution doit pouvoir être partiellement réalisée de manière distribuée (Leymann & Roller, 1997). En effet, des sous-workflows doivent pouvoir être répartis sur différentes PAMMA (comme, par exemple, ceci est proposé dans (Ehrler et al., 2005; Yan et al., 2007)). Il est ainsi possible de minimiser les chances qu'une PAMMA devienne un goulet d'étranglement et tire les performances globales du système vers le bas. D'autre part, il devient aisé de mettre à jour certains sous-workflows sans impacter sur la totalité du workflow.

Cette structure gigogne des applications va, de la sorte, pouvoir faire l'objet d'un traitement récursif et potentiellement distribué. En particulier, sur les modèles des niveaux 2 et 3, il va être déterminé si un élément est atomique, c'est-à-dire s'il est directement instanciable ou s'il est composé d'autres éléments de son propre niveau d'abstraction : cette étape va alors conditionner la récursion et/ou la distribution de portions du chargement de l'application (**Figure 166**). L'instanciation d'une application et son exécution sont étroitement liées. En effet, de par la nature même des applications décrites à l'aide de workflows, une application ne charge que les éléments relatifs à son activité courante, les utilise pour réaliser cette activité puis les libère pour passer à l'activité suivante. La **Figure 166** présente un algorithme de principe du chargement des applications par les PAMMA, où seul un fonctionnement général est présenté. Une représentation plus détaillée devrait faire apparaître la gestion d'une pile permettant le traitement récursif et distribué des workflows et des SMA.

La **Figure 167** apporte quelques précisions quant aux relations qu'entretiennent les éléments de différents niveaux d'abstraction. En effet, cette figure montre comment une entité du niveau $i + 1$ (où $i \in [1,3]$ et où le niveau 4 désigne le gestionnaire de workflow) crée et

initialise une entité du niveau i . Cette dernière va ensuite effectuer son travail, faire son rapport à l'entité du niveau $i + 1$ puis mettre un terme à son existence. Ces relations n'ont évidemment lieu d'être que s'il existe un apport holarchique entre ces entités.

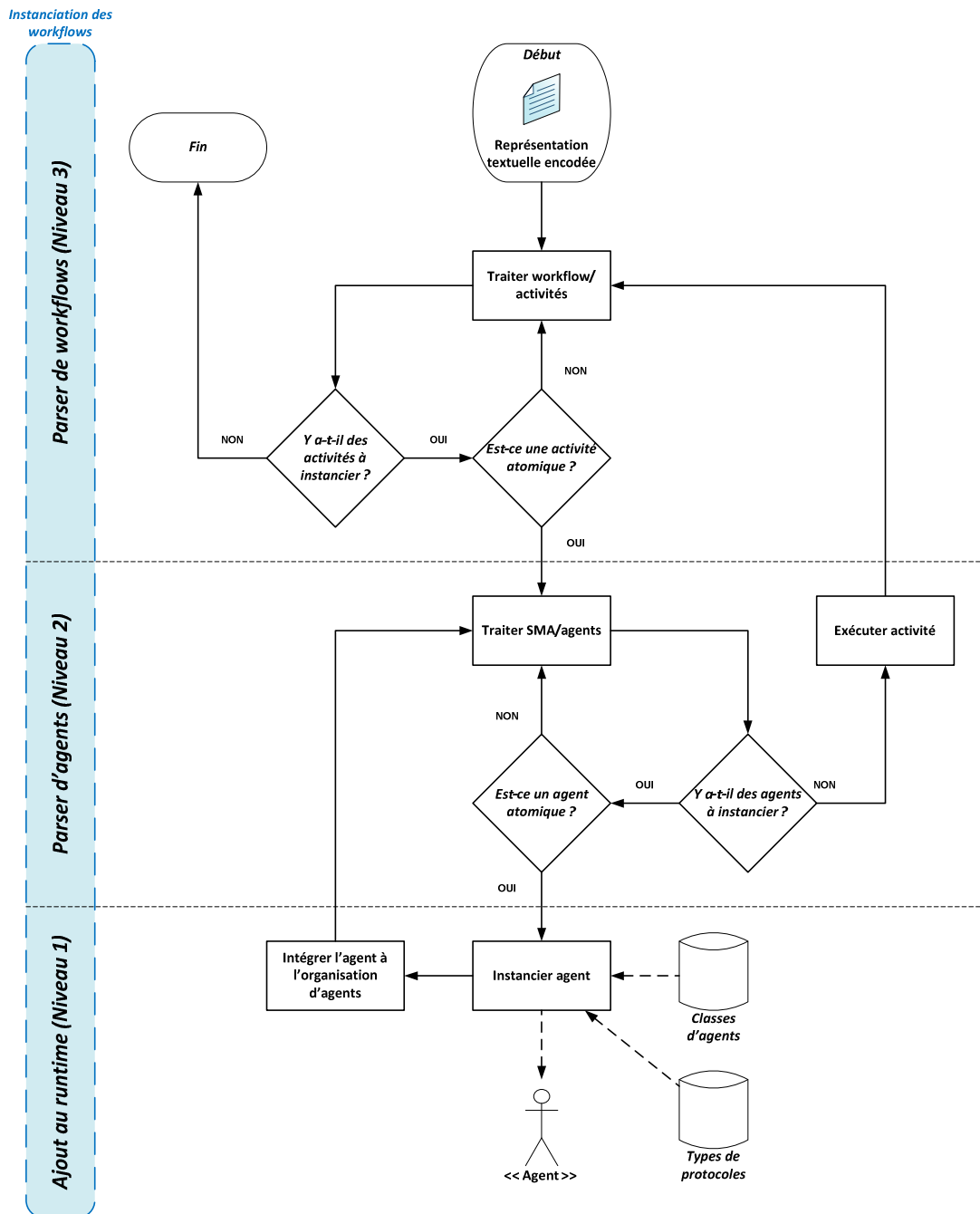


Figure 166 : Ordinoigramme du principe d'instanciation des applications

On remarquera que la population d'agents est rarement déterminée avant l'exécution du programme, c'est-à-dire lors du « runtime » (Giorgini & Henderson-Sellers, 2005). Il est ainsi possible de définir « à la volée » des activités, et a fortiori des agents, correspondant aux besoins applicatifs des utilisateurs. A ce titre, pendant l'exécution d'un workflow, l'utilisateur doit pouvoir s'affranchir d'une tâche ou la substituer par une autre présente ou non dans le workflow (Dadam et al., 2000; Quaglini et al., 2000). Ce type de modification du comportement attendu du workflow devra être adjoit d'une justification fournie par

l'utilisateur. Il s'agit d'un mode de gestion dynamique des workflows par l'utilisateur sur lequel nous reviendrons ultérieurement (voir la sous-section **L'aide à la décision**).

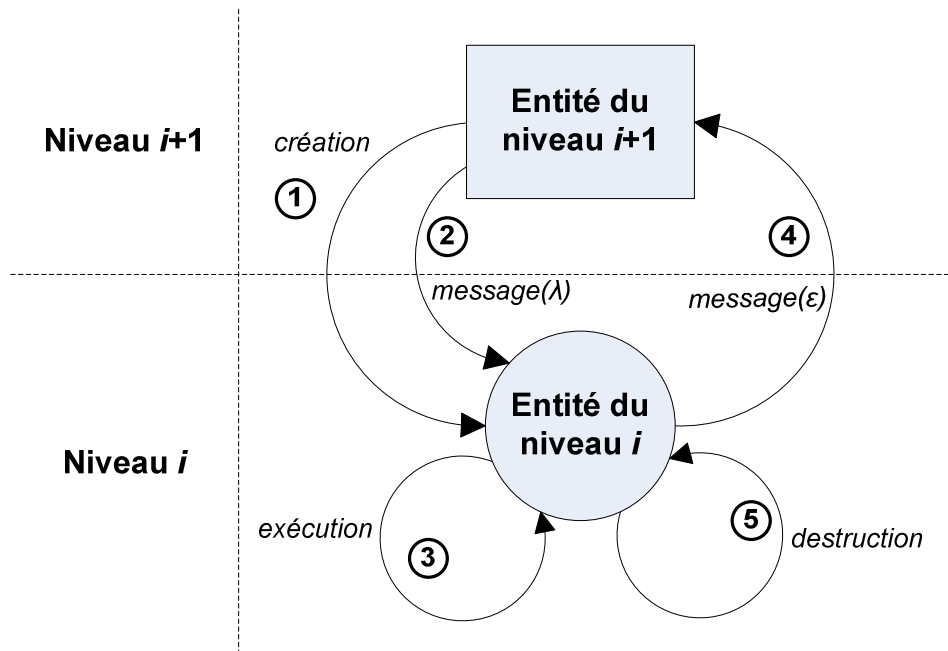


Figure 167 : Relations entre des entités de niveaux conceptuels différents

Finalement, nous précisons que les instanciations d'activités sont réalisées par le gestionnaire de workflows alors que celles d'agents le sont par le manager d'agents.

b) L'exécution de l'application par les agents

Bien que la **Figure 167** présente une approche générique pour l'exécution d'applications, elle ne rend compte ni des subtilités de notre approche ni des spécificités des différents niveaux d'abstraction impliquant des agents. Nous allons maintenant apporter quelques précisions que nous jugeons indispensables à la compréhension des mécaniques d'exécution, à l'aide de SMA, des activités des applications (**Figure 168**). Ces informations ont un intérêt pour la conception même des applications, étant donné que leur maîtrise peut conditionner le degré d'élégance des modélisations réalisées.

Chaque agent doit être apte à recevoir et envoyer des messages, ainsi qu'à adopter des comportements proactifs ou réactifs lui permettant de poursuivre des buts qui lui sont propres (Shoham, 1993). Bien que les comportements puissent être de natures extrêmement variées, les buts qu'ils servent s'inscrivent dans la poursuite des buts globaux du SMA auquel ces agents appartiennent. Pour cela, ces derniers doivent interagir et s'organiser, ce qui conditionne l'architecture du SMA. Ainsi, dans le cas de la définition et l'exécution de processus par notre PAMMA, il existe différents types d'architectures de SMA dans lesquelles les agents sont :

- Connectés statiquement (comme, par un exemple, dans (Kirn et al., 2003)).
- Connectés dynamiquement (comme, par exemple, dans (Gleizes & Glize, 2002)).

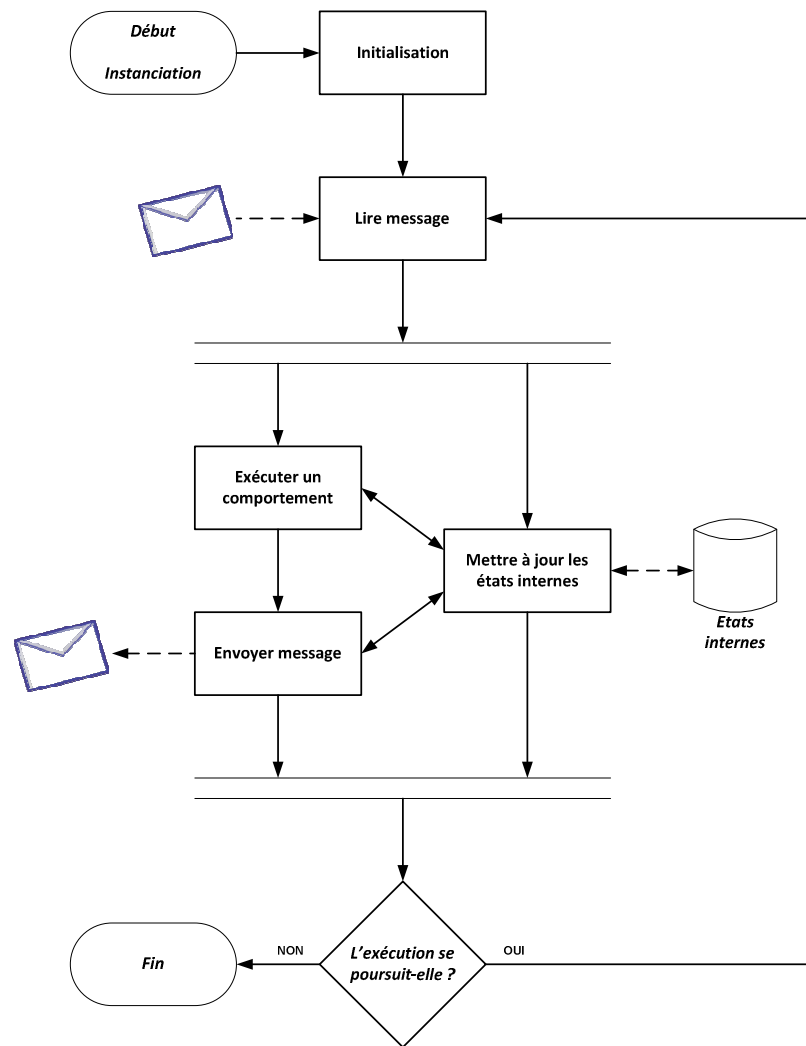


Figure 168 : Ordinogramme du principe d'exécution d'un agent

Notre solution cherche à permettre ces deux types d'approche à l'aide des services et des médiateurs/facilitateurs de la plate-forme. L'utilisation de protocoles d'interactions entre agents prend alors tout son sens. Afin d'éviter toute introduction d'erreur au niveau des protocoles, la méthode la plus simple est de les exécuter directement à partir des représentations conformes aux formalismes dans lesquels ils sont spécifiés (Fonseca i Casas, 2014; Huget, 2001). Ceci revient à interpréter de telles spécifications d'une manière similaire à celle appliquée à l'exécution des workflows. Néanmoins, dans le cas des protocoles, l'interprétation n'est pas réalisée par un agent dédié, ayant une vue de l'ensemble de l'application, mais elle est effectuée au niveau de chaque agent par le gestionnaire de protocoles se situant dans le module de communication (voir la sous-sous-section **La communication entre agents**). Les états de l'ensemble des instances de protocoles peuvent cependant faire l'objet d'une centralisation par le biais d'un agent spécialisé de l'infrastructure du système (comme, par exemple, dans (Esteva et al., 2004) mais sans nécessairement ne se restreindre qu'à une communication entre les agents et ce gestionnaire de protocoles). C'est pourquoi nous avons établi que notre PAMMA devrait disposer d'un agent en charge de la gestion des états des interactions entre agents, tandis que chaque agent devrait intégrer un module d'exécution de ces états d'interaction. Chaque agent consulte alors ce service de gestion des protocoles afin de se constituer localement

une liste des performatives/primitives qu'il s'attend à recevoir ainsi que des comportements qu'il devra alors mettre en œuvre (**Tableau 29**).

Performative/Primitive	Protocole	Comportement associé

Tableau 29 : Types des données gérées par le gestionnaire de protocoles du module de communication

La structure décrite par le **Tableau 29** est composée dynamiquement à l'aide de l'ensemble des protocoles dans lequel l'agent est impliqué à un instant donné. Si la syntaxe d'un message est correctement vérifiée par le répartiteur de messages, ce dernier est pris en charge par le gestionnaire de protocoles. La structure est utilisée à la réception comme suit :

- Si un tuple correspond à la performative/primitive et au protocole spécifiés dans le message, alors lancer le processeur de message avec en entrée :
 - o Le message.
 - o Le comportement associé.
- Sinon, rejeter le message.

Chaque agent a une durée de vie variable qui correspond au mieux à la durée de vie du SMA auquel il appartient. De la même manière, ces SMA, définis au niveau 2, ont une durée de vie correspondant au maximum à celle de l'activité de niveau 3 à laquelle ils se réfèrent. En outre, similairement aux activités qui peuvent entretenir une relation d'encapsulation (**Figure 164**), les SMA participant à instancier l'application pourront être encapsulés. Leur portée temporelle et opérationnelle sera alors conditionnée par leur niveau d'encapsulation.

c) Le fonctionnement de services indispensables de la PAMMA

Notre but est de proposer une boîte à outils, à la fois robuste et souple, constituée d'agents qu'il sera possible d'instancier, paramétrer et organiser en fonction de la situation. Parmi ces agents, ceux assurant des services de la PAMMA sont particulièrement importants dans la mesure où ils sont sensés être toujours disponibles et donc utilisables par n'importe quelle application fonctionnant sur la plate-forme. Nous avons décidé de fournir des renseignements sur les services, couramment présents dans les plates-formes MA existantes (voir la sous-sous-section **De la microarchitecture à la macro-architecture**), suivants :

- Le manager d'agents, qui gère les pages blanches (**Figure 141**).
- Le manager de services, qui gère les pages jaunes (**Figure 142**).
- Le manager de rôles et protocoles, qui maintient les états des interactions.

Ces trois premiers services utilisent des tableaux blancs/noirs pour présenter aux agents des informations. Nous rappelons que les tableaux blancs/noirs (Corkill, 2003) sont des dépôts d'informations mis à la disposition de l'ensemble des agents. Ils permettent de mutualiser des données qu'il aurait sinon fallu dupliquer au sein de plusieurs agents différents. On remarquera également qu'est instaurée une sorte de protocole implicite entre les agents et les tableaux noirs. Ces tableaux peuvent être constitués de plusieurs fragments, et les accès concurrents y sont contrôlés, par exemple, à l'aide de sémaphores.

Le manager d'agents en lui-même a en charge l'instanciation et la destruction des agents. Le tableau noir/blanc des pages blanches (**Tableau 30**) permet de référencer tous les agents instanciés localement. Chacun d'eux se voit attribuer un identifiant unique au moment de son instanciation. Consulter les pages blanches permet d'obtenir une référence à l'agent au sein de la PAMMA afin de pouvoir interagir avec lui.

ID	Nom de l'agent	Classe de l'agent	Application	Accointances	Référence
:	:	:	:	:	:

Tableau 30: Types des données gérées par le tableau noir/blanc des pages blanches

On précisera que, pour chaque agent, les accointances consistent en une liste d'identifiants d'agents. Cette liste est mise à jour dès qu'un agent ajoute (ou supprime) un agent à sa liste interne d'agents « connus », c'est-à-dire qu'il a directement la référence de l'agent, son adresse, et qu'il n'a plus besoin de passer par les pages blanches pour le localiser. Les agents présents dans cette liste vont pouvoir être contactés :

- Pour suppression, si l'agent est détruit.
- Pour mise à jour, si l'agent est déplacé.

Le manager de services permet de référencer tous les services utilisables par les applications locales (et éventuellement distantes). Ces services, locaux ou distants, sont identifiés à l'aide d'une adresse (**Figure 169**) et d'une liste des interfaces des services proposés (**Tableau 31**). Ces interfaces sont des paires de performatives/primitives et de protocoles.

Adresse de l'agent	Interface des services
:	:

Tableau 31 : Types des données gérées par le tableau noir/blanc des pages jaunes

Les adresses servent à retrouver facilement un agent dans les pages blanches. En effet, elles contiennent toutes les informations utiles à une telle identification (**Figure 169**).

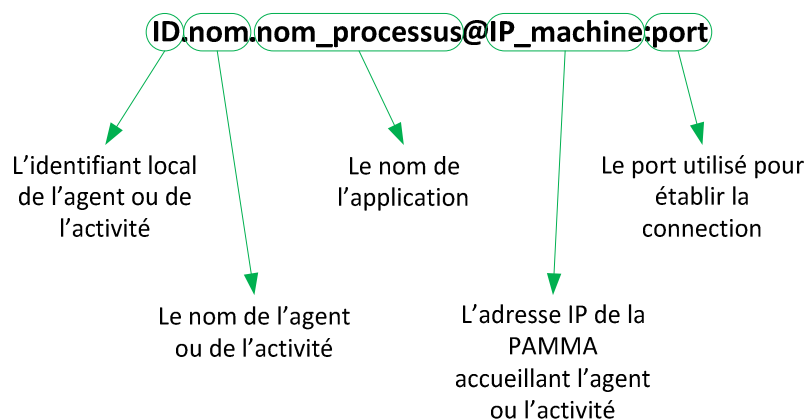


Figure 169 : Système d'adressage de la PAMMA

L'IP et le port ne sont utilisés que pour les agents distants. Ces deux informations seront utilisées par les médiateurs/facilitateurs pour contacter la PAMMA distante qui accueille le

service en question. De plus, on peut remarquer que ce système peut s'appliquer aux agents mais également aux activités dans le cas de sous-workflows distribués.

Le manager de rôles et protocoles sert à garantir que les interactions entre agents au sein d'un SMA se passent conformément aux protocoles spécifiés. Ce service permet de stocker les protocoles sous une forme consultable et exécutable par les agents (**Figure 170**). Effectivement, dans un environnement ouvert, il n'est pas réaliste de vouloir encoder les protocoles au sein de chaque agent (Paurobally & Cunningham, 2003). Nahm et Ishikawa (2005) proposent un concept similaire au travers de l'interface qu'ils nomment HIA.

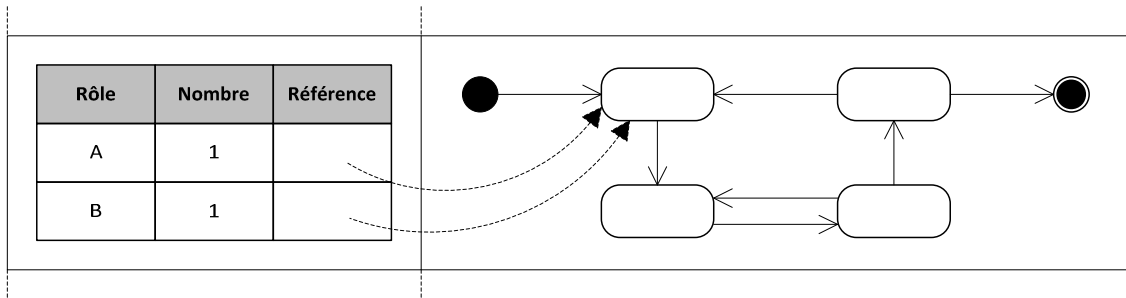


Figure 170 : Exemple de fragment du tableau noir/blanc des états des protocoles

Concernant la **Figure 170**, on peut préciser que l'accès à un tel fragment est régulé à l'aide d'un sémaphore afin d'éviter les accès concurrents aux informations de cette instance de protocole. Une liste d'agents « écoutant » les modifications sur le fragment peut également être gérée au niveau de ce dernier afin, par exemple, de transmettre ces informations au service de monitoring (qui peut, par exemple, être similaire à ce qu'on rencontre avec Enhydra Shark (Together Teamsolutions Co. Ltd., 2013b)). Il est effectivement important de pouvoir ainsi analyser la qualité des interactions entre agents (Bakar & Selamat, 2013).

Pour finir, nous signalerons que les classes de ces agents de service doivent directement être disponibles pour le développement d'applications au niveau de l'éditeur. De cette façon, il est possible d'instancier un service privé d'une application ou partie d'application spécifique.

d) La destruction des agents/applications

Une gestion correcte de la mémoire des applications passe par la destruction des éléments qui ne servent plus. Pour garantir des performances optimales, il est effectivement primordial de gérer efficacement la mémoire utilisée par le système (da Silva & de Lucena, 2004). Les agents en charge de cette destruction sont :

- Le gestionnaire de workflows dans le cas des activités.
- Le manager d'agents dans le cas des agents.

Comme présenté sur la **Figure 167**, la destruction est précédée d'une notification à l'entité de niveau supérieur. Celle-ci permet de transmettre les résultats obtenus mais également d'initier les traitements découlant de la suppression (mise à jour des pages blanches, passage à l'activité suivante, etc.). On ne détaillera pas la gestion de la mémoire allouée à une PAMMA, cependant c'est un point important à étudier ultérieurement.

D) L'utilisation du framework au travers de l'éditeur

Pour clôturer ce chapitre, nous allons donner un aperçu du processus de développement d'applications sur la PAMMA ainsi qu'une vue d'ensemble de l'architecture de cette dernière. Nous expliquerons plus loin dans ce document (voir la partie **L'analyse des résultats et les perspectives**) de quelle manière peut précisément se monter une collaboration entre professionnels de santé et expert en informatique afin de développer des applications pour une ou plusieurs PAMMA.

1) Le processus de développement d'applications

Le processus de développement d'applications pour la PAMMA s'articule autour de l'éditeur que nous avons maintes fois traité. Cet éditeur est un système de conception assistée par ordinateur (comme peut, par exemple, l'être agentTool (Garcia-Ojeda et al., 2009a, 2009b)). Pour rappel, il propose de travailler sur trois niveaux d'abstractions différentes :

- Le niveau 3, qui permet la réalisation de l'analyse fonctionnelle, c'est-à-dire la description des fonctionnalités qu'on veut voir émerger.
- Le niveau 2, qui consiste en la conception technique de l'application, c'est-à-dire la constitution de SMA constitués d'agents et de leur organisation.
- Le niveau 1, qui sert à l'implémentation technologique, c'est-à-dire le codage des modules, des comportements et des protocoles dans un langage de programmation.

Ainsi, conformément aux recommandations de MDA (Object Management Group, 2014c), l'éditeur permet la mise en place d'un processus de développement d'applications pour la PAMMA. Un des points noirs fréquemment rencontrés avec les éditeurs de workflows est qu'ils requièrent l'intervention d'experts métiers et d'ingénieurs logiciels dont la séparation entre leurs domaines d'intervention respectifs n'est pas forcément claire (**Figure 171**). Cela peut rendre l'utilisation de l'outil difficile, c'est pourquoi nous avons séparé ces domaines avec les différents niveaux conceptuels de l'éditeur. Notre outil de conception de workflows doit ainsi intégrer intrinsèquement un workflow de conception de workflows : l'éditeur doit donc proposer une séquence d'étapes à travers ses différents niveaux pour arriver à une application exécutable par la PAMMA.

Les tâches de spécification doivent s'effectuer, de manière itérative, du niveau 3 vers le niveau 1 (comme, par exemple, ceci est décrit dans (Nwana et al., 1999)), c'est-à-dire selon une logique de décomposition cartésienne de la problématique métier. Il est ainsi possible de définir un véritable processus de développement en utilisant notre framework (comme on peut en retrouver un similaire, par exemple, dans (Yu & Schmid, 1999)) :

- Au niveau 3 (workflow), pour le processus métier :
 - o Définir les activités et leur chronologie de manière itérative jusqu'à n'avoir que des activités atomiques.
 - o Définir les objectifs de ces activités.
 - o Définir les ressources humaines et matérielles requises pour l'accomplissement de ces objectifs.
- Au niveau 2 (SMA), pour chaque activité :

- Définir les agents représentant les ressources et les traitements (à partir d'agents existants ou en implémentant de nouveaux agents).
- Définir les comportements de ces agents qui participent à la réalisation des activités.
- Définir la manière dont sont organisés ces agents, à l'aide de protocoles au sein desquels chacun se voit attribuer un rôle en vue de satisfaire un ou plusieurs des objectifs précédemment définis.
- Au niveau 1 (implémentation des agents et protocoles), pour toutes les classes d'agents et de protocoles :
 - Programmer les composants des agents (tels que leurs comportements ...).
 - Définir les rôles des interacteurs, les états des interactions et les transitions (à l'aide des performatives/primitives) entre elles pour spécifier les protocoles.

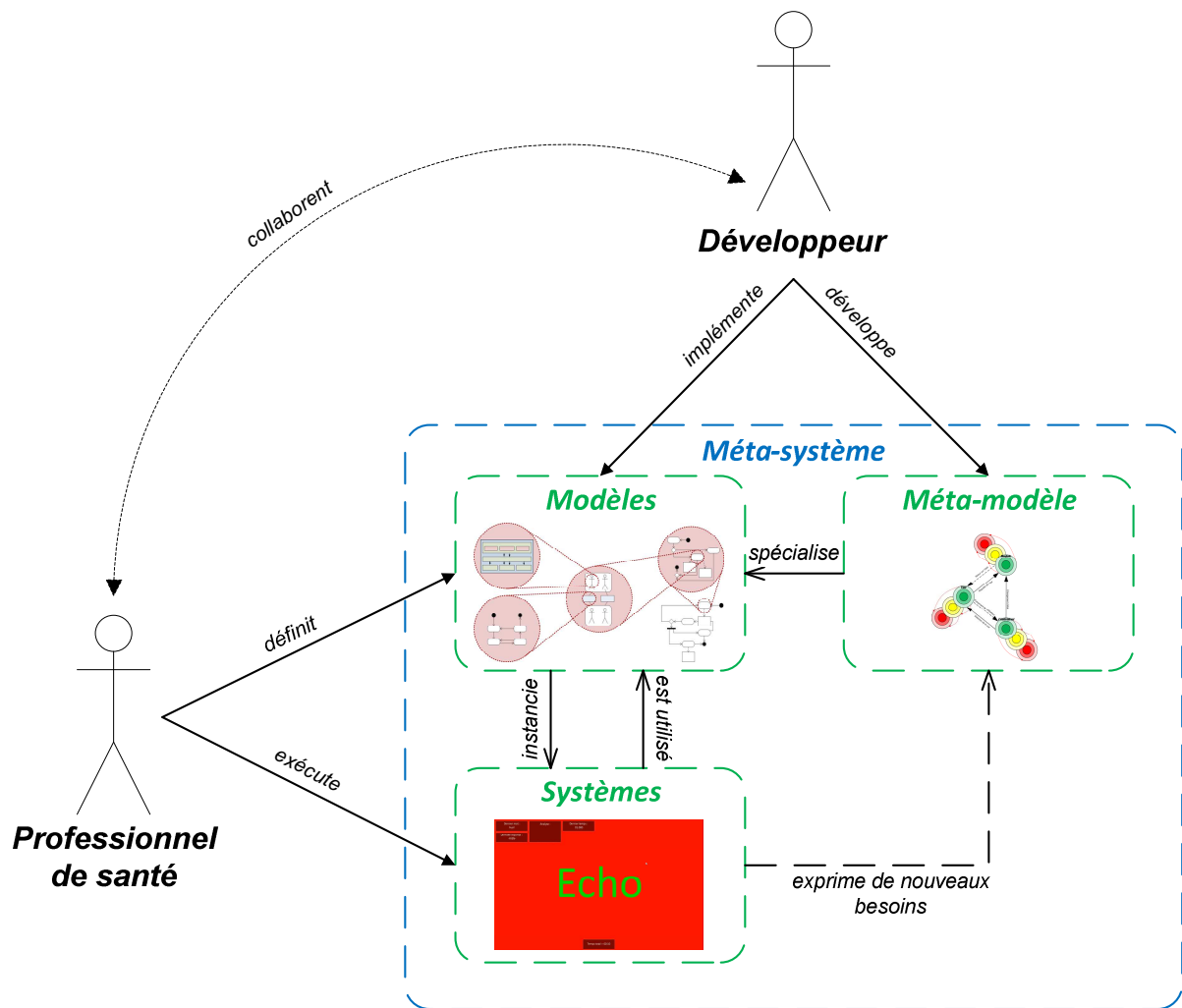


Figure 171 : Représentation du principe d'utilisation du framework

La **Figure 171** présente le méta-système dans lequel s'inscrit le développement d'applications pour la PAMMA. En particulier, on peut y voir que le développement d'applications permet de faire des retours d'expérience sur les éléments du framework (Treuil et al., 2008) et de les améliorer conformément à notre modèle de modélisation (**Figure 35**). Effectivement, c'est à partir des besoins liés aux applications médicales qu'on pourra déterminer quels éléments, que nous avons précédemment appelés « éléments de

base », doivent être disponibles au niveau du méta-modèle (voir, par exemple, le chapitre **L'application de l'UP7: un modèle générique de prise en charge de la dyslexie**).

La **Figure 172** présente un prototype de l'IHM de l'éditeur. En particulier, on peut brièvement préciser que ce dernier doit :

- Permettre la conception d'applications distribuées, c'est-à-dire qu'il permet :
 - o D'accéder à tous les services des pages jaunes (locales ou distantes auxquelles l'utilisateur a un droit d'accès).
 - o D'accéder également aux différents éléments des services de données, internes ou externes à la PAMMA.
 - o D'éditer et d'exécuter des applications conformément avec ce qu'autorisent les privilèges de l'utilisateur qui utilise l'éditeur.
- Faciliter le développement des comportements des agents à l'aide d'IHM adaptées aux différentes couches du framework (à la Vue au Contrôleur ou au Modèle).

Nous pouvons ainsi nous concentrer sur la question essentielle « quels sont les types d'agents requis ? », plutôt que « combien d'agents seront instanciés ? » (Mahmoud & Maamar, 2006). Le nombre d'agents ne sera effectivement déterminé qu'à l'exécution en fonction des besoins des utilisateurs.

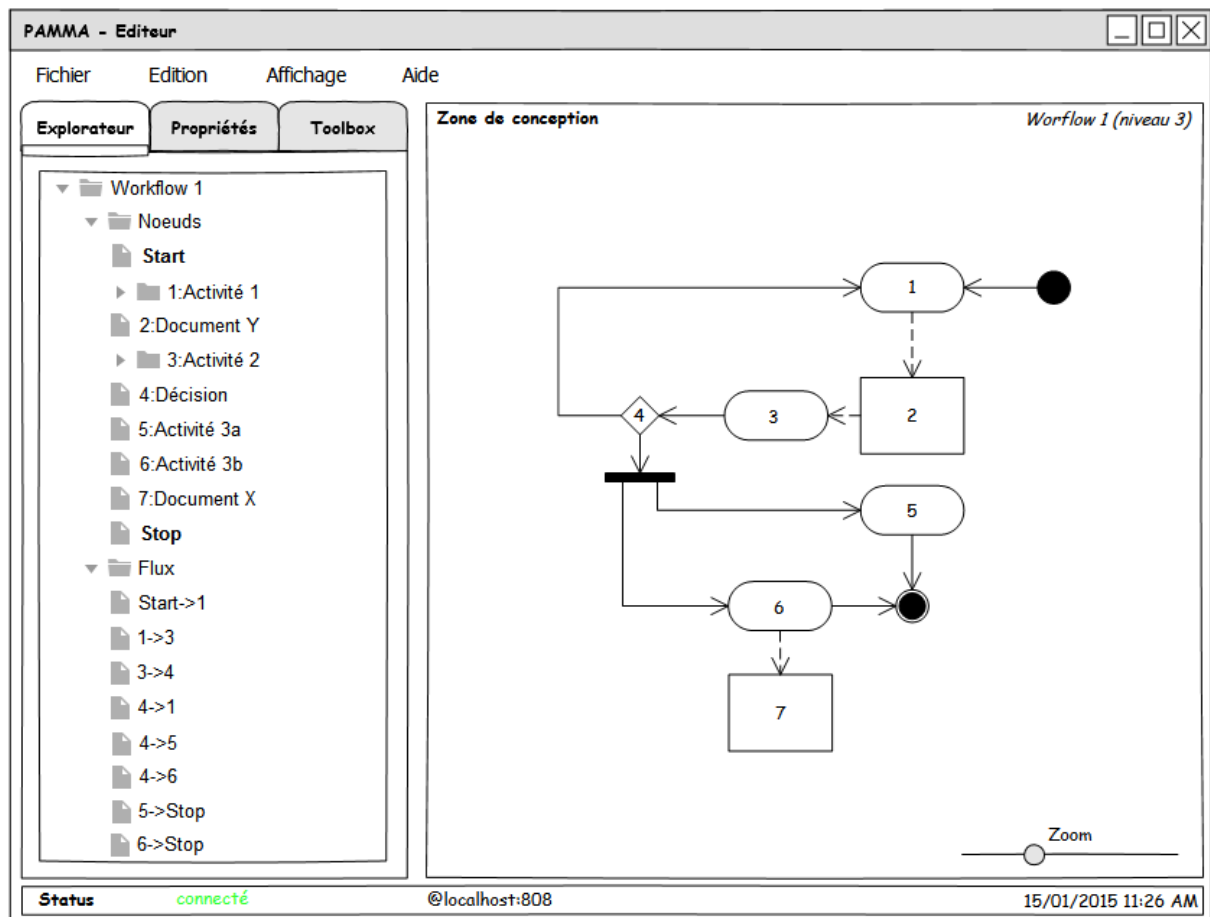


Figure 172 : Prototype de l'IHM de l'éditeur

Pour illustrer ce propos, nous allons maintenant montrer brièvement comment l'éditeur peut permettre de réaliser une application avec l'exemple des grandes lignes de la création d'un nouveau service λ sur une PAMMA. On précisera qu'un tel service est simplement

destiné à être appelé par d'autres applications. Ainsi, tout débute par l'analyse fonctionnelle triviale qui consiste à dire que nous avons besoin d'un workflow comprenant une seule activité qui correspond à la mise à disposition du service (**Figure 173**). Pour cet exemple, nous avons utilisé un diagramme d'états-transitions d'UML (Object Management Group, 2014e). Ce service se lance avec la réalisation de l'activité et se termine avec la fin de celle-ci. Il peut donc être ponctuel, si ce workflow est exécuté à chaque fois qu'on a besoin de lui, ou persistant si l'on considère que l'activité peut se prolonger sur une longue période et que le service va potentiellement être utilisé par différentes applications a priori inconnues.



Figure 173 : Représentation de l'installation d'un service λ au niveau 3 de l'éditeur

Ici, nous adoptons le point de vue d'un service persistant. A ce titre, ce service doit être inscrit aux pages jaunes de la PAMMA afin de renseigner les autres applications et agents de son existence. La **Figure 174** présente des spécifications possibles du SMA, qui permet la réalisation de l'activité de la **Figure 173** en utilisant les notations proposées sur la **Figure 161**, au plus haut niveau de granularité au niveau 2 de l'éditeur. En effet, ce niveau résume le SMA à un seul agent mais celui-ci pourrait être holarchiquement décomposé en d'autres agents de plus bas niveaux de granularité toujours au niveau 2.

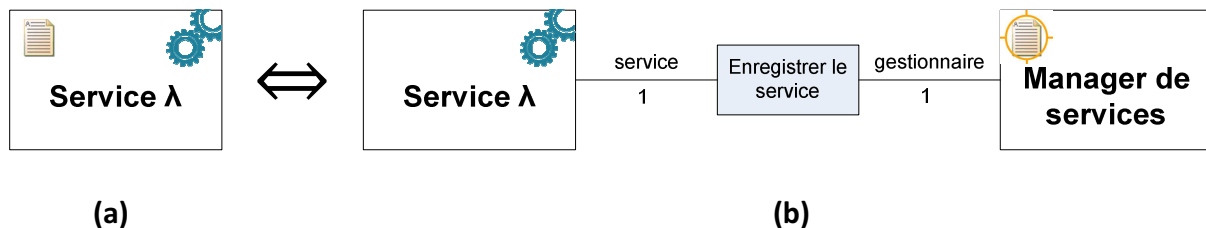


Figure 174 : Représentation d'un SMA assurant un service λ au niveau 2 de l'éditeur

Il est notable que nous avons proposé deux représentations équivalentes à ce SMA. D'une part, nous avons utilisé une notation condensée qui précise que l'agent doit faire l'objet d'une inscription aux pages jaunes (**Figure 174 (a)**), et, d'autre part, nous avons fait apparaître le service de pages jaunes comme étant un service auprès duquel l'agent doit explicitement s'enregistrer via le protocole adéquat (**Figure 174 (b)**). La première de ces représentations (s'appuyant sur la notation de la **Figure 161 (c)**) sous-entend donc tout ce qui a été détaillé sur la seconde. Similairement, on pourrait penser qu'il en est de même avec la notation d'appel à un service externe à l'application (**Figure 161 (d)**). Cependant, ce n'est pas le cas car cette dernière est auto-définie. En effet, comme le montre la **Figure 174 (b)**, le manager de services est tenu de se référencer lui-même (comme il référence d'ailleurs tous les autres services de la PAMMA) : cette notation ne peut donc pas être décomposée et le protocole permettant de contacter le service de pages jaunes doit donc être un axiome pour tous les agents. On notera aussi que le service λ pourrait, à partir d'une notation similaire à celle de la **Figure 174 (b)**, explicitement aller s'inscrire auprès d'un manager distant de services appartenant à une autre PAMMA. Cela n'est pas le cas avec la première notation (**Figure 174 (a)**) qui sous-tend une inscription auprès du service local de pages jaunes. On signalera que chaque application doit exhiber au moins un service qui soit

une interface (graphique ou non). Cette interface doit permettre l'appel direct par l'utilisateur de l'application, ou son appel par une autre application.

Le protocole d'enregistrement auprès du manager de services peut être représenté, comme les workflows au niveau 3, à l'aide d'un diagramme d'états-transitions (**Figure 175**).

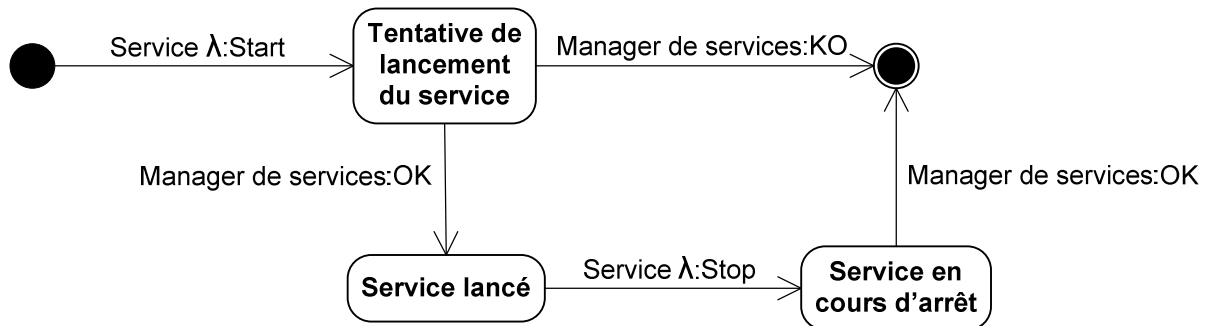


Figure 175 : Représentation de l'enregistrement d'un service λ au niveau 1 de l'éditeur

Ce protocole décrit simplement comment le service va apparaître puis disparaître des pages jaunes. Les types de messages échangés entre le service λ et le manager de services (qu'on pourrait qualifier de performatives/primitives bien qu'assez informelles dans le cas présent) doivent être traitables par ces agents (**Figure 176**). A ce titre leurs implémentations devront, au niveau de leurs gestionnaires de protocoles respectifs (**Tableau 29**), permettre la gestion de ces messages et leur association avec des comportements adaptés.

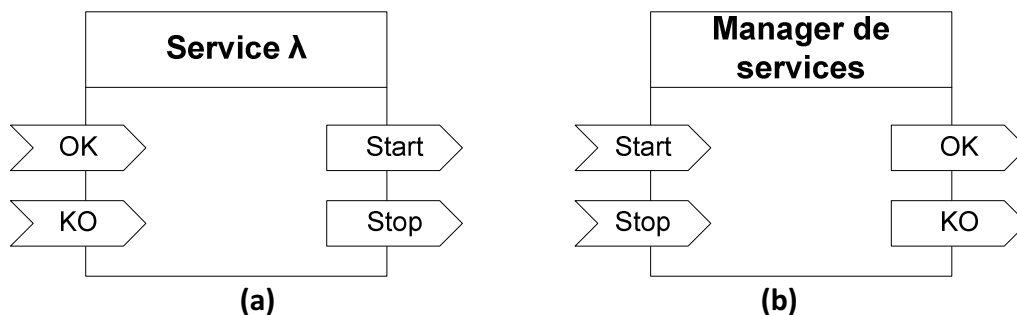


Figure 176 : Représentation abrégée d'un service λ et d'un manager de services au niveau 1 de l'éditeur

Dans le cadre de notre exemple, étant donné que nous n'allons pas réellement implémenter ces agents, ni même définir tous les comportements qu'ils devraient pouvoir exhiber, nous nous contentons ici d'en proposer une représentation abrégée en AUML (Huguet, 2004a)(**Figure 176**).

2) L'architecture détaillée de la PAMMA

Pour conclure cette partie, nous allons passer en revue les différents éléments macro-architecturaux de la PAMMA. Nous apporterons également des précisions supplémentaires concernant certains services de la plate-forme que nous n'avons qu'abordés jusque là.

L'architecture générale de notre PAMMA, telle que nous l'avons décrite, est constituée des composants suivants (**Figure 178**) :

- Un superviseur/runtime, qui permet l'exécution et la gestion des applications décrites sous la forme de workflows. Il se constitue des éléments suivants :
 - Un service de workflows, qui procède à l'exécution des workflows. Ce service comprend également un échéancier. Ces workflows sont les applications.
 - Un manager d'agents, qui est chargé de l'instanciation/destruction des agents et qui assure un service local de pages blanches.
 - Un manager de services, qui liste les services offerts par les agents, locaux ou distants, dans des pages jaunes.
 - Un manager de rôles et de protocoles, dont la fonction est de garantir la conformité des interactions au sein des SMA.
 - Un module sécurité, qui a sous sa coupe les agents des utilisateurs authentifiés. Entre ces derniers, peuvent d'ailleurs s'établir des relations maître-esclave ou pairs, qui les amènent à être collaborateurs, concurrents ou redondants (pour des illustrations de ces modes d'interaction, consulter (Aridor & Lange, 1998; Kook et al., 2011)).
 - Un agent de journalisation et de monitoring du superviseur, des applications/processus instanciés et des messages échangés.
 - Une couche de middleware, comprenant :
 - Un médiateur/facilitateur local, qui permet de gérer les échanges de messages. Ce service peut agir comme un broker et/ou un routeur (comme, par exemple, dans (Rodríguez-Domínguez et al., 2012)).
 - Un manager de données, qui recense :
 - Les utilisateurs de la PAMMA et leurs droits d'accès.
 - Les éléments du framework qui permettent de décrire les applications.
 - Les applications par des modèles à trois niveaux d'abstraction.
 - Des agents d'interfaces vers des services externes de données.
- Un contrôleur réseau, qui permet l'intégration au système de différents matériels ainsi que de différentes PAMMA par le biais des composants suivants :
 - Une couche de middleware, comprenant :
 - Un médiateur/facilitateur, qui permet de gérer les échanges de messages avec les matériels et les PAMMA distantes. Ce service fonctionne comme son homologue local mais permet également de gérer les agents mobiles (comme, par exemple, dans (Rodríguez, Favela, Preciado, & Vizcaíno, 2005)).
 - Un manager des éléments connectés (matériels et PAMMA), qui assure le référencement de ce qui appartient au réseau de la PAMMA.
 - Un agent de journalisation et de monitoring du contrôleur réseau, des tentatives d'accès d'utilisateurs et des messages échangés.

Nous pouvons remarquer qu'on retrouve ce genre de conceptions macro-architecturales aussi bien chez les PAM (Hatcliff et al., 2012) que chez la plate-forme d'agents proposée par FIPA (pour un exemple d'application, consulter (Kook et al., 2011)). Par ailleurs, on signalera que sur notre **Figure 178** chaque élément représente un agent ou un ensemble d'agents.

On pourra également préciser que les PAMMA sont destinées à être distribuées et à former un ensemble d'environnements pour les agents, qui disposent de cette manière d'un réseau d'environnements pour agents mobiles (Green et al., 1997). Ainsi, différentes plates-formes, travaillant de concert, vont permettre la réalisation de SMA distribués et donc décentralisés (Boissier et al., 2004). D'autre part, l'ensemble constitué des bases de données des utilisateurs et des tableaux noirs/blancs des matériels définit l'ensemble des ressources du système. Ces ressources sont vouées à être sollicitées lors de l'exécution des workflows (Casalicchio & Tucci, 2013).

Les éléments constitutifs des applications sont groupés en packages, de la même manière que le sont les éléments du framework (voir la sous-section **L'élaboration du diagramme de paquetage**). Ces packages peuvent d'ailleurs permettre d'exprimer un lien de dépendance entre éléments si, par exemple, certains fournissent un service indispensable au fonctionnement d'autres.

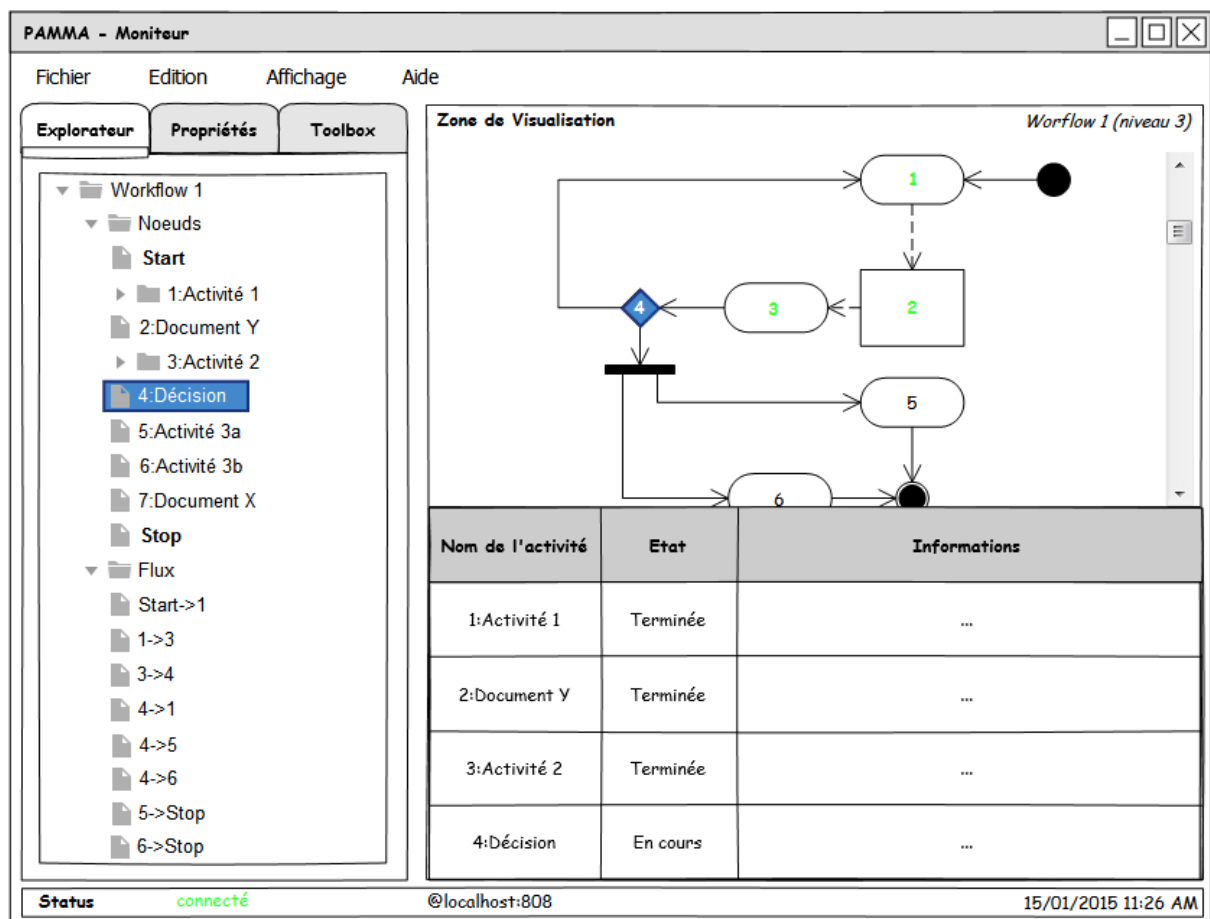


Figure 177 : Prototype de l'IHM du moniteur de workflows

L'échéancier sert à ordonnancer l'exécution d'activités différées par le service de workflows (Miksch et al., 1997) ou, dans le cadre d'une simulation, de planifier les activités simulées (comme, par exemple, dans (Casalicchio & Tucci, 2013)). Ce composant sert aussi de service « horloge », c'est-à-dire qu'il peut retourner la valeur du temps courant d'exécution. On signalera que les ordigrammes ne permettent pas de spécifier ce type de tâches complexes, il faudra donc pouvoir utiliser des formalismes adaptés à ce type de

représentation tels que Asbru (Hammermüller & Miksch, 1999; Miksch et al., 1997; Shahar et al., 1996, 1998).

Dans le cadre de l'exécution d'un workflow, chaque activité doit être scrutée par le service de journalisation et de monitoring afin d'en déduire diverses propriétés telles que le temps nécessaire pour achever l'activité, les différentes ressources utilisées, le temps pris par les ressources pour accomplir les tâches et les temps d'attente des tâches avant traitement par les ressources (Savarimuthu, Purvis, & Fleurke, 2004). Il est ainsi plus aisé d'évaluer les performances des applications au regard de la qualité et du nombre de ressources qui leur sont dévolues. Il est également possible de juger objectivement de l'impact de modifications du workflow. La partie de ce service dédiée aux workflows peut proposer une interface similaire à celle de l'éditeur telle qu'illustrée par la **Figure 177**.

Assurément, on remarque que notre plate-forme est relativement indépendante du contexte d'application médicale. En effet, elle est relativement générique, hormis les interfaces vers des services de données externes, et pourrait très bien être utilisée dans tout autre domaine métier ayant un besoin en termes d'intégration, voire de développement de SMA ou même simplement d'applications.

Pour résumer la contribution de ce chapitre, nous avons commencé par passer en revue différents langages de modélisation étant potentiellement utilisables dans notre éditeur. Ensuite, nous avons réparti ce type de modélisation sur plusieurs niveaux d'abstraction afin de pouvoir utiliser directement des éléments de notre framework MA dans cet éditeur. Pour finir, nous avons expliqué comment des modèles ainsi produits allaient alors pouvoir servir à l'instanciation d'applications sur la PAMMA.

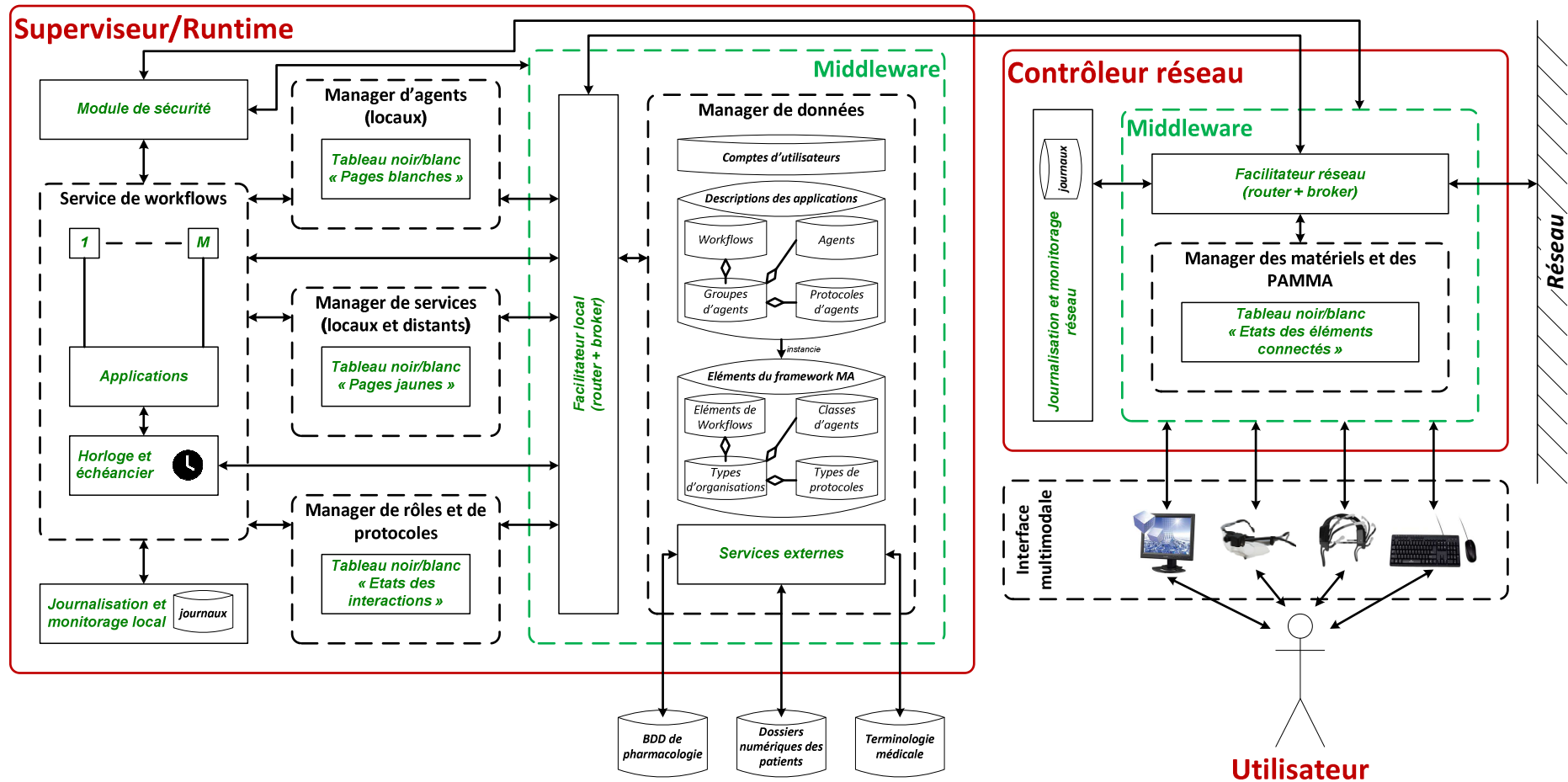


Figure 178 : Schéma global de l'architecture détaillée de la PAMMA

III) L'intégration métier verticale : support de la multimodalité

Dans notre recherche pour que notre système propose des interactions homme-machine plus transparentes, souples et efficaces, nous nous devons de nous pencher sur les possibilités offertes par les interfaces multimodales (voir la section **Les interfaces utilisateur**). En particulier, ces interfaces vont permettre une analyse approfondie, ainsi que menée simultanément sur plusieurs canaux informationnels, du patient qu'on suspecte, dans notre cas, d'être atteint de dyslexie. Effectivement, comme nous l'avons précédemment évoqué (voir la sous-section **Mutualiser nativement des données multimodales**), ces données multimodales peuvent avoir un fort intérêt dans le domaine clinique, ainsi que dans celui de la recherche, entourant cette pathologie neuropsychologique. En particulier, ces données devraient permettre de calculer des métriques objectives conduisant à une amélioration du diagnostic et du suivi de remédiation.

D'autre part, l'intégration à notre système d'une telle interface peut également présenter un intérêt pour les professionnels qui l'utilisent. Notre interface pourrait ainsi être aussi utilisée pour interagir consciemment avec un système informatique, au-delà de l'usage en tant qu'appareil de mesures médicales/paramédicales. Il est important de remarquer que ce genre d'interface se doit d'être direct, compréhensible et prévisible, ce qui, dans le cadre général d'une interaction homme-machine, procure à son utilisateur un sentiment d'accomplissement et de contrôle (Shneiderman & Plaisant, 2010). Sa grande sophistication, inhérente à sa multimodalité, la rend aussi adaptable à de nouveaux usages, ce qui surpasse même le critère de prévisibilité précédemment énoncé (Turk, 1998). Par exemple, on notera que la conception de systèmes de DMP plus efficaces ne doit pas uniquement tenir compte des processus de travail en contexte clinique, mais aussi exploiter les technologies d'interface multimodales émergentes (Weibel et al., 2013). Assurément, cette étape d'intégration verticale est aussi importante que celle d'intégration horizontale (voir le chapitre **L'intégration horizontale : support des processus métiers collaboratifs**). En outre, nous ajoutons que la perspective de la conception de cette interface multimodale comprend celle d'un matériel simple d'utilisation, qu'on peut qualifier de « plug-and-play » (Arney, Fischmeister, Goldman, Lee, & Trausmuth, 2009)).

D'une manière générale, le développement d'interfaces multimodales vise à améliorer les interactions selon deux axes (Oviatt, 2007; Reeves et al., 2004) :

- En les rendant plus naturelles.
- En les consolidant par le biais d'informations complémentaires et/ou redondantes.

Comme toute IHM, notre interface multimodale va se trouver au plus proche des utilisateurs lorsqu'ils utiliseront notre système. Elle va être, de la sorte, amenée à participer directement à la satisfaction des besoins de ses utilisateurs. Ainsi, dans la présente partie, nous allons décrire les besoins fonctionnels que devront satisfaire les composants de notre système, qui permettent la gestion de chacune des modalités constitutives de notre interface multimodale. Nous traiterons aussi brièvement de leurs structures respectives, aussi bien

matérielles/tangibles que logicielles/informationnelles, afin de réfléchir aux avantages et inconvénients qu'elles peuvent apporter ou engendrer lors du futur usage du dispositif. A ce titre, nous rappelons que l'objectif de cette étude est la production de deux prototypes (**PROTO1** et **PROTO2**) ainsi qu'un cahier des charges (**MODEL2**). La réalisation de ces prototypes servent à (Beaudouin-Lafon & Mackay, 2007) :

- Aider à générer des idées et faciliter l'exploration de l'espace de design en encourageant la communication entre les différents acteurs du développement.
- Découvrir des informations pertinentes sur les utilisateurs et leurs pratiques de travail au travers d'évaluations dès lors qu'ils peuvent être testés.

Les résultats suggérés par ces prototypes seront présentés plus loin dans ce manuscrit (voir la partie **L'analyse des résultats et les perspectives**).

Conformément à notre analyse du problème global (voir le chapitre **La division en sous-problèmes**), les modalités sur lesquelles nous travaillerons sont les suivantes (**Figure 179**):

- Le regard de l'utilisateur, pour lequel nous proposerons une réflexion poussée qui devrait, à terme, faciliter l'accès à cette technologie qui, bien qu'efficace, reste encore trop onéreuse dans ses versions actuellement commercialisées.
- Son activité cérébrale, qui, combinée à l'utilisation de nouveaux matériels abordables de **Brain-Computer Interface** fraîchement arrivés sur le marché, ouvre la voie à un diagnostic plus fiable et plus rapide.
- Ses productions verbales, qui sont indissociables d'une communication humaine normale mais que nous n'allons aborder que brièvement, au regard des technologies pour en automatiser l'analyse actuellement disponibles au grand public.

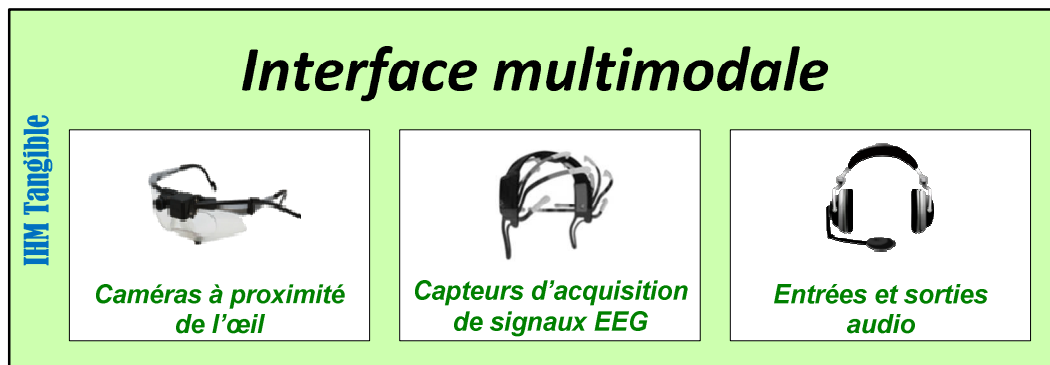


Figure 179 : Schématisation des principaux composants de la partie tangible de notre interface multimodale

Ainsi, dans les chapitres qui vont suivre, nous allons présenter nos recherches sur la conception d'un système de suivi oculaire (voir la section **Le suivi oculaire**) ainsi qu'un système d'analyse des signaux EEG (voir la section **L'électroencéphalographie**). De son côté, la reconnaissance de la parole est une technologie qui est maintenant aboutie et très bon marché (par exemple, inférieure à 100 euros pour une version standard de Dragon NaturallySpeaking (Nuance, 2015)). Il s'agit d'un composant populaire des interfaces multimodales car la parole est une modalité importante de la communication humaine (Turk & Kölsch, 2003). Elle est, effectivement, d'un usage bien plus naturel et flexible que la frappe sur un clavier. Par exemple, elle peut être utilisée dans une approche bimodale, alliant analyse acoustique et optique, afin de rendre la reconnaissance vocale plus fiable et efficace

(pour un exemple d'application MA implémentant cette approche, consulter (Srivastava & Dev, 2013)). L'étude de cette modalité est également bien documentée et des architectures génériques pour sa mise en œuvre sont disponibles dans la littérature (par exemple, consulter (Nguyen, 2003)). Nous n'approfondirons donc pas plus l'étude de la reconnaissance de la parole.

Quelles que soient les modalités incriminées, afin de satisfaire au mieux les exigences fonctionnelles des potentiels utilisateurs, ces interfaces doivent (Lalanne et al., 2009; Oviatt & Cohen, 2000; Reeves et al., 2004) :

- Être dynamiquement adaptables à un large panel d'utilisateurs différents ainsi qu'à des usages variés. Effectivement, il faut permettre à l'utilisateur une interaction optimale en s'adaptant à ses capacités cognitives et physiques, à ses préférences et au contexte d'utilisation.
- Ne pas être intrusives. Assurément, l'utilisateur doit pouvoir disposer de l'entier contrôle sur celles, parmi ses données personnelles, qui entrent dans le système.
- Proposer une redondance des modalités de façon transparente, en entrée comme en sortie. En effet, les fonctionnalités offertes par le système doivent être clairement identifiables, indépendamment de la modalité d'interaction utilisée par l'utilisateur.
- Être tolérantes aux erreurs. Pour des raisons évidentes d'efficacité, il doit être possible, automatiquement ou non, d'utiliser une ou plusieurs modalités pour supporter l'effet souhaité pendant l'interaction.

Ainsi, une interface multimodale complète doit disposer de mécanismes d'adaptation à son utilisateur ainsi que de procédés de fusion des données en provenance de différentes modalités. D'un point de vue logiciel, les mécanismes d'adaptation font partie du profilage intelligent des utilisateurs et, comme précédemment énoncé, nous ne traiterons pas ce sujet (voir la section **La modélisation métier**). Nous aborderons néanmoins la problématique de la fusion de données et en particulier nous évoquerons son intérêt dans les processus d'aide à la décision (voir la section **L'intégration de données multimodales**).

On pourra également préciser que notre analyse de la multimodalité va exclusivement porter sur les modalités d'entrée du point de vue du système (pour un exposé complet sur ce type de modalités, consulter (Bouchet, 2006)). Bien que la multimodalité puisse également s'appliquer aux sorties du système (comme, par exemple, dans (Jovanov et al., 1999)), nous ne nous étendons pas ici sur cet aspect de la discipline. Afin de prendre en considération le futur développement de nos IHM, nous ferons apparaître les interactions et composants interactifs sur des diagrammes de classes (Obrenović & Starčević, 2006). D'autre part, nous signalons que les SMA fournissent une infrastructure essentielle à la coordination des nombreux modules nécessaires aux traitements des différentes modalités (Oviatt, 2007). Généralement, l'intégration et le traitement de ces modalités vont se faire à l'aide des mathématiques, qui sont également fréquemment utilisées pour décrire le comportement d'agents (Treuil et al., 2008). Effectivement, chaque modalité peut être prise en charge à l'aide d'agents d'IHM. En particulier, la propension de ces agents à communiquer de manière asynchrone est bien adaptée à la gestion d'activités parallèles telles qu'observables lors d'une interaction multimodale (Coutaz & Balbo, 1991). De la sorte, nous souhaitons donner, dans le présent chapitre, un aperçu de ce qu'il est possible de faire à l'aide d'agents, qui dans notre framework appartiennent à la Vue et au Contrôleur (voir la section **L'architecture du système**), afin de gérer la multimodalité.

A) Le suivi oculaire

Le suivi oculaire consiste à étudier la direction et la durée d'un regard porté sur un objet ou une région d'intérêt. Il est admis que le regard est révélateur de l'attention visuelle. A ce titre, on peut supposer que si l'on peut suivre les mouvements oculaires d'un individu, alors il est possible de déduire le cheminement de son attention sur une scène qu'il observe (Duchowski, 2007). Ainsi, des informations concernant la façon qu'a cette personne de percevoir cette scène peuvent être recueillies. Un moyen pratique pour déterminer quels mouvements oculaires ont été effectués est d'observer simultanément les changements de position de l'œil de l'observateur au sein de son orbite ainsi que les changements de position de sa tête par rapport à la scène observée. A l'aide des technologies actuelles, une telle observation peut être pratiquée de manière non invasive et révéler rapidement certains dysfonctionnements neurologiques (par exemple, par analyse des saccades (Leigh & Kennard, 2004)). Notre but ici n'est cependant pas de mettre en corrélation des observations avec des déficits cognitifs, d'autant qu'il ne s'agit pas d'une tâche facile (Irwin, 2004), mais de traiter de quelle manière un dispositif de suivi oculaire peut permettre de procéder à de telles observations.

1) Le principe de fonctionnement

L'activité de l'œil, tout comme la façon dont il nous permet de percevoir notre environnement, est complexe. Cependant, pour appréhender le fonctionnement des systèmes de suivi oculaire, il est nécessaire de comprendre l'essentiel de la physiologie du globe oculaire. Ainsi, nous commencerons par une présentation basique du globe oculaire humain et des mouvements oculaires avant d'exposer les techniques modernes employées pour la réalisation du suivi oculaire.

a) L'œil et la vision

Physiologiquement, un globe oculaire humain (**Figure 180**) sert à filtrer les longueurs d'ondes, ainsi que la quantité d'ondes lumineuses à laquelle sont soumis des photorécepteurs couvrant la rétine de telle sorte que le spectre visible d'une personne normale soit compris entre 380 et 780 nm. Ces photorécepteurs absorbent et convertissent alors ces ondes lumineuses en signaux électriques qui sont acheminés au cerveau par l'intermédiaire du nerf optique auquel ils sont tous connectés (Duchowski, 2007). Il existe deux types de photorécepteurs : les cônes et les bâtonnets. Les cônes, qui sont minoritaires, sont responsables de la vision diurne et de la vision des couleurs grâce à leur grande acuité. Les bâtonnets, quant à eux, sont utilisés pour la vision nocturne grâce à leur grande sensibilité à la lumière et leur capacité d'adaptation. Néanmoins, on signalera l'existence d'une zone aveugle sur la rétine, c'est-à-dire dépourvue de photorécepteurs, correspondant à l'endroit où le nerf optique sort du globe oculaire.

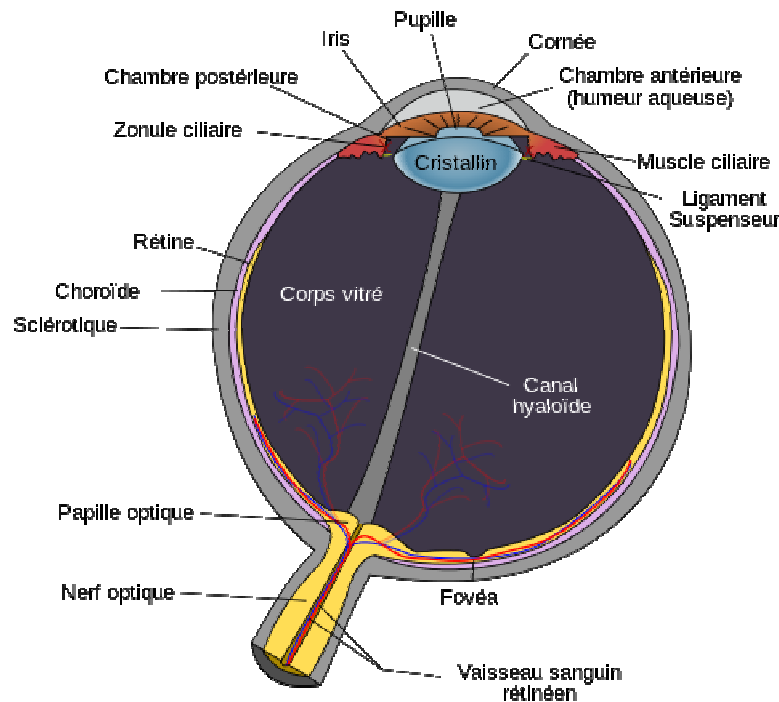


Figure 180 : Schéma d'un globe oculaire humain (source (Wikimedia, 2015))

Le fonctionnement général du globe oculaire est le suivant : la lumière perçue va ainsi pénétrer l'œil par la cornée, traverser l'humeur aqueuse, passer par la pupille qui siège au centre de l'iris, être concentrée par le cristallin qui agit comme une lentille et finalement être projetée sur la rétine après avoir transité par le corps vitré. Pour distinguer clairement un objet ou une région d'intérêt, une personne doit déplacer son œil de sorte que cet objet apparaisse sur sa fovéa, une petite zone d'un diamètre d'environ 1500 μm au centre de la rétine qui est composée principalement de cônes (**Figure 180**). Pour cette raison, la position des yeux de l'observateur donne une assez bonne indication, à l'intérieur du cône défini par la fovéa dans le champ visuel, de la partie spécifique d'une scène qu'il observe (Jacob, 1991). La fovéa est sollicitée par les activités qui nécessitent un haut niveau de détails telles que la lecture (Clifton, Staub, & Rayner, 2007; Rayner, 1998). La partie du champ visuel qui lui est associée correspond à une acuité maximum (**Figure 182**) et est appelée « vision centrale ». Le reste du champ visuel est qualifié de « vision périphérique ». On signalera cependant que le champ visuel n'est pas centré sur l'axe optique, qui passe par le centre de la cornée et le centre de la pupille, mais sur l'axe visuel, qui passe par le centre de la cornée et le centre de la fovéa (**Figure 181**). Il existe ainsi un angle, fréquemment noté α , entre ces deux axes qu'il faudra calculer. Bien que variant d'un individu à l'autre, cet angle peut être supposé d'environ 4-6° (Villanueva, Cerrolaza, & Cabeza, 2007). Alors que l'axe optique est géométriquement identifiable à partir d'une image externe de l'œil, l'axe visuel, qui supporte le vecteur du regard, ne peut être apprécié qu'à partir de l'écart observé entre une région d'intérêt, sur laquelle l'observateur est sensé porter son attention, et l'endroit sur lequel se projette l'axe optique.

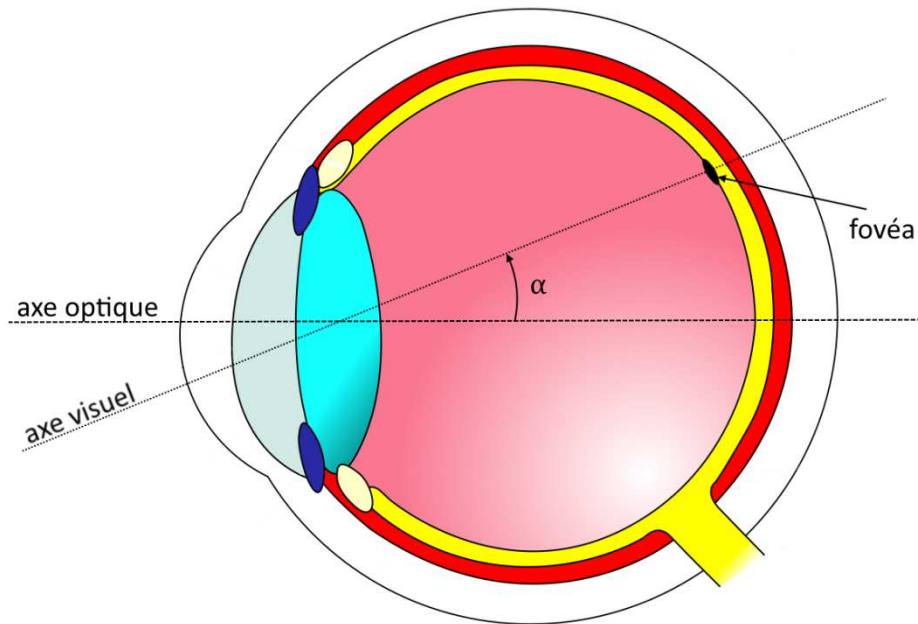


Figure 181 : La différence entre l'axe visuel et l'axe optique de l'œil humain (à partir d'une source (Wikimedia, 2015))

L'orientation du champ visuel est modifiée par la position de la tête et/ou de l'œil pour déplacer l'attention de l'observateur (Duchowski, 2007). En particulier, le globe oculaire peut être repositionné par l'intermédiaire de mouvements oculaires dont sont responsables six muscles. Principalement, ces mouvements peuvent être catalogués en trois catégories :

- Les saccades, qui servent à repositionner très rapidement le regard, conformément aux changements de cibles de l'attention de l'observateur (Irwin, 2004), dans une nouvelle direction stationnaire afin d'obtenir la vision optimale d'un objet ou d'une région d'intérêt. Elles prennent couramment de 20 à 100 ms, avec un pic de vélocité angulaire aux alentours de $800^\circ/s$, et leur latence, c'est-à-dire le délai dû aux traitements cognitifs avant une saccade, prend de 200 à 250 ms (<100 ms dans le cas de « saccades express »)(Duchowski, 2007). Parmi les saccades, on peut distinguer celles qui sont spontanées, réflexes, express, volontaires, mémorisées, anticipées et les anti-saccades (Leigh & Kennard, 2004). Comme leurs dénominations l'indiquent certains de ces mouvements sont hors du contrôle de l'observateur. On peut également signaler qu'environ la moitié des saccades se termine par un mouvement oscillatoire qu'on appelle « glissade » qui correspond à la stabilisation de l'œil avant une fixation (Nyström & Holmqvist, 2010).
- Les fixations, dont le but est de permettre l'acquisition de l'information comprise dans le champ visuel (Irwin, 2004). Elles prennent entre 150 et 600 ms, et en moyenne environ 350 ms (Noureddin, Lawrence, & Man, 2005), et elles incluent des tremblements (aussi désignés par le terme de « nystagmus physiologique »), micro-saccades et dérives lentes qui sont, entre autres, indispensables au maintien d'une excitation neuronale permettant la perception (Martinez-Conde, Macknik, & Hubel, 2004). On précisera que ces mouvements sont indépendants de la volonté de l'observateur et que ce dernier ne se rend même pas compte qu'ils ont lieu. Une fixation d'un seul œil offre un champ de vision approximativement elliptique, d'environ 180° en largeur et 130° en hauteur, de telle sorte que le champ visuel binoculaire couvre une superficie d'environ $(23\ 400)^\circ$. L'information ainsi extraite n'est pas limitée au point de regard, c'est-à-dire au point précis de la scène sur lequel

passé le regard, mais s'étend également autour de ce point de fixation (Irwin, 2004). La quantité d'informations ainsi récupérée dépend de paramètres environnementaux, tels que la nature de ce qui est observé ou la luminosité, ainsi que d'autres propres à l'observateur, comme son âge ou sa connaissance de l'objet de son observation.

- La poursuite lisse, qui correspond à un mouvement régulier de l'œil permettant de suivre un objet en mouvement. Il y a derrière ce type de mouvements la même finalité que dans le cas des saccades, à l'exception près que la cible du regard n'est ici pas statique (Orban de Xivry & Lefèvre, 2007). Il est possible d'analyser l'initiation d'une poursuite, c'est-à-dire sa latence par rapport à la mise en mouvement de sa cible, ainsi que sa trajectoire et son gain, c'est-à-dire le rapport entre sa vitesse et celle de sa cible. La vélocité angulaire maximale d'une poursuite lisse est d'environ 90°/s pour un mouvement prévisible et de 40°/s sinon (Cuba Gyllensten, 2014; Komogortsev & Karpov, 2013).

On signalera que l'attention de l'observateur n'est maintenue que pendant les fixations et les poursuites lisses alors qu'un phénomène de suppression apparaît lors des saccades (Duchowski, 2007; Rayner, 1998). On pourra aussi signaler l'existence d'autres mouvements oculaires tels que la vergence, la rotation de torsion ou le réflexe oculo-vestibulaire.

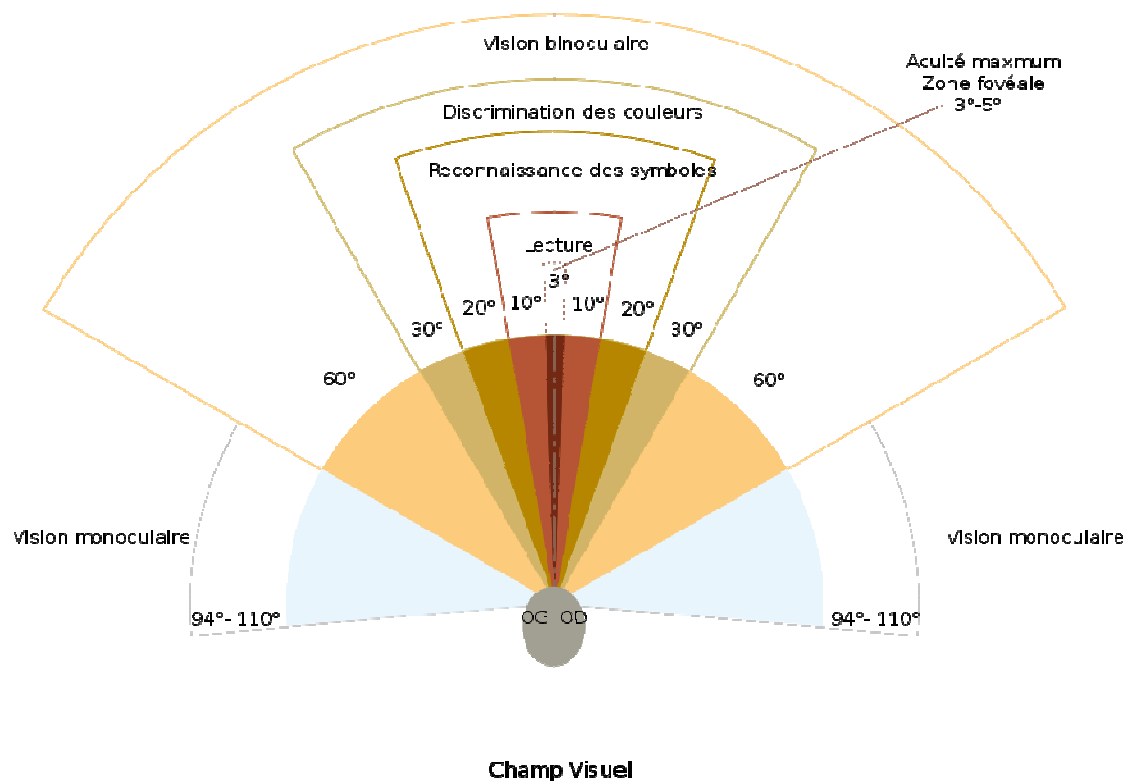


Figure 182 : Champ visuel humain dans le plan horizontal (source (Wikimedia, 2015))

Afin d'étudier objectivement la perception et l'attention visuelles (Rayner, 1998), il existe des systèmes de suivi oculaire permettant l'identification et l'analyse automatique des mouvements de l'œil. Parmi ces dispositifs, les plus précis et les moins invasifs sont ceux ayant recouru à des méthodes d'oculographie par photo ou vidéo (Duchowski, 2007). De

telles méthodes reposent sur la mise en place d'algorithmes d'analyse d'images afin d'extraire les caractéristiques de l'œil. Il existe d'ailleurs différents types de techniques algorithmiques permettant de cette manière de réaliser la détection de l'œil et le suivi du regard. Avant de passer en revue celles qui existent, nous signalons que leur efficacité est limitée par leur fréquence de traitement des images ainsi que la définition utilisée à l'acquisition de ces dernières. En effet, plus un mouvement de l'œil est rapide et/ou de faible amplitude, plus les paramètres précités devront être élevés pour pouvoir arriver à sa détection effective.

b) Les techniques existantes

Nous ne ferons pas ici une revue historique des techniques de suivi oculaire (pour une telle revue, consulter (Kassner & Patera, 2012)) mais nous nous efforcerons de donner un panorama des techniques vidéo-oculographiques qui sont utilisées à l'heure actuelle. Ce type de méthodes permet d'accéder à deux sortes d'informations (Hansen & Ji, 2010) :

- La localisation de l'œil, c'est-à-dire la détection de l'œil et de ses caractéristiques qui consiste à :
 - o Etablir l'existence de l'œil
 - o Interpréter sa position
 - o Suivre l'œil au cours de ses mouvements
- L'estimation du regard, qui revient à effectuer la poursuite du point de regard de l'observateur par le calcul d'un vecteur de regard.

D'un point de vue technique, l'estimation du regard peut être réalisée à l'aide de l'observation de l'œil de l'observateur, de sa tête ou d'une combinaison des deux. Dans le cas où seul l'œil est utilisé, il est possible d'estimer correctement le regard si seulement la tête de l'observateur est immobilisée. Si la tête est utilisée seule, alors il n'est possible que d'obtenir une estimation très grossière du vecteur de regard que nous pourrions qualifier de pseudo-vecteur de regard. Les techniques les moins contraignantes fonctionnellement et les plus précises sont celles combinant une analyse de la position de l'œil et une autre de la position de la tête de l'observateur.

D'un point de vue du matériel, il existe deux types de systèmes : ceux portés sur la tête de l'utilisateur (pour un exemple, consulter (Wagner et al., 2006)) et ceux qui effectuent un suivi oculaire à distance (pour un exemple, consulter (Ji & Zhu, 2002)). Les premiers sont plus intrusifs mais plus précis, et plus souples d'utilisation, que les seconds. En effet, les systèmes de suivi à distance ne sont efficaces que dans une région de l'espace restreinte se situant face à l'appareil de mesure. Les systèmes montés sur la tête de l'utilisateur semblent offrir un meilleur compromis en termes de liberté de mouvement et, plus généralement, d'utilisabilité (Allison, Eizenman, & Cheung, 1996; Li, Babcock, & Parkhurst, 2006). En outre, les images capturées peuvent l'être en lumière naturelle ou sous un éclairage InfraRouge actif (qui est utilisée dans de très nombreux exemples (Campbell & Maglio, 2001; Coutinho & Morimoto, 2006; Jacob, 1991; Li, Winfield, & Parkhurst, 2005; Murawski & Rózanowski, 2013; Schneider, Bex, Barth, & Dorr, 2011; Świrski et al., 2012)). L'utilisation d'une source IR active a pour avantage de ne pas perturber l'utilisateur car il s'agit de fréquences hors du spectre de la lumière visible qui sont, cependant, perceptibles par la grande majorité des

caméras. Il existe deux manières de placer une source IR par rapport à la caméra (Hansen & Ji, 2010) :

- « on-axis », si la source IR est dans l'axe de la caméra, ce qui a pour effet de rendre la pupille brillante. Avec cette technique, la pupille contraste très fortement du reste de l'image.
- « off-axis », si la source IR est hors de l'axe de la caméra, ce qui permet de discerner une pupille noire sur laquelle se forme le reflet de la source IR. Ce reflet peut en réalité être observé sur différentes couches de la structure l'œil (Chi, Zhang, Zheng, Zhang, & Huang, 2009) : on parle alors « d'images de Purkinje » (pour un exemple d'application à l'aide de deux de ces images, consulter (Crane & Steele, 1985)). Il est également possible d'utiliser plusieurs sources actives au lieu d'une seule (pour des exemples, consulter (Coutinho & Morimoto, 2006; Pérez et al., 2003)).

Néanmoins, on remarquera que l'utilisation de ces solutions est inapplicable en extérieur à cause du parasitage provoqué par les IR de la lumière naturelle, et qu'on ne connaît pas les conséquences de l'usage prolongé d'IR sur l'œil (Asteriadis, Tzouveli, Karpouzis, & Kollias, 2007). Les solutions utilisant une ou plusieurs sources IR actives sont donc loin d'être la panacée.

Quelle que soit la nature de l'information recherchée, il faut avoir recourt à une méthode de modélisation de l'œil (Zhiwei Zhu & Ji, 2005). Parmi de telles méthodes, on distingue principalement celles qui reposent sur des modèles préétablis ou sur la détection de caractéristiques (Sheela & Vijaya, 2011). Cette modélisation doit pouvoir mener à une localisation relativement précise de l'œil et à en extraire les informations nécessaires pour établir la direction du regard. Hansen et Ji (2010) proposent d'ailleurs la classification des méthodes de détection plus précise suivante :

- Les méthodes basées sur l'apparence, qui cherchent à établir une correspondance avec un gabarit global/holistique après un calibrage. Ces méthodes peuvent reposer sur une analyse d'images basée sur l'intensité ou un sous-espace.
- Les méthodes basées sur la forme, qui ont recours à l'utilisation de gabarits préalables (Hansen & Pece, 2005). On peut distinguer celles utilisant des gabarits à forme fixe de celles en utilisant à forme déformable.
- Les méthodes basées sur les caractéristiques de l'œil ou d'une partie du visage, c'est-à-dire qui procèdent au repérage de points ou zones particulières (par exemple, consulter (De Santis & Iacoviello, 2009a)).

Il faudra néanmoins garder en mémoire que cette classification ne définit qu'un framework général et, à ce titre, elle ne peut pas classer toutes les techniques existantes. Effectivement, certaines techniques, telles que celles utilisant des opérateurs de symétrie, n'y trouvent pas leur place (Hansen & Ji, 2010). En outre, bien des méthodes adoptent des approches hybrides, c'est-à-dire qu'elles peuvent relever simultanément de plusieurs de ces méthodes.

Consécutivement à cette phase de détection de l'œil, et de son éventuelle caractérisation, il devient possible d'estimer le regard. Cette estimation consiste à déterminer la direction du regard ou directement le point du regard. Cette étude sera l'occasion d'aborder la problématique du calibrage des différents types de système de suivi oculaire.

Nous allons maintenant présenter brièvement les aspects techniques de ces différents genres de méthodes et préciser leurs avantages et inconvénients d'un point de vue fonctionnel.

1. Les méthodes de détection basées sur l'apparence

Ces méthodes consistent à utiliser l'apparence photométrique globale de l'œil pour identifier la direction du regard à partir d'une base d'apparences de référence pour lesquelles la direction du regard est connue. Pour cela, elles peuvent analyser l'image des manières suivantes :

- A partir de la distribution des couleurs, c'est-à-dire qu'elles utilisent le contraste afin de différencier les zones de l'œil.
- A partir des réponses à des filtres de l'œil et de ses abords, c'est-à-dire qu'elles appliquent des gabarits (par exemple, les caractéristiques de Haar qui sont utilisées dans (Świrski et al., 2012)) sur des images de l'œil dont les résultats sont ensuite rangés à l'aide de classificateurs.

On remarquera que ces techniques supportent les changements de luminosité, mais qu'elles rencontrent des problèmes lors de changements d'échelle et rotations. Elles nécessitent également d'être entraînées sur de grandes quantités de données. Bien qu'obtenant de bonnes performances en terme de temps de calcul, ces méthodes produisent des résultats de qualité souvent mitigée (Hansen & Ji, 2010).

2. Les méthodes de détection basées sur la forme

Cette deuxième classe de méthodes cherche à délimiter précisément une portion de l'œil dont le pourtour anatomique repose sur un modèle connu. A ce titre, ces techniques peuvent être amenées à utiliser l'iris, la pupille ou carrément le contour visible de l'œil. Habituellement, elles reposent sur deux éléments principaux (Hansen & Ji, 2010) :

- Un modèle géométrique de l'œil, qui peut aller d'une simple ellipse à une modélisation bien plus complexe. Les paramètres de ce modèle définissent quelles déformations lui sont autorisées.
- Une mesure de similarité, qui permet d'établir les valeurs des paramètres du modèle à partir d'images de l'œil.

Dans le cas de modèles géométriques simples, l'identification de la forme peut reposer sur des approches par votes (par exemple, à l'aide de transformations de Hough (pour une telle méthode généralisée et insensible à l'orientation ou l'échelle, consulter (Yip, Tam, & Leung, 1995))) ou par ajustement de formes (par exemple, par ajustement d'ellipses (Kohlbecher et al., 2008)). Les modèles géométriques plus complexes sont des gabarits visant à capturer les différentes caractéristiques morphologiques de l'œil (par exemple, consulter (Lam & Yan, 1996)).

Globalement, ces techniques sont relativement robustes aux changements de formes, d'échelles et aux rotations, et elles s'affichent comme des méthodes génériques et généralement précises. Néanmoins, d'une manière générale, elles sont sensibles au

contraste, elles sont gourmandes en calculs, elles requièrent une prise d'images proche de l'œil et elles gèrent mal les occlusions (Hansen & Pece, 2005).

3. Les méthodes de détection basées sur les caractéristiques oculaires

Naturellement, ce dernier type de méthodes utilise des propriétés anatomiques de parties de l'œil, telles que le limbe cornéen, c'est-à-dire la frontière entre la cornée et la sclérotique (**Figure 180**), la pupille ou la réflexion cornéenne. Comme pour les méthodes basées sur l'apparence, ces méthodes peuvent procéder à des analyses basées sur :

- L'intensité, afin de détecter par contraste différentes régions de l'œil (pour un exemple utilisant ainsi les niveaux de gris, consulter (De Santis & Iacoviello, 2006, 2009a)).
- Les réponses à des filtres, c'est-à-dire le renforcement de certaines caractéristiques au dépend d'autres (à l'aide, par exemple, de filtres de Gabor qui sont utilisés dans (Chen & Kubo, 2007)). On notera que les filtres non-linéaires semblent donner de meilleurs résultats que ceux qui sont linéaires (Hansen & Ji, 2010).

Néanmoins, ces analyses seront effectuées localement et non d'une manière globale, comme le sont celles des techniques basées sur l'apparence. On remarquera également que ces techniques sont sensibles à l'éclairage, qu'elles requièrent des images de haute qualité et qu'elles gèrent mal les occlusions.

4. Les méthodes d'estimation du regard

L'estimation du regard est une opération qui consiste à déduire la direction du regard à partir d'informations relatives à l'œil. Dans le cas des techniques de vidéo-oculographie, ces informations peuvent prendre la forme de caractéristiques oculaires ou des points d'intérêt établis à l'aide d'images de l'œil. Le regard peut ensuite être déduit comme étant une fonction prenant ces informations en entrée. Cette relation entre le regard et l'image peut être établie de différentes manières (Cerroloza, Villanueva, & Cabeza, 2008).

Les méthodes, qui s'appuient sur la forme et/ou les caractéristiques de tout ou partie de l'œil, peuvent mettre différentes approches à profit pour déterminer cette relation (Hansen & Ji, 2010). D'un côté, des méthodes de mise en correspondance utilisent des expressions polynômiales (Stampe, 1993) afin d'établir les coordonnées du point observé en fonction des caractéristiques de l'image et un ensemble de coefficients inconnus qui sont déduits lors d'un calibrage (par exemple, en mettant en œuvre des régressions linéaires 2D (Morimoto & Mimica, 2005) ou pour estimer par stéréoscopie l'angle α (**Figure 181**) à l'aide d'un unique point (Nagamatsu, Kamahara, & Tanaka, 2008)). En résumé, ces méthodes créent simplement une « carte » entre les valeurs de ces caractéristiques et les coordonnées de l'écran. De l'autre côté, des modèles reposant sur la géométrie, en 3D, du système peuvent exprimer le regard en fonction de la configuration du système et de l'observateur. A ce titre, ces modèles requièrent également qu'un calibrage de la géométrie du système soit réalisé pour déterminer (Hansen & Ji, 2010) :

- Les paramètres intrinsèques de la caméra (Isern González, Cabrera Gámez, Hernández Sosa, & Domínguez Brito, 2007)).
- Les positions et orientations relatives des différents éléments (caméra, écran, source de lumière, œil, etc.).
- La courbure cornéenne et le décalage angulaire entre les axes visuel et optique (comme, par exemple, dans le système 3250R utilisé par Jacob en 1991 ou avec l'EMR-8 utilisé Miyoshi et Murata en 2001). Ces paramètres intrinsèques de l'œil sont considérés comme fixes pendant une session.

Il devient ensuite possible d'estimer la direction du regard au sein de l'environnement tridimensionnel ainsi modélisé. Si les paramètres relatifs au matériel et à son installation sont connus, on parlera de système « complètement calibré ».

Dans un cas comme dans l'autre, ces étapes de calibrage doivent être réalisées pour chaque utilisateur avant chaque usage, sauf en ce qui concerne les paramètres intrinsèques de la caméra qui ne nécessitent que d'être évalués une unique fois. Il existe des techniques permettant de s'en affranchir mais elles ne permettent que de procéder à un suivi oculaire d'une précision relativement faible (par exemple, consulter (Ji & Zhu, 2002)). Effectivement, il est indispensable d'effectuer au moins un calibrage afin de déterminer l'angle α , entre l'axe optique et l'axe visuel, qu'il n'est pas possible de déduire d'une observation externe de l'œil seul. Si aucun calibrage n'est réalisé, le système produira au minimum une erreur de mesure de l'ordre de cet angle α , c'est-à-dire d'environ 4-6° chez un individu normal (Villanueva et al., 2007).

Dans le cas des méthodes de détection de l'œil basées sur l'apparence, on signalera qu'elles permettent de s'affranchir d'un calibrage géométrique (Hansen & Ji, 2010). Effectivement, les systèmes qui emploient ces méthodes s'entraînent directement sur des images de l'œil, indépendamment de leur niveau de qualité, pour en extraire implicitement les caractéristiques et la géométrie qu'elles associent directement avec des coordonnées du point de regard. A ce titre, leur apprentissage repose sur un grand nombre de points de calibrage. On pourra remarquer que ces méthodes sont sensibles aux mouvements de la tête et à la réfraction produite par le port de lunette ou l'œil lui-même.

2) Les utilisations possibles

Initialement, les dispositifs de suivi oculaire ont été mis au point comme des outils d'étude objective pour la psychologie (Kassner & Patera, 2012). En particulier, ils ont permis de commencer l'observation précise des mouvements oculaires pendant la lecture (Clifton et al., 2007; Rayner, 1998), plutôt que de se contenter des retours verbaux des observateurs tels que pratiqués jusqu'alors. Ces appareils ont ensuite été améliorés afin de pouvoir être utilisés sur des tâches pour lesquelles les mouvements des yeux sont moins linéaires que dans les cas de la lecture, comme, par exemple, dans le cas de l'observation d'images avec ou sans instructions préalables. Ces technologies consistaient à enregistrer les mouvements oculaires estimés sur du film photographique, à reconstituer à partir de tels enregistrements un histogramme de l'évolution de la position du point de regard de l'observateur sur l'objet de son observation puis à finalement analyser statistiquement cet histogramme pour en déduire des traits comportementaux relatifs à la technique individuelle mise en œuvre

pendant l'observation. Les analyses des résultats individuels ont également permis d'étudier les différences inter-individus.

De nos jours, les dispositifs de suivi oculaire ne sont plus uniquement utilisés à des fins médicales et paramédicales mais également, par exemple, pour effectuer des études en stratégie commerciale ou encore afin d'être exploités en tant que dispositif d'IHM (Duchowski, 2002; Mele & Federici, 2012b). Ils sont, à ce titre, de plus en plus employés dans des systèmes interactifs (Morimoto & Mimica, 2005) afin d'inférer, à partir d'un comportement oculaire observé, l'état cognitif de l'utilisateur comme, par exemple, son niveau de fatigue ou sa réceptivité lors d'un échange interpersonnel (Al-Rahayfeh & Faezipour, 2013). Cependant, l'ergonomie, la fiabilité et le prix des systèmes actuels de suivi oculaire expliquent largement leur sous-utilisation hors du domaine de l'ophtalmologie (San Agustin Lopez, 2009). C'est pourquoi, des projets visant à dépasser ces inconvénients et à démocratiser l'utilisation de ces systèmes ont vu le jour. On pourra signaler, par exemple, l'existence des projets openEyes (Babcock & Pelz, 2004; Li et al., 2006, 2005), ITU Gaze Tracker (San Agustin Lopez, 2009; San Agustin Lopez et al., 2010) et Pupil Project (Kassner, Patera, & Bulling, 2014; Kassner & Patera, 2012).

La **Figure 183**, qui s'appuie sur le méta-modèle de description des modalités d'IHM que nous avons retenu (**Figure 15**) (Obrenović & Starčević, 2004, 2006), propose une modélisation UML générique des éléments importants de la modalité de suivi oculaire par vidéographie.

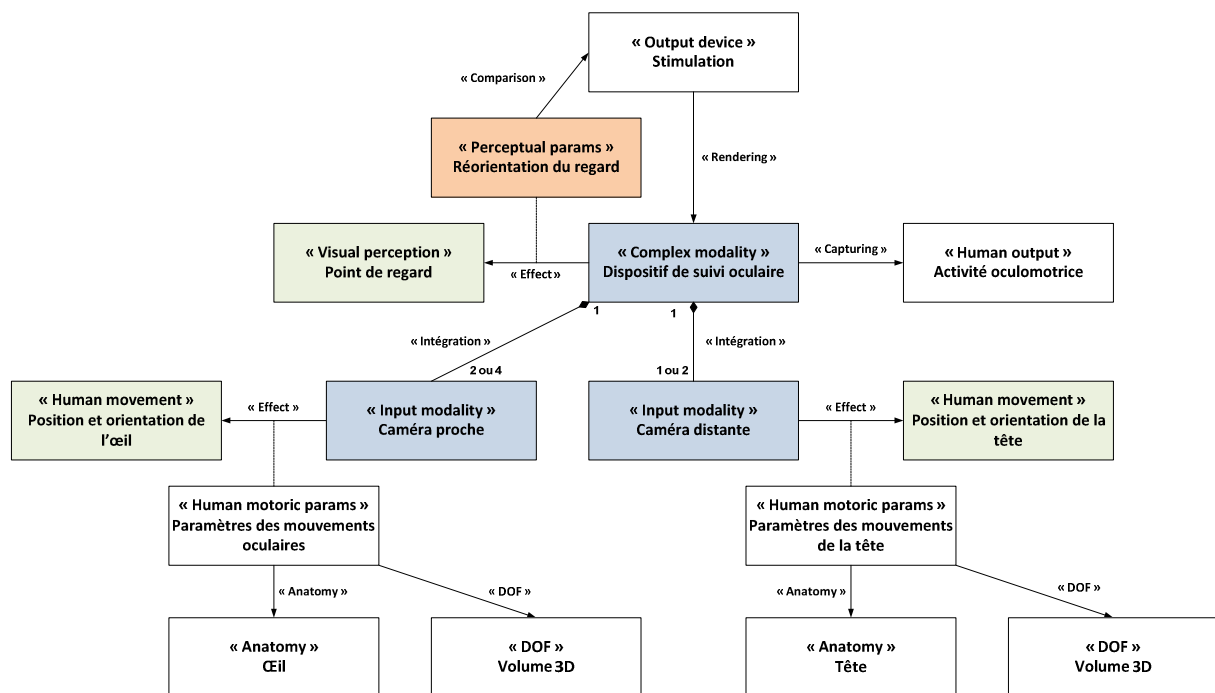


Figure 183 : Diagramme de classe de la description de la modalité de suivi oculaire

a) Un appareil de mesures pour l'analyse de la perception visuelle

Dans le cadre de l'aide diagnostique et du suivi de programmes de remédiation que nous voulons apporter aux patients souffrant de troubles neuropsychologiques, en particulier

ceux atteints de dyslexie, nous avons besoin de réaliser le suivi des mouvements oculaires de l'utilisateur afin de pouvoir recueillir des données relatives au comportement de ce dernier. Effectivement, les déficits neurologiques dont sont atteintes ces personnes peuvent se manifester au travers de modifications des mouvements oculaires par rapport à des sujets normaux. De tels dispositifs sont, ainsi, déjà utilisés dans le dépistage/diagnostic de la dyslexie (Al-Edaily, Al-Wabil, & Al-Ohali, 2013a; Novák et al., 2004) ou d'autres troubles neuropsychologiques, tels que la maladie de Parkinson (Cubizolle, 2012). En outre, notre système doit pouvoir être utilisé en tant que psycho-technologie, c'est-à-dire qu'il doit pouvoir être utilisé pour émuler, étendre, amplifier ou modifier les fonctions sensorimoteurs, psychologiques ou cognitives (Mele & Federici, 2012a). En effet, ces capacités lui permettent d'être utilisé pour une remédiation qui peut s'effectuer selon les deux axes de la rééducation cognitive et la compensation fonctionnelle (par exemple, consulter (Biedert, Buscher, & Dengel, 2009)). Néanmoins, on remarque que les appareils ainsi utilisés sont bien souvent encombrants et onéreux, et leur accès est a fortiori limité (Schneider, 2011).

De telles méthodes de suivi oculaire sont utilisées depuis la fin des années 70 afin d'étudier en temps réel les processus cognitifs (Mele & Federici, 2012a) et, en particulier, ceux impliqués dans la lecture (par exemple, consulter (Sparrow & Miellet, 2002)). La technique de suivi du regard, basée sur de l'analyse vidéo, est utile pour analyser le traitement des cibles visuelles, par exemple, lors de tâches de lecture (Campbell & Maglio, 2001). Par ailleurs, dans le cas où la tête de l'observateur est maintenue par une mentonnière, elle permet de réduire drastiquement les interférences dues aux mouvements de la tête ou du corps lors d'une performance. Il est également possible de combiner une analyse d'images de l'œil avec des images de la scène observée afin d'étudier la perception visuelle dans des conditions moins contrôlées (pour des exemples d'applications, consulter (Kassner & Patera, 2012; Yu & Eizenman, 2004)). D'ailleurs, de nombreuses études expérimentales ont montré une étroite relation entre les mouvements oculaires survenant au cours de tâches d'attention visuelle et la façon dont l'information est traitée au niveau cognitif (Duchowski, 2002; Rayner, 1998). Les tâches de lecture, auxquelles nous nous intéressons, peuvent donc être analysées d'un point de vue comportemental, via le dispositif de suivi oculaire, afin de produire des informations relatives aux processus cognitifs qui sous-tendent les performances lors de ces mêmes tâches. Par exemple, les rétro-saccades ont été liées à une faiblesse de la mémoire à court terme, et la durée des fixations a été liée à la facilité ou la difficulté de traitement de l'élément cible (Al-Wabil & Al-Sheaha, 2010).

Dans le cas de la dyslexie développementale, un déficit consécutif à des capacités visuo-attentionnelles amoindries doit également pouvoir être observé (Bosse, Tainturier, & Valdois, 2007; Bosse & Valdois, 2009; Prado et al., 2007). Il a d'ailleurs été montré que l'on peut ainsi observer des mouvements oculaires caractéristiques de cette pathologie (Bucci, Nassibi, Gerard, Bui-Quoc, & Seassau, 2012; Schulte-Körne & Bruder, 2010). Ainsi, pour l'étude de celle-ci, cette technique est utilisée dans l'analyse de la perception visuelle lors de tâches verbales et/ou non-verbales (par exemple, consulter (Kim, Lombardino, Cowles, & Altmann, 2014)). Cette technique a de plus pour avantage d'être maintenant non-invasive et aisément combinable/couplable avec d'autres dispositifs d'analyse, comme par exemple l'EEG (Mele & Federici, 2012b). Chez les dyslexiques, on observe, comme d'une manière générale chez les mauvais lecteurs, un nombre important de rétro-saccades ainsi qu'un plus

grand nombre de fixations que chez les normaux lecteurs (Macaš, 2005; Rayner, 1998). On précisera que ce constat semble être une conséquence du déficit en lecture et non sa cause.

Pour des applications médicales et/ou paramédicales, ce type de dispositif doit jouir d'un haut niveau de performances. Effectivement, la précision avec ce genre de système peut être insuffisante au regard de la capacité perceptive de la fovéa (Jacob, 1991), ce qui peut engendrer une trop grande incertitude quant aux mesures comportementales au cours des tâches que nous voulons voir réaliser par l'utilisateur. Ainsi, pour un tel système, il conviendra d'avoir une erreur inférieure ou égale à 1° , c'est-à-dire une erreur de l'ordre de la partie du champ de vision prise en charge par la fovéa qui est d'environ $(3^\circ)^2$ (**Figure 182**), pour déterminer où l'utilisateur regarde nettement (Chi et al., 2009; Irwin, 2004). Néanmoins, dans un contexte clinique dans lequel l'étude des muscles oculaires serait importante, un système de suivi oculaire devrait pouvoir afficher une précision allant jusqu'à 0.1° (Leigh & Kennard, 2004).

Nous remarquons également que la vision occupe un rôle majeur dans la communication entre êtres humains car elle révélatrice d'une prise d'information. De cette manière, un ergonomiste peut, à l'aide de l'analyse des mouvements oculaires, identifier quelles parties d'une interface utilisateur reçoivent l'attention visuelle de ce dernier lors de l'exécution de tâches (Jacob & Karn, 2003). D'une manière générale, que l'on peut également étendre aux interfaces multimodales dans leur ensemble, il est permis de procéder à des mesures et annotations relatives aux interactions et expériences de l'utilisateur (Ferré et al., 2001; Weiss et al., 2012). Par l'observation et l'analyse de sessions d'utilisation, il est alors possible d'obtenir des informations sur d'éventuelles erreurs de conception de l'interface utilisateur (comme, par exemple, dans le cas d'un moteur de recherche (Brumby, Cutrell, & Sarin, 2006) ou de jeux vidéo (Allen, Gough, Mehdi, & Wink, 2003)). Ainsi, on peut pointer un comportement vis-à-vis d'une interface aussi bien qu'adapter dynamiquement cette dernière à ce comportement.

b) Un dispositif d'IHM

Le point de regard de l'utilisateur peut être utilisé comme donnée d'entrée d'IHM (Morimoto & Mimica, 2005; Vidal, 2014). Dans le cas d'applications interactives, l'utilisation du regard de l'utilisateur doit, contrairement à celui des applications diagnostiques, se traduire par un changement ou une réaction de l'interface (Duchowski, 2002). Les méthodes de suivi oculaire sont d'ailleurs utilisées, fréquemment dans le contexte de la compensation de handicaps, afin de remplacer la souris lors de l'utilisation d'un ordinateur (Dernoncourt, 2012; Moita, Oliveira, Santos, & Silva, 2012; San Agustin Lopez, 2009; Su, Wang, & Chen, 2006). Un dispositif à cet usage doit donc pouvoir fonctionner en temps réel, et à ce titre avoir de très bonnes performances de calcul (comme, par exemple, dans le cas du dispositif présenté dans (Murawski, 2010)). Pour ce type d'application en temps réel, on peut se permettre de perdre un peu de précision à la faveur du temps de calcul (De Santis & Iacoviello, 2009b). On signalera également que, dans le cas d'un SMA, l'architecture et le comportement des agents impliqués dans ces traitements jouent un rôle majeur pour l'obtention de telles performances (Ocello, 2003).

La position des yeux peut être vue comme un élément d'information utilisable lors d'un dialogue homme-machine comportant une grande variété de périphériques d'entrée, plutôt que comme un périphérique d'entrée à actionnement intentionnel (Asteriadis et al., 2007; Jacob, 1991). Effectivement, il est problématique de réaliser une sélection précise à l'aide du regard qui est amené, si l'on ne prend aucune mesure spécifique, à sélectionner tous les éléments affichés (Jimenez, Gutierrez, Latorre, & Zaragoza, 2008; Vidal, 2014) : ce problème est fréquemment assimilé au « touché de Midas » (Jacob, 1991). Il peut donc être plus aisé d'utiliser le regard de l'utilisateur, afin d'uniquement conditionner le contexte applicatif à ce dernier (comme, par exemple, dans les logiciels *The eyeBook* (Biedert et al., 2009) et *iDict* (Hyrskykari, 2006) utilisés en lecture) plutôt que de l'utiliser directement à des fins sélectives (Duchowski, 2002; Jacob & Karn, 2003). On pourra d'ailleurs remarquer que, dans un contexte général, les données relatives aux mouvements du regard conduisent à deux types d'interfaces très différents : celles sélectives et celles non-sélectives (Hyrskykari, 2006). Les interfaces sélectives utilisent la position du regard de l'utilisateur afin de transmettre directement des commandes au système. En pratique, les micro-saccades de l'œil, qui sont inintentionnelles, limitent la précision du suivi oculaire pour ce type d'application (Jacob, 1991). En revanche, les interfaces non-sélectives utilisent les informations du regard de l'utilisateur afin d'adapter indirectement le système aux besoins de l'utilisateur (Campbell & Maglio, 2001).

La vision par ordinateur, dont le suivi oculaire réalisé à l'aide de caméras fait partie au même titre que les méthodes d'analyse du visage (Mäkinen, 2007), est une technique importante dans la conception d'interfaces multimodales (Turk & Kölsch, 2003). En effet, la multimodalité joue un rôle important dans le discours, qui par nature combine des informations auditives et visuelles (Srivastava & Dev, 2013). Une étude (Xiao, Wong, Umali, & Pomplun, 2007) a constaté que des stimuli auditifs présentés lors de tâches de détection visuelle produisent des mouvements oculaires plus homogènes qu'en l'absence de stimulation acoustique. Le signal audio-vidéo est donc perçu et traité comme un signal continu et homogène. On pourra ainsi remarquer que la vision de l'utilisateur peut être utilisée naturellement en complément d'autres modalités d'entrées dans le système (Vidal, 2014). A ce titre, dans le contexte des IHM naturelles, l'analyse de données multimodale, prenant en compte la vision de l'utilisateur, permet d'améliorer les performances et la robustesse de l'interaction (comme, par exemple, dans le cas de la reconnaissance vocale (Abbas & Srivastava, 2012; Srivastava & Dev, 2013)). On pourra noter que cette fusion de données multimodales est aussi d'un grand intérêt dans le domaine des systèmes d'analyses biométriques (Gavrilova & Monwar, 2011).

3) La conception de notre dispositif de suivi oculaire

Nous allons maintenant proposer un processus de suivi oculaire dans sa globalité. Pour chacune des étapes constituant ce processus, nous nous efforcerons de présenter des alternatives possibles. Nous préciserons que notre approche se veut avant tout guidée par différentes exigences fonctionnelles que nous évoquerons. A ce titre, notre système n'est pas à prendre comme étant la meilleure solution possible mais comme une façon de répondre à notre problématique. Néanmoins, il est à noter que nous ne rentrons pas dans

l'étude et l'explication des comportements oculaires observés mais discuterons uniquement de la manière de les enregistrer et de les identifier automatiquement.

a) Un aperçu du fonctionnement général

Comme énoncé précédemment, nous comptons réaliser un système de suivi oculaire à visée médicale et paramédicale (voir la section **L'objectif PROTO1**), qui, par la suite, pourra aussi être utilisé à d'autres fins par un plus large public. Nous allons devoir nous assurer que notre dispositif réponde fonctionnellement à un certain nombre de contraintes. On peut trouver dans la littérature une énumération des caractéristiques que n'importe quelle technique de suivi oculaire devrait, dans l'idéal, posséder pour garantir des mesures utilisables en IHM (Chi et al., 2009; Mele & Federici, 2012b; Morimoto & Mimica, 2005) :

- L'exactitude, c'est-à-dire avoir une erreur inférieure ou égale à 1°.
- La fiabilité, qui passe par un comportement répétable du dispositif.
- La robustesse, qui garantit le fonctionnement sous différentes conditions (luminosité, lunettes, etc.) et sur différents utilisateurs.
- La non-intrusivité, c'est-à-dire que l'utilisation de l'appareil ne provoque ni blessure ni inconfort.
- La liberté des mouvements de la tête, qui permet une utilisation plus naturelle du dispositif.
- L'absence d'étalonnage initial, ce qui signifie que le système doit se calibrer automatiquement et instantanément.
- Les résultats en temps réel, qui est une condition sine qua non pour un dispositif d'IHM.

On pourra ajouter, comme évoqué par Beymer et Russell en 2005, que le dispositif ne doit pas être trop encombrant, de sorte à ne pas altérer le comportement de l'utilisateur vis-à-vis des tâches qui lui sont demandées, et que son utilisation doit être relativement simple (Hansen & Pece, 2005). Notre système se doit, aussi bien au niveau matériel que logiciel, de respecter ces critères pour être efficacement utilisable. Comme nous l'avons précédemment évoqué, le prix d'achat de ce genre d'appareil est également une contrainte, et nous estimons que le nôtre devrait être bon marché pour toucher un large public (comme, par exemple, cela est également signalé dans les études (De Santis & Iacoviello, 2009a; Hansen & Ji, 2010; Nouredin et al., 2005)). En outre, cet ensemble d'exigences de performances/qualité se traduit, comme le soulignent Chi et al. en 2009, par plusieurs difficultés majeures auxquelles nous allons devoir apporter des solutions :

- Comment assurer la robustesse du système, en tenant compte des conditions environnementales et des interférences causées par l'utilisateur ?
- Comment garantir l'exactitude de l'estimation de la direction du regard ainsi qu'un étalonnage simple tout en laissant la liberté des mouvements de la tête ?
- Comment combiner la précision d'un modèle géométrique 3D du globe oculaire avec la simplicité de constitution d'un modèle de mappage basé sur des données expérimentales ?
- Comment gérer les saccades oculaires et les mouvements inconscients afin d'utiliser le suivi oculaire en IHM ?

D'un point de vue technique, l'ensemble du système peut, d'une manière basique, être décomposé de la sorte (Allen et al., 2003; Hansen & Ji, 2010; Mele & Federici, 2012a) :

- L'acquisition des données (Miyoshi & Murata, 2001a), c'est-à-dire l'enregistrement des mouvements oculaires comprenant :
 - o La détection de l'œil de l'utilisateur.
 - o La détection de la tête de l'utilisateur.
 - o L'estimation du regard de l'utilisateur.
- Le traitement des données (Jacob, 1991; Salvucci & Goldberg, 2000), c'est-à-dire la reconnaissance de mouvements oculaires tels que les saccades et les fixations. Cette procédure inclut :
 - o Le filtrage du bruit.
 - o L'identification des mouvements oculaires.
- La cartographie des données, ce qui revient à mettre en relation les mouvements oculaires et des éléments de la scène observée.
- L'analyse des données, c'est-à-dire l'identification de comportements oculaires à partir des relations précédemment établies.
- L'interprétation des données, à savoir le fait de relier les comportements observés à des critères d'utilisabilité.

Nous allons détailler ces différents points mais, auparavant, nous pouvons déjà donner une représentation de l'ensemble du système (**Figure 184**), tel que nous l'imaginons, ainsi que de ses différents composants, tels qu'ils sont présents dans la littérature (De Santis & Iacoviello, 2009a; Hansen & Ji, 2010; Villanueva, Cabeza, & Porta, 2006).

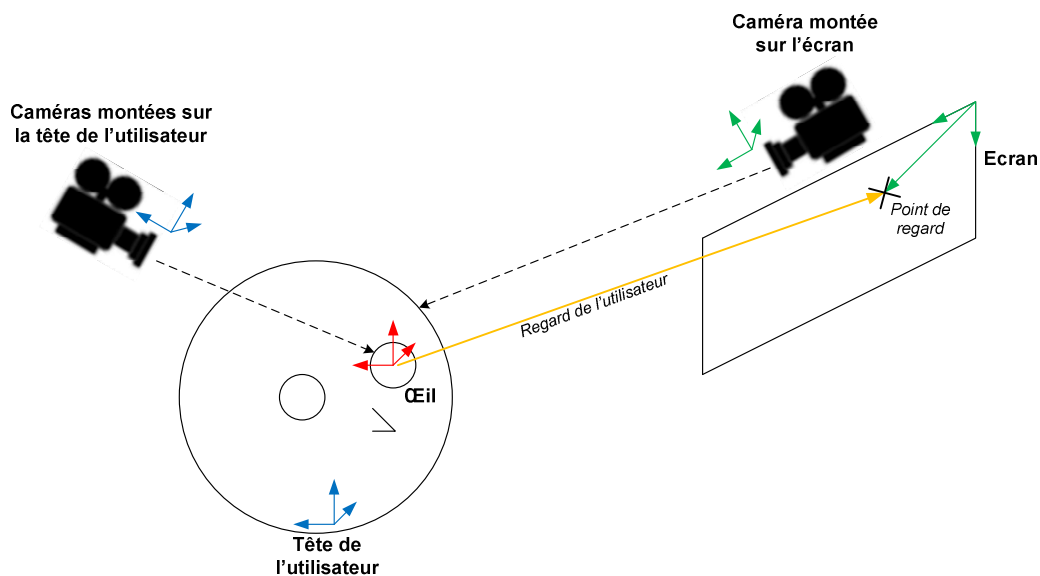


Figure 184 : Schéma général du dispositif de suivi oculaire et des repères manipulés

Sur la **Figure 184**, les repères de même couleur (**rouge**, **bleu** et **vert**) sont liés, c'est-à-dire que leurs positions relatives sont constantes et, par la même occasion, seront connues consécutivement au paramétrage initial du système. Au niveau algorithmique, de tels repères seront donc assimilés en un seul car des transformations à paramètres constants permettent de passer de l'un à l'autre. Nous pouvons ainsi nous limiter à l'utilisation de trois repères : le repère de l'œil de l'utilisateur (**rouge**), celui de la tête de ce dernier (**bleu**) et pour finir, celui de l'écran (**vert**). Pour désigner ce dernier repère, on pourra parler de

« repère général ». Cette représentation est d'ailleurs une extension de celle proposée par Villanueva et al. en 2006, dans la mesure où elle introduit des repères supplémentaires liés à la tête de l'utilisateur. Ceci permettra en conditions réelles d'offrir plus de liberté de mouvements à l'utilisateur (Allison et al., 1996; Hansen & Ji, 2010; Świrski et al., 2012), en alliant la prise d'images proches, via des caméras montées sur la tête de l'utilisateur, à celle prise depuis une position assimilable à celle de l'écran. On offre à l'utilisateur une bien plus grande liberté des mouvements de la tête que, par exemple, dans les études conduites par Al-Wabil, Meldah, Al-Suwaidan, et AlZahrani en 2010. En effet, il est ainsi permis de s'affranchir de la majorité des cas d'occlusions, et d'autres cas extrêmes dus aux positions des yeux et de la tête par rapport à la caméra, qui posent fréquemment problème pour ce genre d'applications (Hansen & Ji, 2010).

D'autre part, ce découpage en deux sous-problèmes d'identification à différentes échelles, à savoir d'une part l'identification de l'œil et d'autre part celle de la tête, étant réalisé par au moins deux caméras distinctes (Kohlbecher et al., 2008), nous avons la possibilité d'utiliser des caméras de moins haute résolution et donc meilleur marché que pour une identification en cascades réalisée à l'aide d'une caméra unique. Il aurait été également possible d'utiliser une caméra « pan-tilt » (comme, par exemple, dans (Beymer & Flickner, 2003; Coutinho & Morimoto, 2006; Nouredin et al., 2005)), cependant la mise en œuvre d'une telle solution coûte cher et souffre d'un problème de latence consécutif aux mouvements que doit effectuer la caméra. La **Figure 185** donne un aperçu des traitements à effectuer ainsi que des objets que nous allons manipuler.

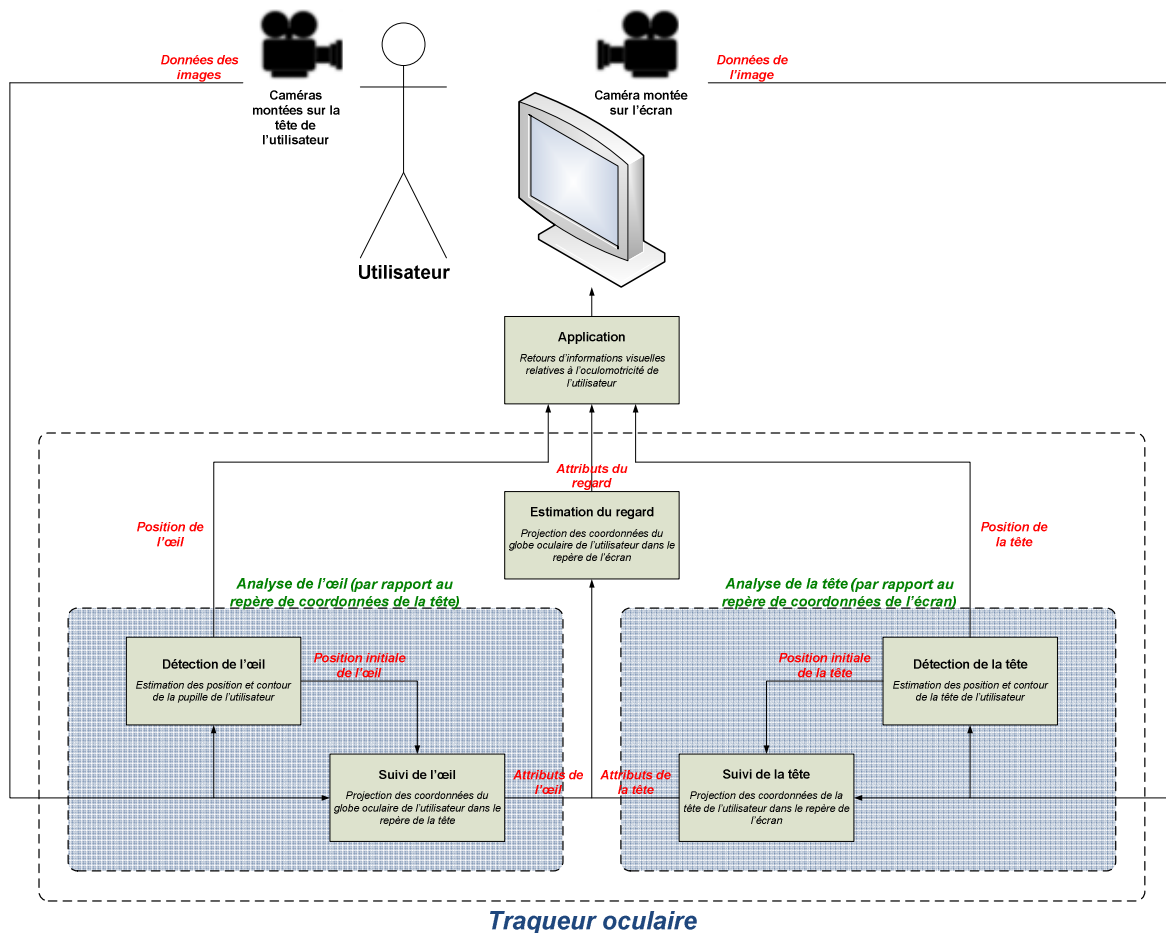


Figure 185 : Schéma des composants et flux de notre système de suivi du regard

On pourra cependant remarquer que ce choix d'utiliser plusieurs caméras simultanément va nécessiter la synchronisation des acquisitions. Cela va complexifier un peu plus l'analyse de ces images, en particulier dans le cas d'une analyse en temps réel (Schneider et al., 2011). Concernant les caméras filmant les yeux, nous en utiliserons deux par œil afin de procéder à une reconstruction 3D de la pupille de l'utilisateur, dans le but de déterminer l'axe optique de ce dernier. En effet, l'œil, ou même l'iris, peut être assez difficile à discerner en raison d'occlusions possibles par l'ossature du visage ou par les cils et sourcils, alors que la pupille est, quant à elle, toujours visible et contraste normalement suffisamment avec le reste de l'œil pour être aisément reconnaissable (Morimoto & Mimica, 2005). En outre, lorsque l'œil pointe vers une direction spécifique, il est mis en rotation. Cette rotation peut être décomposée en deux rotations appliquées à l'état initial de l'œil (Coutinho & Morimoto, 2006). Ainsi, nous comptons pouvoir déterminer, dans les plans de l'image des deux caméras utilisées pour chaque œil, ces deux rotations. Si nous n'utilisons qu'une seule caméra à cet effet, il nous faudrait gérer différents cas ambigus et procéder à des étapes supplémentaires de calibrage (Kohlbecher et al., 2008; Wang, Sung, & Venkateswarlu, 2005). On remarquera néanmoins que la proximité de l'œil avec ces caméras, dans un tel dispositif de suivi monté sur la tête de l'utilisateur, implique qu'elles doivent être positionnées en respectant un angle important par rapport à l'axe visuel, de façon à ne pas bloquer le regard de l'utilisateur. Ceci rend l'observation et l'identification de la pupille plus délicates à cause de l'obstruction potentielle des cils et de la forme elliptique que prend la pupille vue sous cet angle (Świrski et al., 2012).

Pour finir, on signalera que la **Figure 185** ne représente que la partie des traitements correspondant à l'acquisition des données (Miyoshi & Murata, 2001a). Effectivement, nous allons concentrer notre étude sur cette partie du processus pour laquelle nous proposerons finalement une architecture MA. On notera aussi que, puisque les deux yeux pointent généralement au même endroit, il est habituel de ne suivre qu'un seul œil (Jacob, 1991). Dans notre cas, étant donné que nos utilisateurs pourraient souffrir de pathologies impactant la vergence, il est préférable de suivre les deux yeux en parallèle. L'architecture MA que nous utilisons est naturellement adaptée à ce type de traitements car un SMA supporte intrinsèquement les traitements en parallèle et/ou distribués (Srivastava & Dev, 2013). D'autre part, les organisations d'agents étant facilement reconfigurables, ceci rend le système adaptatif à différents contextes et permet ainsi de tester différentes stratégies (Hutzler et al., 2006). Néanmoins, comme il est facile de répliquer les traitements pour l'analyse d'un œil afin de les appliquer à l'autre, nous ne travaillerons que sur un unique œil.

b) L'acquisition des données

L'acquisition des données relatives à l'œil et à ses mouvements prend une part importante dans la réalisation d'un système de suivi oculaire. Cette partie du processus comprend ainsi les traitements, de complexités non-négligeables, suivants (Hansen & Ji, 2010) :

- La détection de l'œil de l'utilisateur
- La détection de la tête de l'utilisateur
- L'estimation du regard de l'utilisateur

En règle générale, la partie matérielle des capteurs optiques, tels que ceux que nous utiliserons pour réaliser le suivi oculaire, assure l'acquisition des données brutes de l'image

ainsi que des prétraitements (Murawski & Rózanowski, 2013). Le reste des traitements est habituellement réalisé logiciellement (De Santis & Iacoviello, 2009b). Nous allons, pour chacun de ces traitements logiciels, détailler les algorithmes et méthodes mis en œuvre dans notre système. Concernant leur implémentation (voir la section **L'objectif PROTO1**), nous nous appuyerons sur l'API dénommée OpenCV (Intel & Willow Garage, 2013) ainsi que sur l'Intel® Perceptual Computing SDK (Intel, 2013) sans forcément rentrer dans les détails de la programmation (pour plus d'informations sur les usages courants d'OpenCV, consulter (Laganière, 2011)). D'un point de vue logiciel, on peut parler de middleware pour désigner les composants assurant ces traitements.

On signalera que les images, dont nous faisons l'acquisition en 256 niveaux de gris, pourraient être prétraitées pour en retirer le bruit dû à l'utilisation d'un matériel électronique et d'un dispositif optique (par exemple, consulter (Li et al., 2005)). Dans le cas de notre système, nous avons décidé de supposer que la segmentation par seuillage que nous effectuerons est assez grossière pour ne pas être affectée par le bruit d'acquisition des données. A ce titre, nous ne procéderons pas à un tel filtrage préalable.

1. La détection de l'œil de l'utilisateur

Concernant la détection de l'œil, nous allons passer en revue des solutions d'analyse en temps réel d'images de la tête ou de l'œil de l'utilisateur afin de déterminer un traitement approprié à nos besoins. En particulier, de telles informations sont très utiles pour l'étude et la compréhension de la communication humaine et elles peuvent être utilisées afin de déterminer où se porte l'attention d'une personne (pour des exemples d'application, consulter (Tessendorf et al., 2010; Weibel et al., 2013)). En outre, notre approche doit nous permettre de nous affranchir des conditions de luminosité et traiter des images capturées avec un matériel bon marché. On peut également ajouter que, dans le domaine de l'imagerie par ordinateur, les défauts typiques des données échantillonnées, tels que les artefacts d'échantillonnage, le crénelage spatial et le bruit, peuvent rendre flous et déconnecter les bords recherchés (Hegadi, Kop, & Hangarge, 2010). A ce titre, notre solution se devra d'être robuste, précise et rapide afin de pallier à ces limitations, et les contourner efficacement le cas échéant.

Dans cette perspective, nous allons en premier lieu faire appel à un seuillage par un algorithme amélioré de la méthode d'Otsu (Liao, Chen, & Chung, 2001; Otsu, 1979). Cette méthode a, entre autres, l'avantage de ne nécessiter aucun apprentissage ni modèle prédictif des mouvements de l'œil car elle est basée sur ses caractéristiques de forme et d'apparence (De Santis & Iacoviello, 2009a; Hansen & Ji, 2010). Effectivement, les méthodes de modèles déformables, de correspondance de modèle, ou les méthodes statistiques nécessitent parfois un apprentissage approfondi, ce qui limite les possibilités de proposer une méthode directement prête à l'emploi. De plus, ces méthodes sont sensibles aux changements d'éclairage, de formes et de couleurs des yeux, et peuvent ainsi nécessiter des interventions manuelles, et un grand nombre de calculs (Noureddin et al., 2005). Ainsi, leur emploi peut amener à négliger certaines différences interindividuelles, telles que les anomalies potentiellement pathologiques, qu'il pourrait être très souhaitable de pouvoir détecter (Angelini, Jin, & Laine, 2005). Dans notre cas, la méthode de seuillage retenue doit permettre de diminuer le nombre d'informations contenues dans l'image, accélérant et

simplifiant d'autant les futurs traitements, sans pour autant perdre les informations essentielles à la détection de l'œil de l'utilisateur (De Santis & Iacoviello, 2009b).

Pour cette détection de l'œil de l'utilisateur, nous comptons notamment détecter la pupille de ce dernier. Dans ce domaine, il est d'ailleurs avéré que la segmentation d'images par une méthode de « level set » discret, telle que décrite par De Santis et Iacoviello en 2007, démontre l'intérêt de tenir compte de la topologie discrète des images pour réaliser efficacement un tel traitement. En outre, ce genre de méthodes est particulièrement adapté pour déterminer automatiquement les limites d'un objet (Sethian, 1999, 2002). Il est alors spécifiquement possible de récupérer les propriétés géométriques intrinsèques du contour ainsi déterminé (Han, Xu, & Prince, 2003), et, dès lors, obtenir les caractéristiques de l'œil nécessaires à la détermination de la direction du regard de l'utilisateur. Malheureusement, on pourra remarquer que, du point de vue du temps de calcul (Angelini et al., 2005; Shi & Karl, 2005a), ces méthodes peuvent s'avérer lentes, ce qui conduit souvent à ne pouvoir travailler que sur seulement une partie de l'image, qui est déterminée après la définition d'une région d'intérêt (pour des exemples, consulter (De Santis & Iacoviello, 2006; Pérez et al., 2003; Świrski et al., 2012)). Dans notre cas, pour ne pas avoir à procéder à un tel zonage des images, nous effectuerons une acquisition vidéo suffisamment proche de l'œil, qui ne nécessitera pas d'ailleurs une très haute résolution d'image. Nous souhaitons obtenir, de la sorte, directement des images telles que celles obtenues après un zonage et nous affranchir de calculs superflus. On ajoutera que nous souhaitons utiliser des algorithmes de level set rapides s'appuyant sur la discrétisation (De Santis & Iacoviello, 2007; Shi & Karl, 2005b), et sur des calculs s'affranchissant de la résolution d'équations aux dérivées partielles (Shi & Karl, 2005a), afin de tendre vers un traitement des images en temps réel (Shi & Karl, 2008).

Dès que nous disposons du contour de la pupille de notre utilisateur, nous allons pouvoir tenter d'extraire les caractéristiques de cette dernière. Pour cela, nous allons déterminer les paramètres de l'ellipse que forme ce contour (Fitzgibbon, Pilu, & Fisher, 1999; Wu, 2008), qui n'est autre que la projection de la pupille de l'utilisateur sur le plan capturé par la caméra. En utilisant deux caméras, il est ensuite possible, par stéréoscopie (Kohlbecher et al., 2008), d'établir le vecteur normal au centre de l'œil passant, c'est-à-dire le vecteur directeur de l'axe optique (**Figure 181**).

Au final, la solution retenue, en plus d'être très rapide grâce à l'amélioration de la procédure d'Otsu (déjà très rapide mais dont le temps de calcul augmente exponentiellement avec le nombre de classes désirées (Liao et al., 2001)), à la méthode de level set discret sans résoudre les équations différentielles partielles (Shi & Karl, 2005a) et à la technique stéréoscopique d'ajustement direct d'ellipse (Kohlbecher et al., 2008; Wu, 2008), cherche à pallier les faiblesses habituelles des méthodes basées sur les caractéristiques de l'œil. Ce genre de solution a prouvé sa précision et sa robustesse dans le cas des images à faible contraste (De Santis & Iacoviello, 2006, 2007) mais aussi sa vitesse dans des applications en temps réel (Shi & Karl, 2005c; Zhang, 2006). Ainsi, nous allons détailler chacune des différentes méthodes utilisées pour notre traitement algorithmique de détection de l'œil, à savoir :

- La segmentation en quatre niveaux de gris.
- L'identification de la pupille.
- L'extraction des caractéristiques oculaires.

Notre procédure de détection de l'œil de l'utilisateur sera aussi bien détaillée au niveau formel/théorique qu'au niveau pratique/algorithmique.

a. La segmentation en quatre niveaux de gris

Afin de procéder à une segmentation multi-classes, c'est-à-dire à un traitement d'images qui vise à séparer les pixels en différents ensembles respectant certains critères, les méthodes de seuillages sont fréquemment utilisées (Eichmann & Luessi, 2005). Ces méthodes de segmentation de bas niveau sont fréquemment employées pour séparer des objets d'un arrière-plan en s'appuyant sur l'intensité des pixels qui les constituent. Pour procéder à une telle segmentation en M classes, il faut déterminer $M - 1$ seuils qui seront utilisés pour répartir ces pixels en différents groupes selon les valeurs de leur intensité par rapport à ces seuils. Nous n'allons pas chercher ni à rentrer en profondeur dans la définition du problème de la segmentation en quatre niveaux de gris, ni à présenter les différentes façons de l'aborder (pour approfondir le sujet, consulter (Luessi, Eichmann, Schuster, & Katsaggelos, 2009; Sezgin & Sankur, 2004)), ni même à développer de nouvelles méthodes pour le résoudre. Nous nous contenterons d'appliquer une méthode correspondant à nos besoins fonctionnels tout en respectant les contraintes que nous nous sommes fixées.

A ce titre, nous avons observé que la méthode d'Otsu (Otsu, 1979) est l'une des méthodes de sélection de seuil les plus utilisées pour les images issues du monde réel en tenant compte de l'uniformité des régions et de leurs formes (Sezgin & Sankur, 2004). En effet, Sezgin et Sankur en 2004 ont évalué les performances quantitatives, hormis la complexité algorithmique qui devient critique dans le cas d'un seuillage multi-niveaux, de différents algorithmes et ils ont établi que les plus performants pour l'analyse d'images étaient ceux basés sur les clusters (comme, par exemple, la méthode d'Otsu ou celle de Kittler (Kittler & Illingworth, 1986)) ou l'entropie (par exemple, consulter (Kapur, Sahoo, & Wong, 1985)). Cette méthode semble donc adaptée à notre problématique et à l'analyse d'images de $2^8 = 256$ niveaux de gris qui sont capturées en périphérie de l'œil. Cependant, elle a la faiblesse de s'appuyer sur une recherche exhaustive pour évaluer le critère de maximisation de la variance interclasse. D'ailleurs, si le nombre de classes que l'on souhaite obtenir augmente, la méthode d'Otsu nécessitera trop de temps de calcul pour être pratique pour la détermination de seuils multi-niveaux (Liao et al., 2001; Luessi et al., 2009). Dans notre cas, nous avons besoin de déterminer quatre niveaux de gris (Lam & Yan, 1996), c'est-à-dire quatre classes de pixels et donc trois seuils différents afin de ne pas perdre d'informations, comme c'est le cas avec la binarisation de ce type d'images (Kittler & Illingworth, 1985). Nous allons ainsi devoir utiliser un algorithme reprenant la classification de la méthode d'Otsu tout en améliorant sa rapidité d'exécution. En 2001, Liao et al. ont proposé une telle méthode, nous allons donc l'appliquer à notre problématique. Cette méthode se base, comme celle originalement proposée par Otsu en 1979, sur l'histogramme normalisé des niveaux de gris de l'image. Il faut donc, en premier lieu, calculer l'histogramme de l'image puis le normaliser (pour plus de détails, consulter (Eichmann & Luessi, 2005)). Ici, nous souhaitons seuiller une image de L niveaux de gris en $M = 4$ classes, ainsi il nous faut calculer $M - 1 = 3$ seuils, à savoir $\{t_1, \dots, t_{M-1}\} = \{t_1, t_2, t_3\}$. Avec cette notation, nous pouvons décrire les classes recherchées ainsi :

$$\{C_1[1, \dots, t_1], \dots, C_i[t_{i-1} + 1, \dots, t_i], \dots, C_M[t_{M-1} + 1, \dots, L]\}$$

Soit, dans notre cas où $M = 4$:

$$\{C_1[1, \dots, t_1], C_2[t_1 + 1, \dots, t_2], C_3[t_2 + 1, \dots, t_3], C_4[t_3 + 1, \dots, L]\}$$

La méthode proposée doit nous permettre de trouver les meilleurs seuils pour réaliser cette classification. Ces seuils sont appelés seuils optimaux et notés $\{t_1^*, \dots, t_{M-1}^*\} = \{t_1^*, t_2^*, t_3^*\}$. Afin de les déterminer, nous allons utiliser les moments cumulatifs d'ordre zéro et un, c'est-à-dire des indicateurs de la dispersion des pixels au sein des classes que nous voulons déterminer. En d'autres termes, nous voulons, à partir des informations relatives à la probabilité d'être dans une classe ainsi que la valeur moyenne de l'ensemble des pixels constituant une classe, déterminer avec précision les seuils entre ces classes. Ainsi, nous déterminerons les seuils optimaux, au travers de la maximisation d'une « fonction objectif », de la manière suivante :

$$\{t_1^*, \dots, t_{M-1}^*\} = \max_{1 \leq t_1 < \dots < t_{M-1} < L} \{(\sigma_B')^2(t_1, \dots, t_{M-1})\}$$

Soit, dans notre cas où $M = 4$:

$$\{t_1^*, t_2^*, t_3^*\} = \max_{1 \leq t_1 < t_2 < t_3 < L} \{(\sigma_B')^2(t_1, t_2, t_3)\}$$

où $(\sigma_B')^2$ est appelée la variance interclasse modifiée, obtenue par une première optimisation de la méthode d'Otsu, en retirant les termes de la variance interclasse $(\sigma_B)^2$ ne dépendant pas du choix des seuils (Liao et al., 2001). $(\sigma_B')^2$ est définie de la sorte :

$$(\sigma_B')^2 = \sum_{k=1}^M \omega_k \mu_k^2$$

avec

$$\omega_k = \sum_{i \in C_k} p_i = \sum_{i=t_{k-1}+1}^{t_k} p_i \quad \text{où } i \in [1, \dots, L], k \in [1, \dots, M], t_0 = 0 \text{ et } t_M = L$$

$$\mu_k = \sum_{i \in C_k} ip_i / \omega_k = \frac{\mu(k)}{\omega_k} \quad \text{où } \mu(k) = \sum_{i \in C_k} ip_i = \sum_{i=t_{k-1}+1}^{t_k} ip_i$$

Dans notre cas, soit avec $M = 4$, le problème revient à calculer la variance interclasse modifiée suivante :

$$(\sigma_B')^2 = \omega_0 \mu_0^2 + \omega_1 \mu_1^2 + \omega_2 \mu_2^2 + \omega_3 \mu_3^2$$

Une autre optimisation, introduite par cette méthode, relève du pré-calcul. En effet, afin de calculer la variance interclasse modifiée optimale, nous allons être amené à comparer un grand nombre de valeurs obtenues via des opérations en grande partie communes. Ainsi, au lieu de recalculer plusieurs fois les mêmes opérations, nous allons pouvoir gagner un temps précieux en stockant les résultats (Luessi et al., 2009). Nous pré-calculerons ensuite les moments d'ordre zéro et un, entre les niveaux de gris u et v , que nous stockerons dans deux tableaux P et S (Liao et al., 2001). Nous nous servirons de ces tableaux pour déterminer ω_k et $\mu(k)$, en pré-calculant la variance interclasse modifiée. Finalement, nous maximiserons la somme de ces variances interclasses modifiées afin de déterminer les valeurs optimales des seuils, avant de seuiller l'image à l'aide de ces derniers.

En pratique, nous allons partir d'une image pour laquelle nous allons réaliser les pré-calculs précédemment cités, puis nous procéderons à l'évaluation des seuils optimaux avant de réaliser le seuillage final. Algorithmiquement, ceci va s'effectuer en six étapes :

1. Calculer l'histogramme normalisé de l'image afin d'obtenir, pour chaque niveau de gris $1 < v \leq L$, la probabilité p_v d'un pixel de l'image d'être du niveau de gris v .
2. Initialiser les pré-calculs des moments par rapport à la première classe, c'est-à-dire au 1^{er} niveau de gris soit quand $u = 1$, de la façon suivante :

$$P(1, v + 1) = P(1, v) + p_{v+1} \quad \text{avec } P(1, 0) = 0$$

$$S(1, v + 1) = S(1, v) + (v + 1)p_{v+1} \quad \text{avec } S(1, 0) = 0$$

3. Remplir les tableaux pré-calculés P et S , c'est-à-dire pour $1 < u \leq L$, via les formules :

$$P(u, v) = P(1, v) + P(1, u - 1) \quad \text{en particulier } \omega_k = P(t_{k-1} + 1, t_k)$$

$$S(u, v) = S(1, v) + S(1, u - 1) \quad \text{en particulier } \mu(k) = S(t_{k-1} + 1, t_k)$$

4. Pré-calculer la variance interclasse modifiée puis la stocker dans un tableau H , explicitée pour la classe C_k , avec $1 \leq k \leq M$, de la sorte :

$$H(t_{k-1} + 1, t_k) = \frac{S(t_{k-1} + 1, t_k)^2}{P(t_{k-1} + 1, t_k)} \quad \text{car } \omega_k \mu_k^2 = \frac{\mu(k)^2}{\omega_k}$$

5. Evaluer les seuils possibles, $\{t_1, \dots, t_{M-1}\} = \{t_1, t_2, t_3\}$, afin de déterminer leurs valeurs optimales $\{t_1^*, \dots, t_{M-1}^*\} = \{t_1^*, t_2^*, t_3^*\}$, à l'aide de la fonction objectif suivante, dont on cherchera le maximum :

$$(\sigma_B')^2(t_1, t_2, t_3) = H(1, t_1) + H(t_1 + 1, t_2) + H(t_2 + 1, t_3) + H(t_3 + 1, L)$$

6. Seuiller l'image en quatre niveaux de gris à l'aide des seuils optimaux que l'on a déterminé à l'étape précédente $\{t_1^*, t_2^*, t_3^*\}$. En effet, on va pouvoir définir pour chaque pixel $x \in C_k$ un niveau de gris i_k via la formule suivante :

$$\forall x \in C_k, i_k = \frac{(k - 1)L}{M - 1} \quad \text{où } 1 \leq k \leq M$$

Dans le cas de $M = 4$ niveaux de gris, on obtient :

$$i_1 = 0, i_2 = 85, i_3 = 170 \text{ et } i_4 = 255$$

Une fois ce seuillage réalisé, nous avons une image constituée des quatre niveaux de gris nécessaires à l'identification des différentes parties de l'œil (Lam & Yan, 1996). Il nous est donc maintenant possible de procéder à l'identification de la pupille à partir de cette image segmentée par seuillage.

Nous signalons que cette méthode pourrait encore être optimisée à l'aide de la « programmation dynamique » (comme, par exemple, cela est réalisé dans (Luessi et al., 2009)). En effet, cette approche permet de s'affranchir d'une recherche exhaustive telle que nous l'effectuons, et elle peut également bénéficier d'optimisation visant à réduire encore sa complexité algorithmique (pour plus de détails, consulter (Eichmann & Luessi, 2005)).

b. L'identification de la pupille

Maintenant que nous avons réduit par seuillage la complexité et le nombre de nuances de l'image originale, sans perdre pour autant les informations qui nous sont nécessaires (De Santis & Iacoviello, 2009b), nous allons pouvoir procéder à une nouvelle segmentation afin d'identifier spécifiquement la pupille. Cette segmentation, qui peut d'ailleurs être utilisée comme méthode de poursuite d'objets (Yilmaz, Li, & Shah, 2004), va nous permettre d'obtenir le contour de la pupille, ce qui est nécessaire pour en réaliser le suivi dans le temps. Pour cela, nous allons nous attacher à trouver dynamiquement l'interface entre la pupille et l'iris. Dans ce but, nous nous appuyerons sur un modèle déformable (pour une revue des différents types de modèles déformables utilisés en 3D consulter (Montagnat, Delingette, & Ayache, 2001) et en 2D consulter (Hegadi et al., 2010)), qui peut aussi être appelé « contour actif » ou « modèle de surface » selon le cas. Les modèles déformables sont fréquemment utilisés pour déterminer le contour d'organes dans l'imagerie médicale (pour une étude comparative de modèles existants, consulter (He et al., 2008)), grâce à leur sensibilité au bruit limitée et à leur capacité à intégrer des connaissances expertes sur le ou les objets à détecter, et ils s'avèrent donc adaptés à notre problématique (Angelini et al., 2005).

La littérature classe de tels modèles, selon leurs représentations et leurs implémentations (Angelini et al., 2005; Hegadi et al., 2010; Xu, Yezzi, & Prince, 2001), en deux types : ceux à contours actifs paramétriques (par exemple, la méthode dite des « snakes » (Kass, Witkin, & Terzopoulos, 1988) ou encore celle des « balloons » (Cohen, 1991)) et ceux à contours actifs géométriques, géodésiques et de level set basés sur des analyses de gradients ou de régions (pour des exemples de référence, consulter (Caselles, Catté, Coll, & Dibos, 1993; Caselles, Kimmel, & Sapiro, 1997; Chan & Vese, 2001; Malladi, Sethian, & Vemuri, 1995)). Les contours actifs paramétriques représentent explicitement des courbes ou surfaces paramétrées, alors que les contours actifs, qui s'appuient sur des level set, les représentent implicitement comme des fonctions évoluant selon une formulation eulérienne, c'est-à-dire selon une formulation qui régit habituellement l'évolution des fluides non-visqueux (Sethian, 1999). Ceci revient à mettre à jour, dans le temps, la fonction de level set pour un niveau donné, au lieu de décrire explicitement l'évolution de la courbe au cours du temps à l'aide d'une formulation lagrangienne (Hegadi et al., 2010).

Dans le cas des contours actifs paramétriques, le choix de la paramétrisation n'est pas évident, en particulier si la méthode doit servir à l'analyse d'images évoluant dans le temps, comme, par exemple, cela est le cas dans une séquence vidéo. D'autre part, avec ce type de méthodes, le contour initial, qui est fixé arbitrairement ou plus souvent qui repose sur une expertise du contour attendu et amené à évoluer en accord avec la formulation du problème (Niessen, ter Haar Romeny, & Viergever, 1998), doit être relativement proche du contour recherché sous peine de ne pas arriver à déterminer les bords recherchés (Angelini et al., 2005). Quant aux modèles à contours actifs de level set, ils présentent plusieurs avantages par rapport à cette première catégorie, qui leur est antérieure (Xu et al., 2001). En particulier, nous relevons les points suivants (Angelini et al., 2005; Han, Xu, & Prince, 2001) :

- Ces modèles sont indépendants d'un quelconque paramétrage qui pourrait rigidifier le modèle et le complexifier.
- Les level set ont une bonne stabilité numérique et permettent d'obtenir simplement les propriétés géométriques du contour.
- Les changements topologiques y sont effectués automatiquement (pour plus de détails sur les concepts de topologie en imagerie numérique, consulter (Kong & Rosenfeld, 1989)).

- Les contours obtenus ne risquent pas de s'auto-intersecter.
- La formulation du problème en 2D pourra facilement être étendue en ND.

On remarquera alors que ces modèles à contours actifs sont plus fiables et faciles à développer, qu'ils sont néanmoins plus gourmands en quantité de calculs que leurs prédécesseurs (Iwashita, Kurazume, Tsuji, Hasegawa, & Hara, 2004). Cela peut d'ailleurs être un frein à leur utilisation en temps réel (Angelini et al., 2005; Iwashita et al., 2004; Świrski et al., 2012). Cependant, selon la littérature, il existe des moyens d'optimiser la vitesse de traitement des méthodes s'appuyant sur les level set afin de pouvoir les utiliser pour des applications en temps réel (De Santis & Iacoviello, 2007; Iwashita et al., 2004; Shi & Karl, 2005c). En effet, la grande majorité d'entre elles consiste à résoudre des équations aux dérivées partielles : cela permet d'obtenir des résultats d'une grande précision mais nécessite un certain temps de calcul (Shi & Karl, 2005b; Zhang, 2006). Or, nous travaillons avec du contenu vidéo dans lequel l'espace est uniformément discret avec un pas d'échantillonnage fixé à 1. Il nous est donc possible de travailler avec des versions discrètes des méthodes de level set, ce qui va nous permettre de nous affranchir de la résolution de nombreux calculs. Dans notre contexte applicatif, une approximation au pixel prêt est suffisante, et une précision supérieure ne nous apporterait pas un meilleur rendu (De Santis & Iacoviello, 2007; Shi & Karl, 2005a). D'ailleurs, en 2010, dans leur état de l'art sur les modèles déformables, Hegadi, Kop et Hangarge précisent qu'il peut être plus simple d'envisager le problème de ce point de vue. C'est en partant de ces différents constats que nous avons décidé d'utiliser une méthode utilisant des modèles à contours actifs géodésiques (pour approfondir le sujet, consulter (Sethian, 1999)), afin d'allier la simplicité de conception algorithmique (et donc, par la suite, celle de la maintenance) à la performance lors de la segmentation.

Par ailleurs, nous pouvons aussi discuter de la souplesse topologique qui a longtemps été revendiquée comme un avantage majeur des modèles déformables géodésiques par rapport aux modèles déformables paramétriques (Niessen et al., 1998). Effectivement, il est difficile de gérer les changements topologiques engendrés par la division et la fusion avec les modèles déformables paramétriques, contrairement aux level set, pour lesquels la paramétrisation du contour est établie seulement après la complète déformation du modèle (Hegadi et al., 2010). Néanmoins, une telle flexibilité n'est pas toujours souhaitable, surtout si la topologie finalement attendue est bien connue (Han et al., 2003). Dans notre cas, nous devons obtenir systématiquement un seul et unique contour elliptique, potentiellement tronqué mais toujours convexe, qui correspond à la projection de la pupille de l'utilisateur sur l'image acquise par la caméra. Il est donc important pour nous de pouvoir conserver la topologie du contour initial, qui est identique à celle de l'objet dont nous voulons déterminer les bords (pour une revue des transformations topologiques discrètes, consulter (Couprie & Bertrand, 2012)). Ainsi, nous avons intégré à notre algorithme des éléments de la méthode utilisée par Han et al. en 2001 pour garantir la préservation topologique de notre technique de level set (Bertrand, 1994; Rosenfeld, Kong, & Nakamura, 1998). On remarquera d'ailleurs que la méthode de level set par propagation de « bande étroite » (pour plus d'informations concernant cette méthode, consulter (Malladi et al., 1995; Sethian, 1999)), qui a été améliorée par Han et al., avait été choisie pour sa rapidité d'exécution. On peut donc escompter que ces améliorations fonctionnent aussi avec notre méthode discrète, qui est sensée être encore plus performante que celle par bande étroite (Shi & Karl, 2005b).

Pour finir, avant de détailler la modélisation du problème, on ajoutera que ce type de modèles présente aussi des avantages vis-à-vis d'autres méthodes de détection de bords, telles que, par exemple, celles basées sur la diffusion anisotropique qui sont sensibles à la luminosité, au contraste et au bruit (Perona & Malik, 1990). Effectivement, en utilisant des modèles à contours actifs géodésiques, il sera toujours possible de contraindre l'évolution du modèle, ce qui est d'ailleurs le principe de base de ce type d'approches (Zhang, Zhang, Song, & Zhou, 2010), afin de tenir compte simultanément des caractéristiques de l'image traitée (c'est ce que l'on appelle une approche « bottom-up ») ainsi que des propriétés géométriques connues du ou des objets (c'est ce que l'on appelle une approche « top-down ») que nous voulons extraire. On signalera qu'il s'agit également du principe de base d'autres méthodes, pas forcément géodésiques, qui utilisent les modèles déformables, telles que les « snakes » (Kass et al., 1988). Certaines de ces contraintes peuvent évidemment être l'expression de connaissances expertes relatives au cadre applicatif pour lequel est utilisée la technique de segmentation (Niessen et al., 1998). Dans leur état de l'art sur les modèles déformables, Hegadi, Kop et Hangarge (2010) stipulent d'ailleurs que l'intégration de connaissances a priori dans un modèle déformable peut à un résultat plus robustes et précis.

Nous allons pouvoir maintenant rentrer dans le détail des méthodes s'appuyant sur les level set. Notons d'abord que cette technique d'évolution d'un contour $\Gamma(t)$ repose sur l'application d'un champ de vitesse/force $F(t)$ (Sethian, 2002; Shi & Karl, 2005b)(**Figure 186 (a)**). Ce champ est fonction d'un paramètre temporel t , ce qui permet de maîtriser l'évolution/les mouvements du contour $\Gamma(t)$, défini par l'équation différentielle suivante :

$$\frac{d\Gamma(t)}{dt} = F(t) \vec{N} \quad \text{où } \vec{N} \text{ est la vecteur normal à l'extérieur de } \Gamma(t)$$

Ceci se traduit graphiquement par le fait que ce champ de vitesse/force $F(t)$ est normal au contour $\Gamma(t)$ défini dans l'espace \mathbb{Z}^n , discrétisation de \mathbb{R}^n . Etant donné que nous travaillons avec des vidéos, c'est-à-dire des successions d'images en 2D, dans la suite de ce document nous fixerons $n = 2$, et à ce titre nous pourrions utiliser le terme « courbe » pour qualifier le contour inscrit dans l'image $I: (A = (A_x, A_y) \subset \mathbb{Z}^2) \rightarrow (B \subset \mathbb{Z})$ (où, en particulier, $B = \{i_1 = 0, i_2 = 85, i_3 = 170, i_4 = 255\}$ suite à la segmentation en quatre niveaux de gris décrite précédemment). En effet, avec l'image en nuances de gris dont nous disposons, utiliser l'intensité du gris est un moyen simple et efficace de se servir des informations intrinsèques de l'image pour déterminer ce contour (Kass et al., 1988). Cette dernière constatation peut surprendre dans la mesure où elle intègre, à notre méthode de contours actifs géodésiques, des éléments issus de l'étude des contours actifs paramétriques. Cependant, une telle utilisation est légitime car ces différents types de modèles déformables peuvent partager leurs définitions des vitesses/forces externes (He et al., 2008; Hegadi et al., 2010). Comme précédemment énoncé, nous ne comptons pas utiliser de méthodes trop compliquées ni coûteuses en termes de calculs, ce qui empêcherait de les utiliser pour des traitements en temps réel (Zhang, 2006). Nous allons donc utiliser la méthode de level set proposée par Shi, en 2005, qui repose sur l'idée de compétition entre régions (Zhu & Yuille, 1996) : la pupille, région que nous voulons circonscrire, sera à distinguer du reste de l'image. Avec cette technique, nous ne chercherons jamais à résoudre des équations aux dérivées partielles et nous n'utiliserons pratiquement que des opérateurs d'ajout et de suppression d'objets dans des listes. Il devient alors possible de réaliser simplement et efficacement un algorithme en temps réel de traitement d'images par level set (Shi & Karl, 2005a). Pour illustrer nos propos, nous pouvons nous aider de la **Figure 186**.

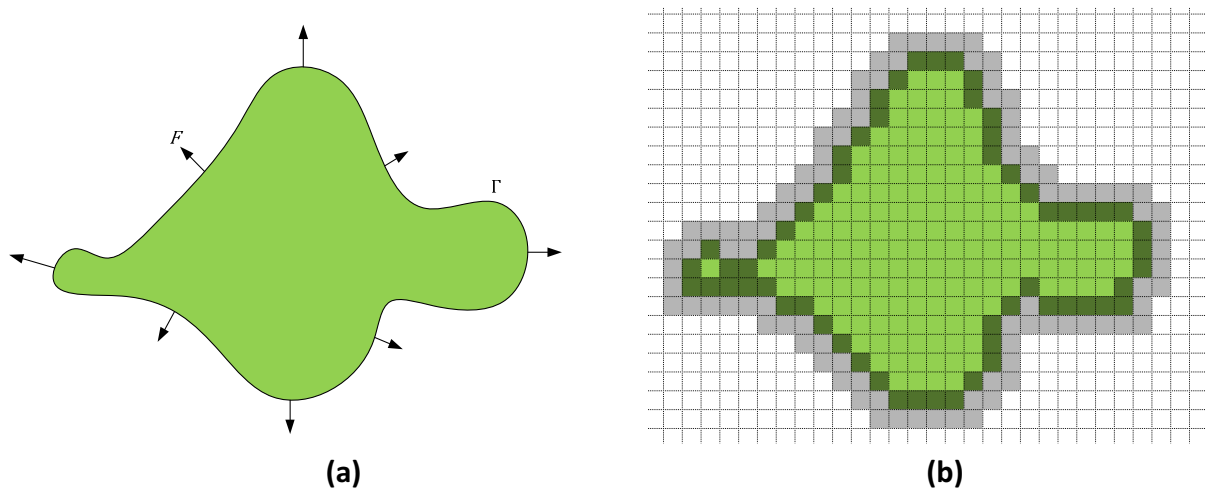


Figure 186 : La courbe $\Gamma(t)$ **(a)** en évolution dans \mathbb{R}^2 **(b)** puis discrétisée dans \mathbb{Z}^2

Premièrement, signalons que la **Figure 186 (b)** permet la caractérisation des différents types de points/pixels composant l'image I :

- Les points à l'intérieur de l'objet, qui constituent le « premier plan », sont soit en **vert clair** soit en **vert foncé** s'ils bornent le contour recherché : on notera cet ensemble X .
- Les points à l'extérieur de l'objet, qui constituent le « fond », sont soit en **blanc** soit en **gris clair** s'ils bornent le contour recherché : on notera cet ensemble \bar{X} .

Au lieu d'étudier dans le temps le front du contour $C(x)$, tel que $\Gamma(t) = \{x \in (A \subset \mathbb{Z}^2) \mid C(x) = t\}$ dont l'évolution serait conditionnée par $|\nabla C(x)|F(t) = 1$, l'idée fondamentale des méthodes s'appuyant sur des level set est d'intégrer la courbe recherchée à une fonction de dimension supérieure : $\phi(x, t)$ (Angelini et al., 2005). On peut également supposer que la courbe $\Gamma(t)$ est représentée implicitement en tant que level set de niveau zéro d'une fonction lipschitzienne $\phi(x, t)$. Ainsi, à un pas de temps donné, chaque point/pixel x est soit à l'intérieur soit à l'extérieur du contour en fonction du signe de $\phi(x, t)$. Cette fonction peut, par conséquent, arbitrairement être définie de la sorte (Han et al., 2003; Shi & Karl, 2005b) :

$$\forall x \in (A \subset \mathbb{Z}^2), \begin{cases} \phi(x, t) < 0, \text{ si } x \in \mathbb{Z}^2(\Gamma(t)) \\ \phi(x, t) > 0, \text{ sinon} \end{cases} \quad \text{où } (\mathbb{Z}^2(\Gamma(t)) = X) \subset \mathbb{Z}^2 \text{ est le sous-espace délimité par } \Gamma(t)$$

Plus particulièrement, pour notre algorithme, en respectant les codes de couleur appliqués à la **Figure 186 (b)**, nous utiliserons les valeurs suivantes pour $\phi(x, t)$:

- Pour les points à l'intérieur de l'objet, c'est-à-dire en **vert clair** : $\phi(x, t) = -3$.
- Pour les points à l'intérieur de l'objet et qui bornent le contour, c'est-à-dire en **vert foncé** : $\phi(x, t) = -1$.
- Pour les points à l'extérieur de l'objet, c'est-à-dire en **blanc** : $\phi(x, t) = 3$.
- Pour les points à l'extérieur de l'objet et qui bornent le contour, c'est-à-dire en **gris clair** : $\phi(x, t) = 1$.

La fonction $\phi(x, t)$ étant par définition continue, le théorème des valeurs intermédiaires nous permet d'affirmer que le contour que nous recherchons passe entre deux pixels pour lesquels $\phi(x, t)$ change de signe. Effectivement, le contour recherché étant le level set de niveau zéro de la fonction $\phi(x, t)$, il va satisfaire la condition :

$$\phi(\Gamma(t), t) = 0$$

L'évolution de la courbe $\Gamma(t)$ débute nécessairement avec un contour initial, et nous pouvons poser $\Gamma(0) = \Gamma_0$. On peut ainsi modifier l'équation précédente de la sorte :

$$\phi(\Gamma_0, 0) = 0$$

De la même manière, nous pouvons poser $\phi(x, 0) = \phi_0(x)$. Avec cette notation, l'équation décrivant la fonction lipschitzienne à l'état initial devient :

$$\phi_0(\Gamma_0) = 0$$

On remarquera que, pour des raisons d'efficacité, comme dans la méthode dite du « fast marching » (Sethian, 1999), il est évident qu'effectuer des calculs sur l'ensemble du domaine de calcul est inutile, et qu'il est plus efficace d'effectuer des calculs dans un voisinage du level set initial (Hegadi et al., 2010; Sethian, 2002). Ainsi, nous ne calculerons pas l'intégralité de la fonction $\phi(x, t)$ à chaque instant. Nous nous contenterons, à un instant t fixé, c'est-à-dire consécutivement à une itération donnée, d'en faire une mise à jour dans un voisinage du contour en évolution. Ce genre d'approches minimise l'effet du bruit et des artefacts, qui surviennent généralement au cours d'un suivi, et elle accroît la stabilité de la solution (Yilmaz et al., 2004). La segmentation est alors accomplie en plaçant le contour initial Γ_0 dans le domaine de l'image et en lui permettant de se déformer en fonction d'une équation décrivant son évolution (Hegadi et al., 2010). Pour analyser dynamiquement cette évolution de $\phi(x, t)$, il faut classiquement résoudre l'équation aux dérivées partielles suivantes (Sethian, 2002; Shi & Karl, 2005a) :

$$\frac{d\phi(x, t)}{dt} + F(t)|\nabla\phi(x, t)| = 0$$

Cette résolution est fréquemment accélérée à l'aide d'une méthode par bande étroite, qui permet de ne travailler que sur un voisinage du contour, ou par fast marching, qui consiste à résoudre une équation générale statique de type Hamilton-Jacobi (Sethian, 1999). Ce second type de résolution passe par l'approximation d'une solution de cette équation grâce à une interpolation polynomiale (Osher & Sethian, 1988). Concernant notre application de la méthode de level set, l'algorithme particulièrement rapide utilisé consiste à faire évoluer la courbe à l'aide d'opérations simples sans résoudre d'équations aux dérivées partielles (Shi & Karl, 2005b).

Par ailleurs, afin de lisser en douceur les bords de la courbe $\phi(x, t)$, et d'accessoirement accélérer la procédure, un filtre de Gauss isotrope est aussi mis en place (pour plus de détails concernant le filtrage par convolution, consulter (Nixon & Aguado, 2002)). Il est, en effet, important d'avoir conscience que le bruit, présent sur les images originellement capturées, peut, consécutivement au seuillage, produire des groupes de pixels non-connectés (Preston, 1981). De tels filtres ont déjà fait la preuve de leur efficacité pour étouffer un tel bruit dans de précédentes implémentations de méthodes de level set (pour des exemples, consulter (Banerjee & Bhattacharya, 2010; Caselles et al., 1997; Shi & Karl, 2005a; Zhang et al., 2010)). On peut également remarquer que l'utilisation d'un filtre gaussien va avoir tendance à pousser le contour à prendre une forme elliptique (Xu et al., 2001), aidant donc le contour à tendre vers la forme souhaitée. Avec ces optimisations algorithmiques nous comptons, comme dans la réalisation de Shi et Karl de 2005, pouvoir réaliser le suivi d'un objet en temps réel, qui sera la pupille de l'utilisateur dans notre cas. Mathématiquement, l'ajout de ce filtrage revient à décomposer le champ de vitesse/force F en deux sous-champs (Shi & Karl, 2005a) :

- F_{ext} , un champ de vitesse/force représentant la compétition entre zones s'appuyant sur les données de l'image I , et caractérisées par les valeurs de $\phi(x, t)$.
- F_{int} , un champ de vitesse/force lissant la courbe $\Gamma(t)$ grâce à ses propres propriétés, à l'aide d'un filtre de Gauss.

Cette décomposition permet de considérer de manière indépendante les forces conditionnées par les données brutes de l'image, à travers les valeurs F_{ext} , et celles adjointes comme terme de régulation, à travers les valeurs F_{int} (Montagnat et al., 2001). Nous pouvons donc poser $F(t) = F_{int}(t) + F_{ext}(t)$, ce qui permet de réécrire l'équation d'évolution de la courbe Γ de la façon suivante :

$$\frac{d\Gamma(t)}{dt} = (F_{int}(t) + F_{ext}(t)) \vec{N}$$

avec

$$F_{ext}(t) = i_2 - I(x) \quad \text{où } i_2 \text{ est le niveau de gris le plus sombre après seuillage}$$

$$F_{int}(t) = G_{\sigma,K}(x) * \phi(x, t) \quad \text{où } \forall x = (u_x, v_x) \in (A \subset \mathbb{Z}^2), G_{\sigma,K}(x) = \frac{1}{K^2} \sum_{y \in B} e^{-\frac{u_y^2 + v_y^2}{2\sigma^2}},$$

$$\text{et } B = \left(\left[u_x - \frac{K-1}{2}, u_x + \frac{K-1}{2} \right], \left[v_x - \frac{K-1}{2}, v_x + \frac{K-1}{2} \right] \right) \subset A$$

Cette définition de $F_{ext}(t)$ nous permet de préciser que l'intérieur de l'objet correspond à une des parties les plus sombres de l'image. En outre, cette fonction ne dépend que de l'image traitée, mais il serait possible d'améliorer sa définition pour qu'elle intègre un gradient qui faciliterait la convergence du level set (par exemple, consulter (Iwashita et al., 2004)).

La définition de $F_{int}(t)$, quant à elle, lisse le contour et permet à la fonction d'évoluer plus rapidement vers un état stable. D'après de précédentes implémentations que nous avons étudiées, nous pouvons préciser les paramètres du noyau de convolution gaussien $G_{\sigma,K}(x)$ et les valeurs qu'ils prennent généralement dans un contexte applicatif similaire au nôtre (Shi & Karl, 2005c; Zhang et al., 2010) :

- σ est l'écart type du filtre, et prendra une valeur comprise entre 0.8 et 1.5.
- K est la taille du filtre, et prendra une valeur dans $\{3, 5, 7\}$.

On notera que, à chaque passe de l'algorithme, nous allons appliquer itérativement ce filtre N_{int} fois. Si nous n'avons pas imposé des contraintes de préservation topologique, cela aurait pour effet de faire disparaître tous les trous résiduels de rayons inférieurs ou égaux à N_{int} (Preston, 1981; Shi & Karl, 2005a).

D'un point de vue algorithmique, la méthode que nous employons va consister à réaliser deux types de passes différentes : une pour chacune des deux forces $F_{ext}(t)$ et $F_{int}(t)$ (Shi & Karl, 2005a). Il est ainsi possible d'approximer le processus d'évolution de la courbe du level set sans pour autant avoir à résoudre d'équations aux dérivées partielles. En effet, pour la segmentation d'images et les applications de suivi d'objets, ce qui nous intéresse c'est le contour de l'objet final et non la précision du processus d'évolution courbe (Zhang, 2006). Ainsi, pour chaque image, nous repartirons des résultats du suivi de la dernière image plutôt que des courbes initiales. De cette façon, notre méthode de segmentation devient une véritable méthode de suivi d'objet (Yilmaz et al., 2004). Ce suivi permet d'ailleurs

d'économiser du temps de calcul, puisque la détection complète d'une caractéristique de l'œil, en l'occurrence du contour de la pupille, à chaque trame est très consommatrice de cette ressource (Asteriadis et al., 2007). Globalement, notre méthode consiste à faire évoluer Γ , afin de localiser les contours de la pupille dans l'image courante, puis à passer au traitement de l'image suivante de la vidéo quand l'évolution est stable ou que le nombre maximum d'itérations est atteint (Shi & Karl, 2005c). Pour cela, nous allons utiliser deux listes : L_{in} et L_{out} , qui servent à déterminer un encadrement du contour recherché. Chacune de ces deux listes va disposer d'une procédure permettant de « pousser l'autre liste » et ainsi de faire évoluer le contour. Ces procédures vont avoir pour effet de déplacer un point/pixel d'une liste à l'autre. Elles sont définies de la façon suivante (Shi & Karl, 2005a) :

- $check_in(x)$, qui déplace un point/pixel x de L_{out} vers L_{in} :
 1. Ajouter x à L_{in} , supprimer x de L_{out} et fixer $\phi(x, t) = -1$
 2. Ajouter à L_{out} tous les voisins de x tels que $\phi(x, t) = 3$, et fixer $\phi(x, t) = 1$
- $check_out(x)$, qui déplace un point/pixel x de L_{in} vers L_{out} :
 1. Ajouter x à L_{out} , supprimer x de L_{in} et fixer $\phi(x, t) = 1$
 2. Ajouter à L_{in} tous les voisins de x tels que $\phi(x, t) = -3$, et fixer $\phi(x, t) = -1$

Ces procédures utilisent la dénomination de « voisins », ce qui nous oblige maintenant à définir la notion de voisinage. Effectivement, pour éviter tout paradoxe ou ambiguïté topologique (Hilitch, 1969; Rosenfeld, 1970, 1974), il va nous falloir choisir une règle de connectivité cohérente en 2D, soit (4, 8) soit (8, 4) (pour une revue sur la topologie numérique, consulter (Kong & Rosenfeld, 1989)). C'est-à-dire que, d'une façon complémentaire, les points/pixels à l'intérieur du contour vont avoir soit 4 soit 8 voisins (**Figure 187**), et que les points/pixels du « fond » vont respectivement en avoir soit 8 soit 4. Effectivement, on aimera, en général, se ramener au cas d'une image binaire pour laquelle il existe deux types de points/pixels (Bertrand, Everat, & Couprie, 1997) :

- Ceux appartenant au « premier plan », typiquement l'objet d'intérêt de l'image : ces points/pixels constitueront l'ensemble X . Cet ensemble est fini.
- Ceux appartenant à « l'arrière plan » ou au « fond », correspondant à tout ce qui ne fait pas partie de l'objet d'intérêt : ces points/pixels constitueront l'ensemble \bar{X} . Cet ensemble contient au moins une composante infinie incluant les points/pixels appartenant \bar{A} , c'est-à-dire ceux à « l'extérieur de l'image » qui appartiennent de facto à « l'arrière plan ».



Figure 187 : Enveloppes des voisinages géodésiques d'un point/pixel x (a) 4-connecté et (b) 8-connecté

Pour clairement illustrer la signification de ces règles, nous rappelons que les images sont définies sur une grille orthogonale en 2D. Dans ce contexte, un pixel x est un carré fermé (voir **Figure 186 (b)**), dont les arêtes sont de longueur 1 et parallèles aux axes des coordonnées, et les intersections sur la grille ont des coordonnées entières (Klette, 2003).

Comme discuté par Preston en 1981, la définition du voisinage n'est pas anodine et va conditionner le résultat des opérations logiques qui vont y être effectuées. En effet, il est avéré que vérifier les pixels appartenant à l'objet lors d'une détection de contour est plus rapide si nous cherchons un contour 8-connecté ou si moins de 50% des pixels de l'image appartiennent à l'objet (van Vliet & Verwer, 1988). Nous allons donc utiliser la règle de connectivité (8, 4) qui nous permet de borner le contour recherché avec moins de points qu'avec la règle de connectivité (4, 8) (tel que représenté sur les **Figure 186 (b)** et **Figure 188**). Ce choix prendrait surtout un intérêt si l'on décide de ne travailler qu'avec la seule liste L_{in} , ce qui est rendu possible par sa complémentarité avec L_{out} (Shi & Karl, 2005a). Dans ce cas, cette liste contiendra le moins possible de points nécessaires pour décrire le contour et elle engendrera donc le moins de calculs possible. Il est théoriquement possible d'obtenir ainsi un gain de performance de l'ordre de 50%.

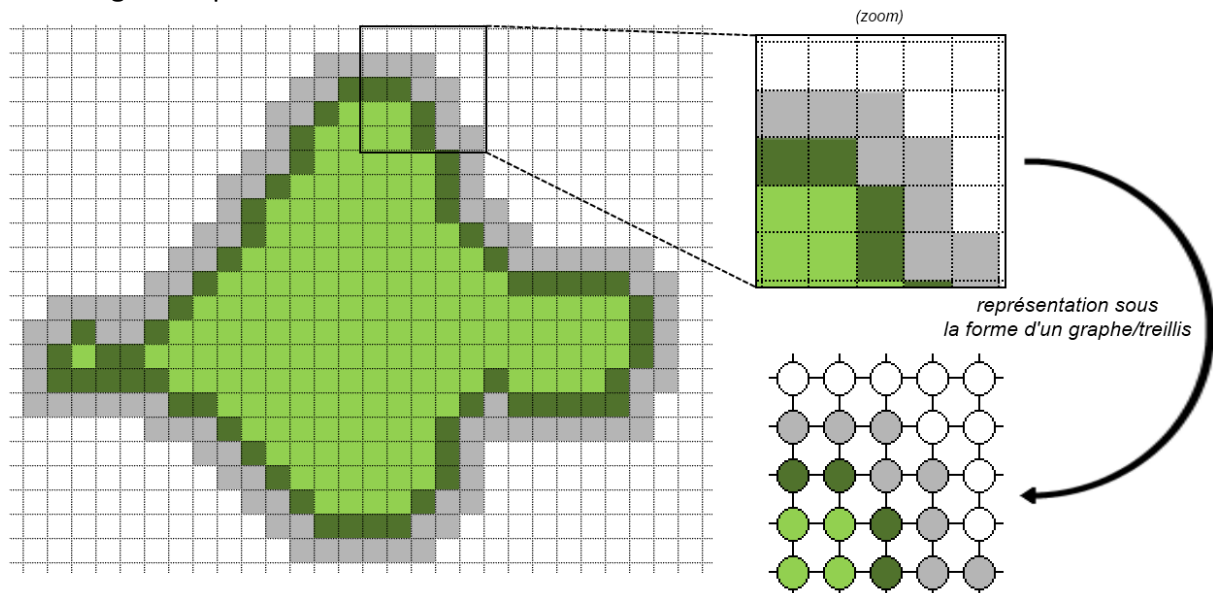


Figure 188 : Représentation d'une image numérique sous forme de graphe

La prise en compte de cette règle de connectivité va également nous permettre de préciser le concept de préservation topologique que nous avons précédemment évoqué (Han et al., 2003). Effectivement, nous avons intégré à notre méthode une vérification au niveau du pixel/point permettant de garantir la conservation topologique du ou des modèles déformables initialement définis (pour plus de détails sur la classification topologique de points en 3D, consulter (Bertrand & Malandain, 1992)). Dans leurs travaux, Han et al. (2001) ont proposé d'ajouter à une méthode de level set existante une analyse consistant à déterminer si un point/pixel peut ou non être changé de zone sans changer la topologie des zones impliquées. Pour cela, nous allons vérifier les changements potentiels de signe de $\phi(x, t)$ au cours de son évolution à l'aide d'une règle simple : le signe peut être effectivement changé si, et seulement si, le point/pixel concerné par ce changement est un point simple (pour une définition du critère de simplicité, consulter (Bertrand, 1994; Kong & Rosenfeld, 1989)). En effet, la particularité intéressante d'un tel point est que son ajout ou

sa suppression d'un ensemble de points ne change pas la topologie de ce dernier (Han et al., 2003; Kong, 1997; Rosenfeld et al., 1998). Nous allons donc évaluer ce critère chaque fois que nous comptons retirer un point/pixel d'une des deux listes L_{in} et L_{out} , et, le cas échéant, ne pas procéder à cette modification si elle devait provoquer un changement topologique. Pour réaliser ces analyses, impliquant des voisinages de pixels, nous utiliserons une représentation en graphe/treillis comme représentée sur la **Figure 188**. D'autre part, on pourra aussi remarquer que cette représentation est bien adaptée car nous travaillons dans un environnement discrétisé, ainsi le contour que nous cherchons à déterminer ne va pas statistiquement passer par un point/pixel mais entre des points/pixels.

Pour parler du critère de simplicité d'un point/pixel x , il faut définir deux concepts (Bertrand, 1994):

- Le voisinage géodésique d'ordre k , relativement soit à X soit à \bar{X} , est l'ensemble des voisins de x à une distance géodésique de x inférieure ou égale à k (pour approfondir la notion d'ordre de connectivité voir (Rosenfeld, 1970)). On le notera $N_n^k(x, X)$ ou $N_n^k(x, \bar{X})$ selon qu'il s'agit d'un voisinage relatif à X ou à \bar{X} . Dans le cas relatif à X , il est par exemple défini récursivement de la sorte :

$$\forall x \in A, \begin{cases} N_n^1(x, X) = N_n^*(x) \cap X \\ N_n^k(x, X) = \cup \{N_n(y) \cap N_n^*(x) \cap X, y \in N_n^{k-1}(x, X)\} \end{cases}$$

- Le nombre topologique correspond au nombre de composantes n (ou \bar{n})-adjacentes à x , appartenant à un de ses voisinages géodésiques. Il est noté $T_n(x, X)$ ou $T_{\bar{n}}(x, \bar{X})$ selon qu'il est relatif à X ou à \bar{X} . Dans le cas relatif à X , on le définit, selon la règle de connectivité, de la manière suivante :

$$T_4(x, X) = \text{Card} \left(C_4(x, N_4^2(x, X)) \right)$$

$$T_8(x, X) = \text{Card} \left(C_8(x, N_8^1(x, X)) \right)$$

où $C_n(x, X)$ est l'ensemble de toutes les composantes n -adjacentes à x et n -connectés de X , et $\text{Card}(\Omega)$ est l'opérateur permettant de dénombrer un ensemble Ω . Cette notion de voisinage géodésique peut être représentée par les éléments de la **Figure 187** à l'ordre 1.

Dans le plan 2D d'une image, en ayant fait le choix de la règle de connectivité choisie (8, 4) et en utilisant la même notation que Han et al. en 2003, un point/pixel x doit satisfaire seulement les deux conditions suivantes pour être un point simple (Bertrand et al., 1997; Bertrand, 1994; Rosenfeld et al., 1998) :

- Etre 8-connecté à une seule et unique composante de points/pixels intérieurs au contour de l'objet, appartenant au « premier plan », c'est-à-dire satisfaire l'équation $T_8(x, X) = 1$.
- Etre 4-connecté à une seule et unique composante de points/pixels extérieurs au contour de l'objet, appartenant au « fond », c'est-à-dire satisfaire l'équation $T_4(x, \bar{X}) = 1$.

Afin d'évaluer le critère de simplicité d'un point/pixel, nous parcourons son voisinage dans un ordre spécifique (**Figure 189**). Pendant ce parcours, nous compterons simplement le nombre de composantes différentes auxquelles il est connecté. En 2D, la notion de connectivité est d'ailleurs suffisante pour caractériser un point simple (Couprie & Bertrand, 2009).



Figure 189 : Représentation **(a)** des quatre parcours élémentaires d'un point/pixel et **(b)** des rotations permettant un de ces parcours

La base de l'image (X, Y) étant définie comme sur la **Figure 189 (b)**, nous pouvons préciser que, graphiquement, parcourir le voisinage du point/pixel x à l'aide de ces rotations revient à le parcourir dans le sens horaire. Ces transformations, composées d'une ou de deux transitions liées, vont ainsi nous permettre de réaliser un parcours élémentaire, puis le parcours élémentaire suivant, etc.. Si l'on considère une transition permettant de passer de y à y' (avec $y, y' \in N_8^*(x)$) :

- Si $I(y) \not\equiv I(y')$, alors $Card_x(N_8^*(x)) ++$ ou $Card_{\bar{x}}(N_8^*(x)) ++$ selon que, respectivement, $y \in X$ ou \bar{X} .
- Sinon rien.

La relation $\not\equiv$ permet de déterminer si deux points/pixels n'appartiennent pas tous les deux au même ensemble, soit X soit \bar{X} , est définie mathématiquement de la sorte :

$$\forall y, y' \in A, I(y) \not\equiv I(y') = \begin{cases} \text{faux} & \text{si } (y \wedge y') \in (X \oplus \bar{X}) \\ \text{vrai} & \text{sinon} \end{cases} \quad \text{où } \oplus \text{ est l'opérateur « ou-exclusif »}$$

On peut expliciter, dans notre espace euclidien du graphe/trellis, les deux rotations qui permettent de réaliser ces transitions et que nous utiliserons donc pour les parcours élémentaires :

$$R_{\frac{\pi}{4}}^+ = \sqrt{2}R_{\frac{\pi}{4}} = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \quad R_{\frac{\pi}{4}}^- = \frac{1}{\sqrt{2}}R_{\frac{\pi}{4}} = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

avec

$$R_{\frac{\pi}{4}} = \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} = \frac{\sqrt{2}}{2} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

On signalera que ces rotations vérifient bien :

$$R_{\frac{\pi}{2}} = R_{\frac{\pi}{4}}^- R_{\frac{\pi}{4}}^+ = R_{\frac{\pi}{4}}^+ R_{\frac{\pi}{4}}^- = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

De la sorte, nous pouvons précisément définir le passage d'un voisin à l'autre des manières suivantes :

$\forall x = (u_x, v_x) \in I, \exists y = (u_y, v_y) \in N_8^*(x)$, tel que :

- En 8-connecté, $\exists y' = (u_{y'}, v_{y'}) \in N_8^*(x)$, tel que :

$$\begin{aligned}
 R_{\frac{\pi}{2}}(y - x) = y' - x &\Leftrightarrow y' = R_{\frac{\pi}{2}}(y - x) + x \\
 &\Leftrightarrow \begin{pmatrix} u_{y'} \\ v_{y'} \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u_y - u_x \\ v_y - v_x \end{pmatrix} + \begin{pmatrix} u_x \\ v_x \end{pmatrix} \\
 &\Leftrightarrow \begin{pmatrix} u_{y'} \\ v_{y'} \end{pmatrix} = \begin{pmatrix} v_x - v_y + u_x \\ u_y - u_x + v_x \end{pmatrix}
 \end{aligned}$$

- En 4-connecté, $\exists y' = (u_{y'}, v_{y'}) \in N_8^*(x)$, tel que, soit :

$$\begin{aligned}
 R_{\frac{\pi}{4}}^+(y - x) = y' - x &\Leftrightarrow y' = R_{\frac{\pi}{4}}^+(y - x) + x \\
 &\Leftrightarrow \begin{pmatrix} u_{y'} \\ v_{y'} \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} u_y - u_x \\ v_y - v_x \end{pmatrix} + \begin{pmatrix} u_x \\ v_x \end{pmatrix} \\
 &\Leftrightarrow \begin{pmatrix} u_{y'} \\ v_{y'} \end{pmatrix} = \begin{pmatrix} u_y - u_x + v_x - v_y + u_x \\ u_y - u_x + v_y - v_x + v_x \end{pmatrix} \\
 &\Leftrightarrow \begin{pmatrix} u_{y'} \\ v_{y'} \end{pmatrix} = \begin{pmatrix} v_x - v_y + u_y \\ u_y - u_x + v_y \end{pmatrix}
 \end{aligned}$$

soit :

$$\begin{aligned}
 R_{\frac{\pi}{4}}^-(y - x) = y' - x &\Leftrightarrow y' = R_{\frac{\pi}{4}}^-(y - x) + x \\
 &\Leftrightarrow \begin{pmatrix} u_{y'} \\ v_{y'} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} u_y - u_x \\ v_y - v_x \end{pmatrix} + \begin{pmatrix} u_x \\ v_x \end{pmatrix} \\
 &\Leftrightarrow \begin{pmatrix} u_{y'} \\ v_{y'} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(u_y - u_x + v_x - v_y) + u_x \\ \frac{1}{2}(u_y - u_x + v_y - v_x) + v_x \end{pmatrix} \\
 &\Leftrightarrow \begin{pmatrix} u_{y'} \\ v_{y'} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} u_y + u_x + v_x - v_y \\ u_y - u_x + v_y + v_x \end{pmatrix}
 \end{aligned}$$

Notre traitement algorithmique est assuré par la procédure *isSimple(x)*. En pratique, cette procédure va prendre en entrée l'image I et les coordonnées d'un point/pixel $x = (u_x, v_x)$, et retournera **vrai** si x est un point simple. Pour cela, nous utiliserons plusieurs variables temporaires :

- $y' = (u_{y'}, v_{y'})$, le point/pixel en cours d'analyse.
- $y = (u_y, v_y)$, le dernier point/pixel analysé.
- $Card_X(N_8^*(x))$, le nombre de composantes du voisinage au sens de la 8-connectivité de x appartenant à X .
- $Card_{\bar{X}}(N_8^*(x))$, le nombre de composantes du voisinage au sens de la 8-connectivité de x appartenant à \bar{X} .

L'algorithme va se dérouler ainsi (G. Garcia & Barra, 2015) :

1. On détermine la position du point/pixel x dans l'image I en testant ses coordonnées. Ceci nous permet de savoir quels parcours élémentaires devront être effectués sur son voisinage. On accède ainsi aux coordonnées du premier point/pixel y_{start} et du dernier point/pixel y_{end} bornant cette partie du parcours global.
2. On initialise : $y' = y_{start}$, $y = 0$ (par convention on considérera qu'un tel point n'appartient ni à X ni à \bar{X}), $Card_X(N_8^*(x)) = 0$ et $Card_{\bar{X}}(N_8^*(x)) = 0$.
3. Si x n'est pas « à l'intérieur de l'image », alors on réalise un « parcours extérieur » constitué de deux transitions :
 - a. Si $I(y_{end}) \neq I(\bar{X})$ alors $Card_{\bar{X}}(N_8^*(x)) + +$.
 - b. Si $I(y_{start}) \neq I(\bar{X})$ alors $Card_X(N_8^*(x)) + +$.
4. On réalise un parcours élémentaire :
 - a. Si $I(y') \neq I(\bar{X})$ on réalise une transition au sens de la 8-connectivité à l'aide de $R_{\frac{\pi}{2}}$:
 - i. Si $I(y') \neq I(y)$ alors $Card_X(N_8^*(x)) + +$.
 - ii. On fixe $y = y'$.
 - iii. On calcule alors directement les coordonnées de y' :
 1. $u_{y'} = v_x - v_y + u_x$.
 2. $v_{y'} = u_y - u_x + v_x$.
 - b. Sinon (c'est-à-dire si $I(y') \neq I(X)$) on réalise successivement les transitions au sens de la 4-connectivité à l'aide de $R_{\frac{\pi}{4}}^+$ et $R_{\frac{\pi}{4}}^-$:
 - i. Si $I(y') \neq I(y)$ alors $Card_{\bar{X}}(N_8^*(x)) + +$.
 - ii. On fixe $y = y'$.
 - iii. On calcule alors directement les coordonnées de y' :
 1. $u_{y'} = v_x - v_y + u_y$.
 2. $v_{y'} = u_y - u_x + v_y$.
 - iv. Si $I(y') \neq I(y)$ alors :
 1. Si $I(y') \neq I(\bar{X})$ alors $Card_X(N_8^*(x)) + +$.
 2. Sinon (c'est-à-dire si $I(y') \neq I(X)$) alors $Card_{\bar{X}}(N_8^*(x)) + +$.
 - v. On fixe $y = y'$.
 - vi. On calcule alors directement les coordonnées de y' :
 1. $u_{y'} = \frac{1}{2}(u_y - u_x + v_x - v_y) + u_x$.
 2. $v_{y'} = \frac{1}{2}(u_y - u_x + v_y - v_x) + v_x$.

On retourne à l'étape 2 tant que $y' \neq y_{end}$.

5. On analyse la dernière transition réalisée, ainsi si $I(y') \neq I(y)$ alors :
 - a. Si $I(y') \neq I(\bar{X})$ alors $Card_X(N_8^*(x)) ++$.
 - b. Sinon (c'est-à-dire si $I(y') \neq I(X)$) alors $Card_{\bar{X}}(N_8^*(x)) ++$.

6. Si $(Card_X(N_8^*(x)) = 1 \wedge Card_{\bar{X}}(N_8^*(x)) = 1)$ retourner **vrai**, sinon retourner **faux**.

On remarquera que déterminer, pour une image en 2D, si un point/pixel est un point simple présente aussi un intérêt fondamental pour les méthodes d'amincissement, de grossissement et de translation dans le domaine de la reconnaissance de formes (Hilitch, 1969; Lohou & Bertrand, 2002; Rosenfeld et al., 1998; Young, Peverini, Verbeek, & van Otterloo, 1981). Ce critère peut également être utilisé en 3D (pour un algorithme rapide d'évaluation du critère de simplicité d'un point en 3D, consulter (Latecki & Ma, 1996)). On pourra de plus constater que, selon Hegadi et al. (2010), en disposant ainsi d'un modèle déformable possédant un grand contrôle sur sa topologie (en fonction de la prise en compte ou non du critère de point simple), nous répondons positivement à un défi de généralité des techniques de segmentation. D'autre part, au niveau de l'augmentation du temps de calcul, Han et al. en ont relevé au maximum une de 7% du temps total de calcul, ce qui, ramené aux performances annoncées par Shi et Karl pour leur algorithme de level set, signifierait une augmentation du temps de traitement de l'ordre d'une milliseconde. On peut donc considérer que l'intégration de ce critère de conservation topologique ne va pas ralentir significativement notre solution et ne devrait pas empêcher son application en temps réel.

En pratique, nous allons partir de l'image, résultante de notre précédent traitement, segmentée en quatre niveaux de gris sur laquelle nous allons identifier le contour de la pupille, c'est-à-dire la partie normalement la plus sombre de l'image. D'un point de vue algorithmique, la méthode va se décomposer de cette manière :

1. Initialiser $\phi(x, t) = \phi_0(\Gamma_0)$, $F_{ext}(t) = i_2 - I(x)$ et les deux listes L_{in} et L_{out} . Ici, nous prendrons pour contour initial Γ_0 , un cercle dont le centre est le centre de l'image avec pour rayon $r = \min(A_x, A_y)/10$. Dans le cadre du suivi oculaire, le contour finalement déterminé pour une image sera ensuite utilisé comme contour initial pour la suivante.

2. Faire évoluer, durant N_{ext} itérations, le contour implicite $\Gamma(t)$ soumis à $F_{ext}(t)$:
 - a. Parcourir L_{out} et pour chaque pixel x tel que $F_{ext}(t) > 0$ et $isSimple(x)$ est vraie : exécuter $check_in(x)$.

 - b. Eliminer les points inutiles de L_{in} , c'est-à-dire les pixels x de L_{in} dont tous les voisins y sont à l'intérieur du contour ($\phi(y, t) < 0$), et fixer $\phi(x, t) = -3$.

 - c. Parcourir L_{in} et pour chaque pixel x tel que $F_{ext}(t) < 0$ et $isSimple(x)$ est vraie : exécuter $check_out(x)$.

 - d. Eliminer les points inutiles de L_{out} , c'est-à-dire les pixels x de L_{out} dont tous les voisins y sont à l'extérieur du contour ($\phi(y, t) > 0$), et fixer $\phi(x, t) = 3$.

- e. Si une convergence de tous les pixels du contour, constitué de L_{in} et L_{out} , est constatée en accord avec le champ de vitesse/force appliqué $F_{ext}(t)$, alors passer à l'étape 3. Sinon, continuer d'itérer.
3. Faire évoluer, durant N_{int} itérations, le contour implicite $\Gamma(t)$ soumis à $F_{int}(t)$:
 - a. Parcourir L_{out} et pour chaque pixel x tel que $F_{int}(t) < 0$ et $isSimple(x)$ est vraie : exécuter $check_in(x)$.
 - b. Eliminer les points inutiles de L_{in} , c'est-à-dire les pixels x de L_{in} dont tous les voisins y sont à l'intérieur du contour ($\phi(y, t) < 0$), et fixer $\phi(x, t) = -3$.
 - c. Parcourir L_{in} et pour chaque pixel x tel que $F_{int}(t) > 0$ et $isSimple(x)$ est vraie : exécuter $check_out(x)$.
 - d. Eliminer les points inutiles de L_{out} , c'est-à-dire les pixels x de L_{out} dont tous les voisins y sont à l'extérieur du contour ($\phi(y, t) > 0$), et fixer $\phi(x, t) = 3$.
 - e. Si une convergence de tous les pixels du contour, constitué de L_{in} et L_{out} , est constatée en accord avec le champ de vitesse/force appliqué $F_{ext}(t)$, alors passer à l'étape 4. Sinon, continuer d'itérer.
 4. Le contour est considéré comme étant trouvé : il est encadré par les pixels, respectivement, de L_{in} et L_{out} .

La condition d'arrêt conditionne fortement le résultat (Niessen et al., 1998). Dans notre cas, cette condition est soit la convergence de tous les points du contour en accord avec le champ de vitesse/force appliqué $F_{ext}(t)$, soit si le nombre d'itérations maximum est atteint (Shi & Karl, 2005a). Bien que cette méthode soit très efficace, on pourra pointer plusieurs choses qui influent également sur le résultat final, à savoir:

- L'influence de l'initialisation Γ_0 , qui est encore plus importante dans le cas de la préservation topologique (Han et al., 2003).
- L'ordre de parcours des éléments des listes L_{in} et L_{out} , qui forcément conditionne l'ordre d'entrée et/ou de sortie de points/pixels dans ces dernières et, de ce fait, influence l'évolution du contour.

c. L'extraction des caractéristiques oculaires

Pour rappel, l'identification des caractéristiques de l'œil va nous permettre de déterminer l'axe optique, c'est-à-dire le vecteur normal à l'œil passant par le centre de la pupille (Hansen & Ji, 2010). Ainsi, en déterminant les paramètres des ellipses observées dans deux différents plans orthogonaux à l'aide de deux caméras, c'est-à-dire les paramètres de deux projections de la pupille, nous pouvons obtenir sa position, sa taille et son orientation dans l'espace (Kohlbecher et al., 2008). Il est alors relativement simple de déterminer le centre de la pupille et les différentes composantes du vecteur normal passant par ce centre. Grâce à la segmentation par level set que nous avons réalisée, nous disposons d'un ensemble de points/pixels composant le contour de la pupille de l'utilisateur. Afin d'extraire les caractéristiques de cette pupille, nous allons faire correspondre ce contour à une forme

connue pour laquelle l'extraction des propriétés géométriques sera aisée. Dans notre cas, et comme énoncé précédemment, il paraît judicieux de vouloir faire correspondre ce contour à une forme elliptique, voire à un cercle. Il va donc nous falloir mettre en place une méthode permettant d'ajuster le contour précédemment déterminé à une ellipse. On pourra également remarquer que ce type d'approche peut aussi être utilisé avec l'iris à la place de la pupille (pour un exemple d'application, consulter (Wang et al., 2005)).

Commençons par définir formellement ce qu'est une ellipse. Dans un plan donné, de repère (O, \vec{i}, \vec{j}) , une ellipse Γ peut être caractérisée de façon unique à l'aide de cinq paramètres (Ahn & Rauh, 1999)(Figure 190) :

- Les coordonnées de son centre : $C = (x_c, y_c)$.
- Les longueurs de ses demi-axes (grand et petit) : a et b , avec $a \geq b > 0$.
- L'angle entre l'axe des abscisses du plan de l'image et l'axe majeur de l'ellipse : α , avec $-90^\circ < \alpha \leq 90^\circ$.

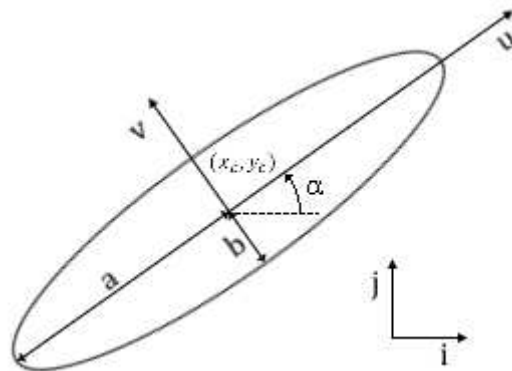


Figure 190 : Représentation d'une ellipse et de ses paramètres dans le plan (d'après (Ahn & Rauh, 1999))

Il est également possible de décrire canoniquement les points qui constituent l'ellipse Γ dans un nouveau repère centré sur son centre, (C, \vec{u}, \vec{v}) , tel que représenté sur la **Figure 190**. Dans ce nouveau repère, trois des paramètres précités vont disparaître (Ahn & Rauh, 1999). En effet, on peut maintenant définir l'ensemble des points constituant l'ellipse Γ , avec seulement a et b , de la façon suivante :

$$\forall (x', y') \in \Gamma, \frac{x'^2}{a^2} + \frac{y'^2}{b^2} = 1$$

avec

$$\begin{aligned} x' &= a \cos \alpha & y' &= b \sin \alpha \\ &= (x - x_c) \cos \alpha - (y - y_c) \sin \alpha & &= (x - x_c) \sin \alpha + (y - y_c) \cos \alpha \end{aligned}$$

Ainsi, dans le repère (O, \vec{i}, \vec{j}) en coordonnées homogènes, on peut exprimer matriciellement cette ellipse (Kohlbecher et al., 2008) :

$$\forall \vec{x} = (x, y) \in \Gamma, \vec{x}^T A \vec{x} = 0 \quad \text{où } A = S^T Q S$$

avec

$$S = \begin{pmatrix} \cos \alpha & -\sin \alpha & -x_c \cos \alpha + y_c \sin \alpha \\ \sin \alpha & \cos \alpha & -x_c \sin \alpha - y_c \cos \alpha \\ 0 & 0 & 1 \end{pmatrix} \quad Q = \begin{pmatrix} \frac{1}{a^2} & 0 & 0 \\ 0 & \frac{1}{b^2} & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

D'une manière générale, une ellipse peut également être vue comme une conique, c'est-à-dire une courbe plane définie comme l'intersection entre un cône de révolution et un plan, et être modélisée implicitement à l'aide du polynôme de second ordre suivant (Kanatani & Sugaya, 2007) :

$$\forall (x, y) \in (\Gamma \subset \mathbb{Z}^2), F(\vec{x}; \vec{\sigma}) = \vec{\sigma} \cdot \vec{x} = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

avec

$$\vec{\sigma} = (A \ B \ C \ D \ E \ F)^T \quad \vec{x} = (x^2 \ xy \ y^2 \ x \ y \ 1)^T$$

On notera qu'il faut également introduire une contrainte quadratique sur cette conique pour s'assurer que ce soit une ellipse, et non une parabole ou une hyperbole. Cette contrainte est de la forme suivante :

$$B^2 - 4AC < 0$$

Dans le repère (O, \vec{i}, \vec{j}) , il est également possible d'établir le lien entre les coefficients de cette expression polynomiale et les paramètres de l'ellipse de la sorte :

$$\begin{cases} x_c = \frac{2CD - BE}{B^2 - 4AC} \\ y_c = \frac{2AE - DB}{B^2 - 4AC} \\ a = \frac{\sqrt{2|q|\sqrt{B^2 + (A - C)^2} - 2q(A + C)}}{8} \\ b = \sqrt{a^2 - s^2} \\ \alpha = \frac{\tan^{-1}\left(\frac{B}{A - C}\right)}{2} \end{cases}$$

à l'aide du facteur de normalisation des coefficients, noté q , défini ainsi :

$$q = \frac{64F(4AC - B^2) - 64(AE^2 - BDE - CD^2)}{(4AC - B^2)^2}$$

et de la distance s , entre le centre de l'ellipse et un de ses points focaux, suivante :

$$s = \frac{\sqrt{|q|\sqrt{B^2 + (A - C)^2}}}{4}$$

D'un point de vue de l'approche adoptée pour déterminer les paramètres des ellipses, il existe deux grandes catégories de méthodes d'ajustement (Wynn & Stewart, 2005) :

- Les méthodes basées sur les limites/bords, par exemple :
 - Les transformations de Hough (par exemple, consulter (Yip et al., 1995)).
 - Les solutions analytiques (par exemple, consulter (De Paor, 1988)).
 - La méthode du cercle de Mohr (Lisle & Robinson, 1995).

- Les techniques des moindres carrés (Ahn & Rauh, 1999; Fitzgibbon et al., 1999)(pour un exemple d'utilisation, consulter (Wang et al., 2005)).
- Les méthodes basées sur les régions/zones (par exemple, consulter (Zunic & Sladoje, 2000)).

La première de ces catégories exploite le fait que les points d'un contour connu appartiennent au bord de l'ellipse. Les autres, quant à elles, consistent à calculer les moments de formes pour déterminer la forme correspondant le mieux au contour. On pourra d'ailleurs remarquer que ces méthodes basées sur les régions sont celles qui sont les moins sensibles aux irrégularités du contour, et donc au bruitage des données (Wynn & Stewart, 2005). Cependant, nous avons déjà déterminé l'ensemble de points/pixels constituant le contour de l'ellipse dont nous voulons déterminer les paramètres. Il nous paraît donc logique d'utiliser préférentiellement une méthode exploitant ce contour comme étant le bord de la forme recherchée. On pourra également signaler qu'il existe d'autres types de méthodes comme, par exemple, celles reposant sur l'utilisation d'algorithmes génétiques (par un exemple d'application, consulter (Ray & Srivastava, 2008)).

Parmi les méthodes qui s'appuient sur les limites, on notera qu'à la fois celles qualifiées d'analytiques et celles du cercle de Mohr (Lisle & Robinson, 1995) manquent cruellement de précision à cause de leur sensibilité au bruit, en particulier si peu de points sont utilisés (Wynn & Stewart, 2005). Ceci les rend donc de facto inutilisables pour notre usage. Dans notre cas, nous disposons d'un grand nombre de points/pixels, qui composent le contour établi par la méthode de level set, ce qui nous permet de faire appel à une méthode pouvant exploiter beaucoup de valeurs de ce type afin de ne plus souffrir du bruit. D'autre part, dans leur étude comparative de 2005, Wynn et Stewart montrent que la précision des méthodes qui utilisent les moindres carrés accroît avec l'augmentation du ratio a/b . Or, nous avons à analyser des images prises proches de l'œil, c'est-à-dire des images sur lesquelles la pupille apparaît comme une ellipse fortement excentrée (Świrski et al., 2012), c'est-à-dire comme une ellipse avec un fort ratio a/b . Ces différents éléments nous ont convaincu d'utiliser une méthode utilisant les moindres carrés. En particulier, nous avons décidé d'utiliser celle décrite par Fitzgibbon et al. (1999) car elle a été utilisée avec succès depuis de nombreuses années dans des applications de suivi oculaire (pour des exemples d'utilisation, consulter (Beymer & Flickner, 2003; Nouredin et al., 2005; Świrski et al., 2012; Wang et al., 2005; Zhiwei Zhu & Ji, 2005)).

A partir de la paramétrisation de l'ellipse présentée sur la **Figure 190**, nous voulons pouvoir déterminer, pour chaque image que nous analyserons, un paramétrage relativement précis de cette forme géométrique. La résolution d'un tel problème, dans le domaine discret de l'image analysée, revient à déterminer une fonction F satisfaisant l'équation implicite (Kanatani & Sugaya, 2007) :

$$\forall (x, y) \in (\Gamma \subset \mathbb{Z}^2), F(\vec{x}; \vec{\sigma}) = 0 \quad \text{où } \vec{\sigma} \text{ est un vecteur paramétrisant la fonction } F$$

Or, dans notre cas, nous rappelons que nous n'avons accès qu'à des valeurs observées. C'est-à-dire que les valeurs qui nous permettront de déterminer F sont soumises à une incertitude de mesure. Pour l'ensemble de l'image, nous disposons donc de valeurs de la forme :

$$\forall (x, y) \in (\Gamma \subset \mathbb{Z}^2), \bar{x} = \vec{x} + e_x \quad \text{où } e_x \text{ représente l'erreur de mesure, le bruit}$$

La méthode que nous allons utiliser doit donc amoindrir l'influence de cette incertitude, de cette erreur. Ainsi le problème revient à déterminer les valeurs de $\vec{\sigma}$ minimisant l'influence de cette erreur. Il peut alors être formulé comme un problème de moindres carrés :

$$\forall (x, y) \in (\Gamma \subset \mathbb{Z}^2), J(\vec{\sigma}) = \min \sum_{(x,y) \in \Gamma} F(\bar{x}; \vec{\sigma})^2$$

Cette minimisation va se faire d'une manière directe ou itérative, de façon à faire converger une estimation $\hat{\sigma}$ vers la véritable valeur de $\vec{\sigma}$ tandis que le bruit e_x décroît (Kanatani & Sugaya, 2007). Dans le cas d'une ellipse, le vecteur $\vec{\sigma}$ peut, par exemple, être composé des paramètres de la conique précédemment décrite, à savoir A, B, C, D, E et F . Le type de méthodes d'ajustement itératives permet alors d'estimer ses différentes composantes : $\hat{A}, \hat{B}, \hat{C}, \hat{D}, \hat{E}$ et \hat{F} . On remarque cependant que la méthode de Fitzgibbon et al. (1999), que nous comptons utiliser, est directe, c'est-à-dire qu'elle va permettre de déterminer sans estimation les valeurs de $\vec{\sigma}$. Cette méthode souffre néanmoins de limitations. Par exemple, elle peut être numériquement instable si l'erreur entre l'ensemble de points/pixels et l'ellipse théorique est trop faible. Cela a donné lieu à des propositions d'amélioration par introduction artificielle de bruit (par exemple, consulter (Greggio, Bernardino, Laschi, Dario, & Santos-Victor, 2010; Maini, 2006)). Dans notre cas, nous allons utiliser les deux listes qui encadrent le contour, L_{in} et L_{out} , ce qui évite d'avoir à introduire un tel bruit étant donné qu'aucun des points de ces listes n'est exactement, par définition, sur le contour de la pupille. D'autre part, pour augmenter la précision de cette méthode, il peut être judicieux de centrer les coordonnées de l'ensemble des n points/pixels considérés autour de leur centre de gravité (Wu, 2008). Avant de commencer l'ajustement d'ellipse, il leur sera ainsi appliqué les translations suivantes :

$$\delta_x = -\frac{\sum_{i=1}^n x_i}{n} \qquad \delta_y = -\frac{\sum_{i=1}^n y_i}{n}$$

Le cœur de la méthode consiste à minimiser $J(\vec{\sigma})$, qui est soumise, dans le cas d'une ellipse, à une contrainte quadratique, notée \mathcal{C} , en résolvant un système généralisé de valeurs propres. Pour procéder à cette résolution et afin d'exclure la solution triviale $\vec{\sigma} = (0 \ 0 \ 0 \ 0 \ 0)^T$ lors de ce calcul, la contrainte quadratique est modifiée de la sorte :

$$B^2 - 4AC = -1 \Leftrightarrow 4AC - B^2 = 1$$

Cela peut s'écrire matriciellement ainsi :

$$\vec{\sigma}^T \mathcal{C} \vec{\sigma} = 1$$

avec

$$\mathcal{C} = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Dans notre cas, la minimisation de $J(\vec{\sigma})$ repose sur une matrice, notée D , qui contient l'ensemble des points des listes L_{in} et L_{out} . Le système à résoudre est de la forme :

$$D^T D \vec{\sigma} = \lambda \mathcal{C} \vec{\sigma} \Leftrightarrow S \vec{\sigma} = \lambda \mathcal{C} \vec{\sigma}$$

avec

$$D = (\vec{x}_1^T \quad \dots \quad \vec{x}_i^T \quad \dots \quad \vec{x}_n^T)^T \quad S = D^T D \quad \lambda \in [\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6]$$

La résolution de ce système permet de déterminer un ensemble de six paires (λ_j, σ_j) , avec $j \in [1,6]$, en contenant l'ensemble des valeurs propres et vecteurs propres. Parmi ces vecteurs propres, un contient la valeur de $\vec{\sigma}$ permettant de minimiser $J(\vec{\sigma})$ (Fitzgibbon, Pilu, & Fisher, 1996) : il s'agit du vecteur propre, noté σ_+ , associé à l'unique valeur propre positive, notée λ_+ . Il devient alors possible de résoudre le système, qui devient :

$$\begin{cases} S\vec{\sigma} = \lambda C\vec{\sigma} \\ \vec{\sigma}^T C\vec{\sigma} = 1 \end{cases}$$

D'un point de vue algorithmique, cette méthode d'ajustement d'ellipse se déroule de la façon suivante :

1. Rassembler l'ensemble des points des listes L_{in} et L_{out} dans une seule liste L_Γ , calculer δ_x et δ_y puis appliquer ces deux translations à chaque point/pixel de L_Γ .
2. Calculer la matrice D : pour chaque (x_i, y_i) point/pixel appartenant à L_Γ , ajouter à D la ligne suivante $(x_i^2 \quad x_i y_i \quad y_i^2 \quad x_i \quad y_i \quad 1)$.
3. Calculer la matrice de dispersion S , telle que $S = D^T D$.
4. Résoudre l'équation $S\vec{\sigma} = \lambda C\vec{\sigma}$, et stocker l'ensemble des résultats, qui en sont respectivement les valeurs et vecteurs propres, dans une liste (λ_j, σ_j) où $j \in [1,6]$.
5. Chercher parmi les valeurs propres de λ_i celle qui est positive, λ_+ : le vecteur propre σ_i associé à cette valeur propre minimise alors $J(\vec{\sigma})$.
6. A partir des valeurs de σ_+ , calculer les paramètres de l'ellipse, c'est-à-dire x_c, y_c, a, b, α et les retourner.

On remarquera, cependant, qu'un certain nombre de points, qui sont supposés appartenir au contour présumé de la pupille, peuvent en réalité être erronés. A ce titre, estimer les caractéristiques de l'œil à partir d'un ajustement d'ellipse prenant en considération ces points peut mener à une relative imprécision. Ainsi, des méthodes ont été mises au point afin de contourner ce problème comme, par exemple, la méthode du RANSAC (Fischler & Bolles, 1981), qui cherche à exclure des calculs les valeurs échantillonnées présentant une erreur trop élevée. Cette dernière a d'ailleurs été utilisée avec succès par des systèmes de suivi oculaire afin de déterminer précisément les paramètres de la pupille (pour des exemples, consulter (Li et al., 2005; San Agustin Lopez, 2009)). Ultérieurement, il pourra être pertinent d'intégrer cette approche à l'algorithme que nous venons de présenter.

Consécutivement à l'application de cet algorithme sur le contour que nous avons précédemment encadré à l'aide de deux listes de points/pixels, les paramètres de l'ellipse formée par l'image de la pupille dans le plan de la caméra sont ainsi obtenus. En outre, comme exposé par Chi et al. en 2009, de nombreuses études se sont penchées sur la façon de détecter les caractéristiques 3D du globe oculaire en temps réel. Il s'avère que la technique de stéréovision donne des résultats permettant de déterminer correctement l'axe optique (Chi et al., 2009). En effet, la stéréoscopie est un procédé de reconstruction de profondeur à partir de deux images de la même scène, ainsi cette technique donne

directement accès à des paramètres 3D de la scène observée, dans notre cas l'œil de l'utilisateur. Procéder à une reconstruction 3D de l'œil par stéréoscopie est une technique qui repose sur la géométrie du système et qui permet généralement de s'affranchir de certains calibrages (Beymer & Flickner, 2003; Nagamatsu, Kamahara, & Tanaka, 2009). Cette technique a d'ailleurs montré son efficacité lorsqu'elle fait appel à de l'ajustement d'ellipses réalisé par deux caméras distinctes ayant fait l'objet d'un calibrage (Kohlbecher et al., 2008). De la sorte, nous comptons pouvoir accéder aux caractéristiques tridimensionnelles de la pupille à partir d'une reconstruction stéréoscopique s'appuyant sur deux images prises par deux caméras filmant l'œil de l'observateur selon deux axes différents. Ces caractéristiques pourront alors être exprimées dans le plan d'une des deux caméras utilisées pour cette reconstruction. Dans le cas de la reconstruction stéréoscopique de la pupille à l'aide de deux ellipses, une méthode possible est de voir le contour 3D de la pupille comme la section commune de deux cônes de révolution dont chacun trouve son sommet au niveau du point focal de leurs caméras respectives (Ma, 1993). Cette technique a déjà été utilisée efficacement par Kohlbecher et al. (2008) pour déterminer les caractéristiques tridimensionnelles de la pupille. Pour cela, il faut que les caméras utilisées aient été préalablement calibrées (voir la sous-sous-section **Le calibrage**). Au terme de cette procédure, nous comptons obtenir les coordonnées du centre de la pupille, noté O , ainsi que le vecteur normal à la pupille, noté \vec{V} , qui est le vecteur directeur de l'axe optique. Nous pouvons simplement définir ces deux éléments ainsi :

$$O = (x_o \quad y_o \quad z_o)^T \qquad \vec{V} = (x_{\vec{v}} \quad y_{\vec{v}} \quad z_{\vec{v}})^T$$

La méthode débute donc avec deux caméras calibrées. Ces dernières permettent de capturer dans leurs plans d'image 2D respectifs des éléments 3D selon la relation (pour $i \in [1,2]$) :

$$\vec{x}_i = R_i \vec{x}_{\text{œil}} + t_i \qquad \text{où } R_i = (r_{i1} \quad r_{i2} \quad r_{i3})$$

où $\vec{x}_i = (x_i \quad y_i \quad z_i)^T$ est un vecteur exprimé dans le repère de la caméra i et $\vec{x}_{\text{œil}}$ est le même vecteur exprimé dans le repère de l'œil. Les caméras sont statiques l'une par rapport à l'autre, et cette relation peut être explicitée à l'aide d'une matrice de rotations R et d'une translation t :

$$R_2 = RR_1 \qquad t_2 = Rt_1 + t$$

Dans ce contexte, les points dans le plan de la pupille de l'observateur peuvent être notés :

$$\vec{x}_{\text{œil}} = (x_{\text{œil}} \quad y_{\text{œil}} \quad 0)^T$$

On peut alors réécrire l'équation précédente, en coordonnées homogènes, sous la forme :

$$\vec{x}_i = G_i \vec{u}_{\text{œil}}$$

avec

$$\vec{u}_{\text{œil}} = (x_{\text{œil}} \quad y_{\text{œil}} \quad 1)^T \qquad G_i = (r_{i1} \quad r_{i2} \quad t_i)$$

Dans le plan de l'image de chaque caméra i , on peut poser (pour $i \in [1,2]$) :

$$\vec{u}_i = (u_i \quad v_i \quad 1)^T \qquad \text{où } u_i = \frac{x_i}{z_i} \text{ et } v_i = \frac{y_i}{z_i}$$

L'équation de changement de repère devient alors :

$$z_i \vec{u}_i = G_i \vec{u}_{\text{œil}}$$

Etant donné que, dans le plan de la pupille, une ellipse est définie par :

$$\overrightarrow{u_{\text{œil}}}^T Q_{\text{œil}} \overrightarrow{u_{\text{œil}}} = 0 \quad \text{où } Q_{\text{œil}} = \begin{pmatrix} \frac{1}{a_{\text{œil}}^2} & 0 & 0 \\ 0 & \frac{1}{b_{\text{œil}}^2} & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

Et ses projections dans les plans propres à chaque caméra i (pour $i \in [1,2]$) par :

$$\overrightarrow{u_i}^T A_i \overrightarrow{u_i} = 0$$

En utilisant la précédente équation de changement de repère, cette dernière expression peut alors s'écrire :

$$\overrightarrow{u_{\text{œil}}}^T G_i^T A_i G_i \overrightarrow{u_{\text{œil}}} = 0$$

Dans le repère de l'œil, comme il s'agit dans les deux cas de la même conique, on a ainsi :

$$G_i^T A_i G_i = k_i Q_{\text{œil}} \quad \text{où } k_i \text{ est un facteur d'échelle puisque que } \overrightarrow{u_i}^T k_i A_i \overrightarrow{u_i} = 0$$

Pour chaque caméra i ($i \in [1,2]$), comme nous connaissons l'ensemble des paramètres x_{c_i} , y_{c_i} , a_i , b_i et α_i , la résolution du système d'égalités formé par cette dernière équation doit nous permettre d'établir les valeurs de G_i , afin d'accéder aux valeurs de R_i et t_i . Pour cela, il est possible de commencer cette résolution sur la sous-matrice supérieure gauche du système (Ma, 1993). Ceci revient à résoudre en premier :

$$(R_i^T A_i R_i)_{\{1,2\}\{1,2\}} = k_i Q_{\text{œil}\{1,2\}\{1,2\}} \quad \text{car } G_{i\{1,2\}\{1,2\}} = R_{i\{1,2\}\{1,2\}} \text{ par définition}$$

A l'aide de la relation permettant de passer d'un repère d'une caméra à l'autre, nous pouvons réécrire cela ainsi :

$$(R_1^T (A_1 - k R^T A_2 R) R_1)_{\{1,2\}\{1,2\}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{où } k = \frac{k_1}{k_2}$$

Ce critère de nullité, de cette sous-matrice supérieure gauche d'une matrice 3×3 , signifie que :

$$\det(A_1 - k R^T A_2 R) = 0 \Leftrightarrow \det((R^T A_2 R)^{-1} A_1 - k I_3) = 0$$

Nous pouvons en déduire que k est une valeur propre de la matrice $(R^T A_2 R)^{-1} A_1$. Il est maintenant possible de calculer k , et le système peut alors être réécrit sous la forme :

$$(R_1^T C R_1)_{\{1,2\}\{1,2\}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{avec } C = A_1 - k R^T A_2 R$$

A ce stade, nous pouvons résoudre R_1 à l'aide de sa valeur propre nulle, qui est notée λ_3 et qui existe car $\det(C) = 0$, et de l'égalité $G_1^T A_1 G_1 = k_1 Q_{\text{œil}}$. En particulier, la définition de $Q_{\text{œil}}$ nous permet d'affirmer que :

$$r_{11}^T A_1 r_{12} = 0$$

Ainsi, à l'aide des autres valeurs propres de C , notées λ_1 et λ_2 , et des vecteurs propres qui leurs sont associés, notés s_1 et s_2 , la dernière colonne r_{13} peut être obtenue ainsi (pour plus de détails concernant cette résolution, consulter (Ma, 1993)) :

$$r_{13} = \pm \frac{\delta_{\lambda_1, \lambda_2}}{\|\delta_{\lambda_1, \lambda_2}\|} \quad \text{où } \delta_{\lambda_1, \lambda_2} = \sqrt{|\lambda_1| s_1} \pm \sqrt{|\lambda_2| s_2}$$

Il y a ainsi quatre valeurs possibles pour r_{13} . Etant donné que nous voulons que le vecteur normal au plan contenant cette ellipse, qui est le contour en 3D de la pupille de l'observateur dans notre cas, pointe vers les caméras, il faut retenir la solution pour laquelle la composante selon l'axe z de cette colonne est positive, c'est-à-dire que la solution doit respecter $r_{13\{3\}} > 0$ et $r_{23\{3\}} > 0$, avec $r_{23} = Rr_{13}$ (Kohlbecher et al., 2008).

Nous pouvons alors déterminer les valeurs des colonnes r_{11} et r_{13} qui sont les vecteurs propres associés aux valeurs propres non nulles de la matrice $A_1 - r_{13}^T r_{13} A_1$, et nous disposons maintenant de l'ensemble des valeurs de R_1 . Les paramètres encore indéterminés peuvent alors être calculés :

$$\begin{aligned}
 & - R_2 = RR_1 \\
 & - t_1 = \begin{pmatrix} r_{11}^T A_1^T \\ r_{11}^T A_1^T \\ r_{21}^T A_2^T R \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 0 \\ -r_{21}^T A_2^T t \end{pmatrix} \\
 & - k_1 = -t_1^T A_1 t_1 \text{ et } k_2 = \frac{k}{k_1} \\
 & - a_{pupille} = \sqrt{\frac{k_1}{r_{11}^T A_1 r_{11}}} \text{ et } b_{pupille} = \sqrt{\frac{k_1}{r_{12}^T A_1 r_{12}}}
 \end{aligned}$$

Dans notre cas, seul le calcul de t_1 présente un intérêt.

Le vecteur normal à la pupille \vec{V} est l'axe z de la base vectorielle du repère de l'œil défini, par rapport au repère de la caméra 1, par la rotation $R_1 = (r_{11} \ r_{12} \ r_{13})$ qu'on pourra maintenant noter $R_{\text{œil}}$. Le centre de la pupille O est situé à t_1 , dans le repère de coordonnées de la caméra 1, du centre optique de cette dernière : à l'avenir, on pourra noter ce vecteur $T_{pupille}$.

Bien que Kohlbecher et al. en 2008 présentent leur méthode de reconstruction de la pupille en 3D comme une méthode permettant de s'affranchir d'un calibrage, ceci n'est pas possible. En effet, cette affirmation se base sur la supposition que l'axe optique et l'axe visuel sont parallèles, ce qui est faux (Hansen & Ji, 2010; Nouredin et al., 2005) et qui mènerait à une inacceptable perte de précision, de l'ordre de 4-6° (Villanueva et al., 2007), au regard des mouvements que nous comptons pouvoir observer (Nagamatsu et al., 2009). Pour pallier à cela, un calibrage à l'aide d'un unique point peut être réalisé afin de déterminer l'angle α entre l'axe optique et l'axe visuel à l'origine de cette erreur d'estimation du point de regard (Nagamatsu et al., 2008).

2. La détection de la tête de l'utilisateur

Si l'on cherche à établir où un utilisateur regarde, déterminer la position et l'orientation de sa tête permet d'avoir une estimation grossière de la direction du vecteur de regard (Al-Rahayfeh & Faezipour, 2013). En effet, lorsqu'une personne porte son attention visuelle sur quelque chose, cet objet se trouve forcément dans son champ visuel, c'est-à-dire dans un espace situé face à l'utilisateur (**Figure 182**). Dans le cas où l'on souhaite réaliser un suivi oculaire, en laissant à l'utilisateur une certaine liberté en ce qui concerne les mouvements de sa tête, connaître la position de cette dernière permet de compenser le suivi fait à l'aide de l'observation seule de l'œil (pour un exemple d'utilisation, consulter (Al-Edaily, Al-Wabil,

& Al-Ohali, 2013b)). Cet espace est d'ailleurs centré autour d'un axe passant par le nasion, à savoir l'intersection de l'os frontal et des os propres du nez qui se trouve directement entre les yeux. De la sorte, il est possible par la simple connaissance de la position et l'orientation de la tête d'un utilisateur d'estimer l'endroit qu'il observe en tirant un pseudo-vecteur de regard passant par le nasion (pour un exemple, consulter (Asteriadis et al., 2007)), tel qu'illustré sur la **Figure 191**. En outre, étant donné qu'à partir de différentes positions de la tête il est possible de regarder au même endroit, un système de suivi oculaire qui ne tient pas compte de la position de la tête de l'observateur souffre nécessairement d'un manque d'invariance à celle-ci (pour un exemple, consulter (San Agustin Lopez, 2009)). Ce type de système requiert nécessairement que la tête de l'utilisateur soit stable, voire immobilisée, pour fonctionner correctement, ce qui interdit les mouvements naturels.

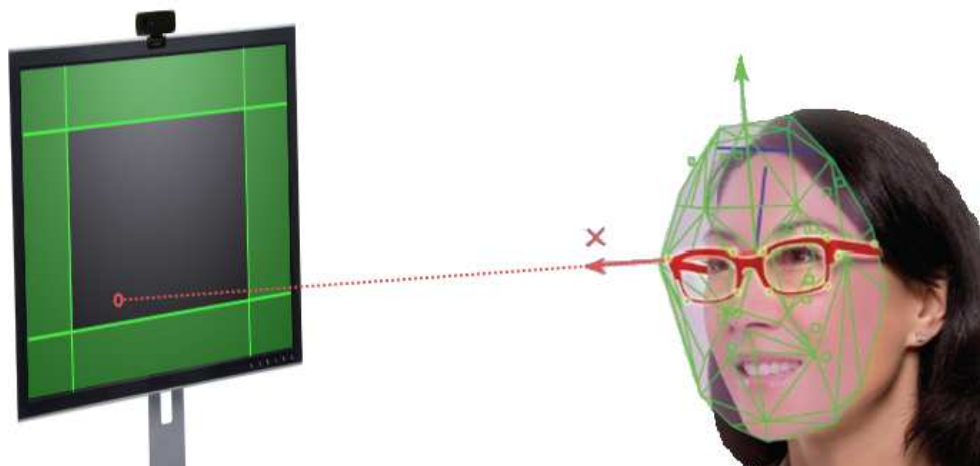


Figure 191 : Représentation de l'estimation du vecteur de regard par simple connaissance des caractéristiques de la tête de l'utilisateur

A fortiori, il est aussi possible, si l'on connaît la position et l'orientation des yeux par rapport au référentiel de la tête de l'utilisateur, de s'affranchir de ce pseudo-vecteur de regard qui passe par le nasion pour le remplacer par les vecteurs de regard passant respectivement par les centres des pupilles de chaque œil.

En ce qui concerne les méthodes de détection de la tête, nous n'allons pas rentrer dans les détails comme nous l'avons fait avec la détection de l'œil. Effectivement, il existe à l'heure actuelle des solutions sur étagère (Zhang & Gomes, 2014) permettant de s'affranchir de tels développements fastidieux (pour une revue des méthodes existantes, consulter (Al-Rahayfeh & Faezipour, 2013)). La problématique est de toute façon relativement similaire à celle rencontrée pour la détection de l'œil, à savoir l'identification d'une forme relativement simple mais avec une très grande variabilité qui dépend aussi bien du sujet observé que de l'environnement d'observation ou encore des capteurs utilisés (Mäkinen, 2007). On remarquera également que les implémentations prêtes à l'emploi, telles que celles que nous souhaitons utiliser, ont pour avantage de bénéficier de multiples optimisations assurant leur niveau de performances. Ainsi, nous avons décidé d'avoir recours au Intel® Perceptual Computing SDK (Intel, 2013) qui utilise la Creative Interactive Gesture Camera (**Figure 192**). Cette caméra dispose d'un capteur RGB haute définition et d'un capteur de profondeur, ainsi que de microphones mais nous n'utiliserons pas ces derniers. On signalera que ce type

de matériel peut également servir à la reconnaissance de gestes (pour un exemple d'utilisation avec un matériel similaire, consulter (Sanchis et al., 2014)).



Figure 192 : Photographie de la Creative Interactive Gesture Camera (Senz3D) (source (Creative Technology Ltd., 2015))

Le capteur de profondeur de cet appareil s'appuie sur le principe de « time-of-flight » (pour plus de précisions concernant cette technique, consulter (Büttgen, Oggier, Lehmann, Kaufmann, & Lustenberger, 2005)) afin d'évaluer la distance entre l'objet et la caméra. Cette technique est une méthode possible pour l'acquisition optique rapide de distances, qui, contrairement aux méthodes classiques, ne requiert ni de filtrages complexes ni de traitements de corrélation (pour une revue des différentes techniques d'évaluation de distance sans contact, consulter (Schwarte, Häusler, & Malz, 2000)). En effet, elle se limite à analyser une fonction d'auto-corrélation, aussi connue sous le nom de démodulation par échantillonnage (pour une brève explication des mécanismes de cette analyse, consulter (Büttgen et al., 2005; Ringbeck & Hagebecker, 2007)). D'autre part, ce type de caméras fournit une information de profondeur pour chaque pixel sans avoir à procéder à la moindre recherche complexe, souvent source d'erreurs. Ainsi, la puissance de calcul d'un système informatique utilisant de telles caméras peut être utilisée pour d'autres traitements que ceux visant à déterminer algorithmiquement la profondeur (Ringbeck & Hagebecker, 2007). Aussi, cette technique peut jouir d'une grande plage d'utilisation de quelques centimètres à plusieurs centaines de mètres (Büttgen et al., 2005). Néanmoins, nous pouvons lui associer les limitations suivantes :

- La précision par pixel à grande distance.
- La suppression avec succès de la lumière ambiante.
- L'adaptation dynamique aux variations de portée et des coefficients de réflexion.
- La résolution d'image, avec l'augmentation de laquelle le prix du capteur peut exploser.

Cette technique est ainsi fréquemment utilisée pour réaliser la détection d'objets, et elle est a fortiori aussi adaptée pour la détection de visage/tête humaine (Ringbeck & Hagebecker, 2007). A ce titre, elle trouve, par exemple, des utilisations dans le reconnaissance vocale multimodale (Abbas & Srivastava, 2012; Srivastava & Dev, 2013) ainsi que pour l'analyse assistée informatiquement de consultations médicales (Weibel et al., 2013). Ces exemples démontrent que son usage est adapté à des contextes similaires au nôtre.

Afin de procéder au suivi de la tête de l'utilisateur, le Intel® Perceptual Computing SDK (Intel, 2013) propose des algorithmes qui permettent, d'une part, de déterminer des points de repère faciaux (**Figure 193**), et, d'autre part, d'estimer l'orientation spatiale de la tête (Bradski, 1998)(**Figure 194**). En particulier, l'obtention de ces points de repère, qui s'appuie

sur l'utilisation des caractéristiques SURF (Li, Wang, & Zhang, 2011; Li & Zhang, 2013)(pour une présentation des méthodes les plus couramment utilisées pour le suivi de la tête, consulter (Zhang & Gomes, 2014)) nous permet d'estimer la position du nasion d'où partira le pseudo-vecteur de regard de l'utilisateur (**Figure 191**). Les coordonnées, dans le repère général, de ce point particulier pourront être notées (X_N^W, Y_N^W, Z_N^W) .

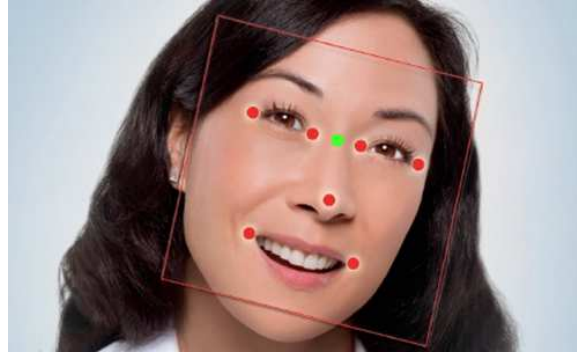


Figure 193 : Représentation de la position estimée du nasion (en vert) à partir de points de repère (en rouge) (à partir d'une source (Intel, 2015))

A l'aide de cette position et de l'orientation de la tête, les mouvements de la tête peuvent être représentés comme une combinaison de translations et de rotations (Allison et al., 1996). Telles que représentées sur la **Figure 194**, ces rotations sont définies à l'aide des angles d'Euler (Duchowski, 2007), qui sont utilisés, d'une manière générale, pour les transformations de corps rigides dans les environnements en 3D par rapport à une orientation initiale connue (LaValle, 2006). Ces angles d'Euler sont au nombre de trois, et peuvent être définis, dans notre cas, de la manière suivante :

- L'angle γ_{roll} de roulis autour de l'axe Z , aussi appelé « roll ».
- L'angle β_{pitch} d'élévation autour de l'axe Y , aussi appelé « pitch ».
- L'angle α_{yaw} de rotation azimutale autour de l'axe X , aussi appelé « yaw ».

L'orientation de la tête est donc définie comme étant un vecteur contenant les valeurs de ces trois angles. Le Intel® Perceptual Computing SDK (Intel, 2013) dispose d'une fonction qui peut retourner un tel vecteur. On pourra préciser matriciellement les trois rotations ainsi définies autour des trois axes du repère de la tête de l'observateur (**Figure 194**) :

$$R_X(\alpha_{yaw}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_{yaw} & -\sin \alpha_{yaw} \\ 0 & \sin \alpha_{yaw} & \cos \alpha_{yaw} \end{pmatrix}$$

$$R_Y(\beta_{pitch}) = \begin{pmatrix} \cos \beta_{pitch} & 0 & \sin \beta_{pitch} \\ 0 & 1 & 0 \\ -\sin \beta_{pitch} & 0 & \cos \beta_{pitch} \end{pmatrix}$$

$$R_Z(\gamma_{roll}) = \begin{pmatrix} \cos \gamma_{roll} & -\sin \gamma_{roll} & 0 \\ \sin \gamma_{roll} & \cos \gamma_{roll} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

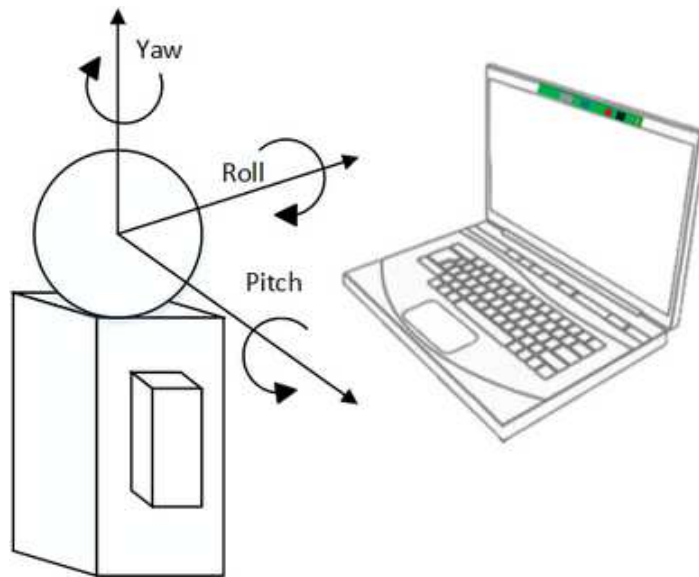


Figure 194 : Représentation des angles d'Euler utilisés pour le positionnement de la tête de l'utilisateur (source (Intel, 2015))

Sur la **Figure 195**, nous avons résumé le processus de traitements qui permet l'obtention de la position et de l'orientation de la tête à l'aide de la Creative Interactive Gesture Camera (Creative Technology Ltd., 2015). Ces valeurs permettent d'expliciter le changement de repère entre le système de coordonnées global, qui est le repère de la caméra et qui est lié à celui de l'écran, et celui de la tête de l'observateur (**Figure 184**). On pourra signaler que le processus proposé est, comme l'ensemble du processus de suivi oculaire, relativement modulaire ce qui offre la possibilité de le faire évoluer d'une manière assez souple. En particulier, pour plus de précisions, on pourra être amené à remplacer tout ou partie de ce traitement par un traitement utilisant des marqueurs IR directement placés sur les lunettes de suivi oculaire. De la même manière, il est également possible de carrément substituer à ce dispositif d'analyse vidéo des capteurs magnétiques (pour des exemples d'utilisation, consulter (Allison et al., 1996; Koons & Thorisson, 1993)) ou gyroscopiques (pour un exemple d'utilisation, consulter (O'Regan & Marnane, 2013)) directement montés sur la tête de l'utilisateur. Ces types de capteurs jouissent, en effet, d'un plus haut niveau de précision que celui actuellement obtenu à l'aide du matériel utilisé.

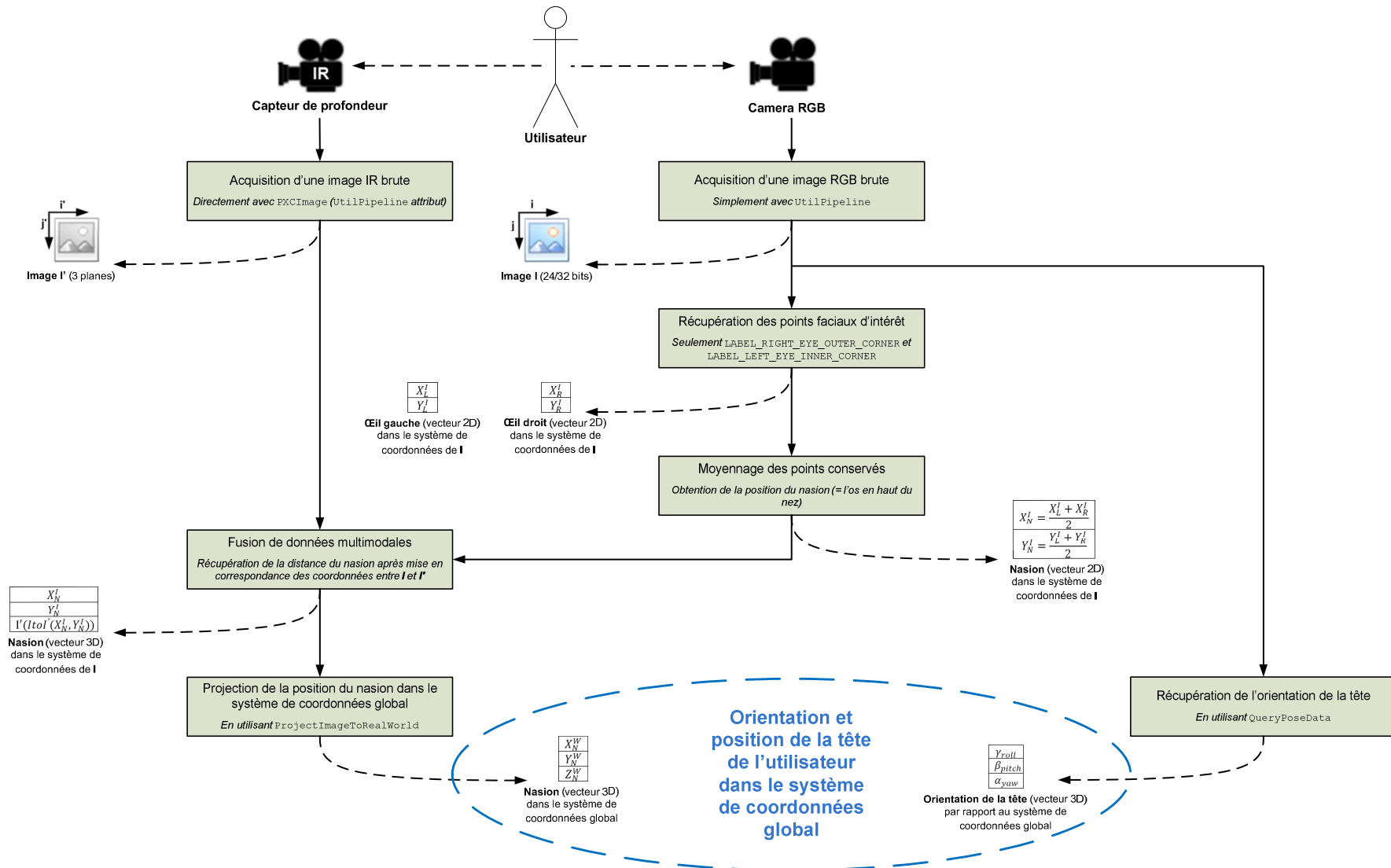


Figure 195 : Schéma détaillé de l'obtention de la position et de l'orientation de la tête

3. L'estimation du regard de l'utilisateur

Afin de déterminer le point de regard d'un observateur, il faut pouvoir établir les coordonnées de l'origine et de la direction de son regard par rapport à un repère dans lequel se trouve également l'objet de son observation. Dans notre cas, il va nous falloir, pour cela, combiner les données relatives à la position et l'orientation de sa tête par rapport au repère de l'écran et celles de la position et l'orientation de son œil par rapport à sa tête (**Figure 184**). Il est alors possible de déterminer l'origine et la direction de son regard par rapport à l'écran et de calculer le point de regard de l'observateur sur celui-ci (Allison et al., 1996). Pour rappel, nous disposons, à ce stade de nos traitements, des informations suivantes :

- La position et l'orientation de la pupille par rapport aux caméras placées sur la tête de l'utilisateur.
- La position et l'orientation de la tête de ce dernier par rapport à la caméra placée statiquement par rapport à l'écran.

Afin d'estimer la position et l'orientation de la pupille par rapport à l'écran, et ainsi établir le point de regard de l'observateur, il faut pouvoir accéder à certaines informations pour l'instant manquantes. En particulier, pour procéder aux différents changements de repères et projections requis, la géométrie de tous les éléments du système doit être connue. A ce titre, il va falloir réaliser plusieurs calibrages avant de déterminer la position du point de regard de l'utilisateur du système de suivi oculaire.

a. Le calibrage

Dans le cas de notre système, qui repose principalement sur la modélisation géométrique des éléments du dispositif, il faut procéder à trois types de calibrages (San Agustin Lopez, 2009) :

- La géométrie des éléments fixes du système, c'est-à-dire l'établissement de leurs positions relatives.
- Les paramètres des caméras à proximité de l'œil (Zhang, 2004), qui peuvent être intrinsèques ou extrinsèques.
- L'angle α , entre l'axe visuel et l'axe optique (Villanueva et al., 2007), qui peut être réalisé à l'aide d'un écran présentant un point.

On pourra remarquer que notre système est sensé être complètement calibré (Hansen & Ji, 2010), ce qui théoriquement lui confère une grande précision. En outre, ce système ne nécessite pas de réaliser le calibrage d'un modèle de correspondance entre l'apparence de l'œil et la position du regard, étant donné que le regard va être estimé à l'aide d'une reconstruction 3D d'une partie de l'œil (voir la sous-sous-section **Les techniques existantes**).

En ce qui concerne la géométrie de notre système, conformément à la **Figure 184**, il nous faut pouvoir établir :

- La relation qui lie le repère de la tête à ceux des caméras qui lui sont attachées.
- La relation qui lie le repère de l'écran à celui de la caméra distante.

La relation qui existe entre le repère de la tête et une des caméras destinées à filmer l'œil de près peut simplement être définie en coordonnées homogènes, à l'aide d'une rotation et d'une translation (Villanueva et al., 2006), ainsi :

$$H_{stereo} = \begin{pmatrix} R_{stereo} & T_{centre\ optique} \\ (0 & 0 & 0) & 1 \end{pmatrix}$$

où R_{stereo} et $T_{centre\ optique}$ sont à directement mesurer sur le dispositif après calibrage de la caméra. On peut considérer que, une fois les éléments du dispositif installés, les paramètres de cette transformation sont statiques. On pourra d'ailleurs les noter de cette manière :

$$R_{stereo} = (i_{stereo} \quad j_{stereo} \quad k_{stereo})$$

$$T_{centre\ optique} = (x_{stereo} \quad y_{stereo} \quad z_{stereo})^T$$

Il est ainsi possible d'exprimer un point ou vecteur de la scène filmée, noté \vec{x}_W dans le repère général, à l'aide de coordonnées exprimées dans le repère propre à la caméra à l'aide de la relation suivante (Ma, 1993) :

$$\vec{x} = R_{stereo}\vec{x}_W + T_{centre\ optique}$$

Chaque œil étant filmé par deux caméras il faut également mesurer la transformation permettant de passer de l'une à l'autre, ce qui est indispensable pour la reconstruction stéréoscopique (voir la sous-sous-sous-section **L'extraction des caractéristiques oculaires**).

Il faut également expliciter l'autre relation statique qui lie le repère de l'écran à celui de la caméra qui sert au suivi de la tête de l'observateur telle que représentée sur la **Figure 196**.

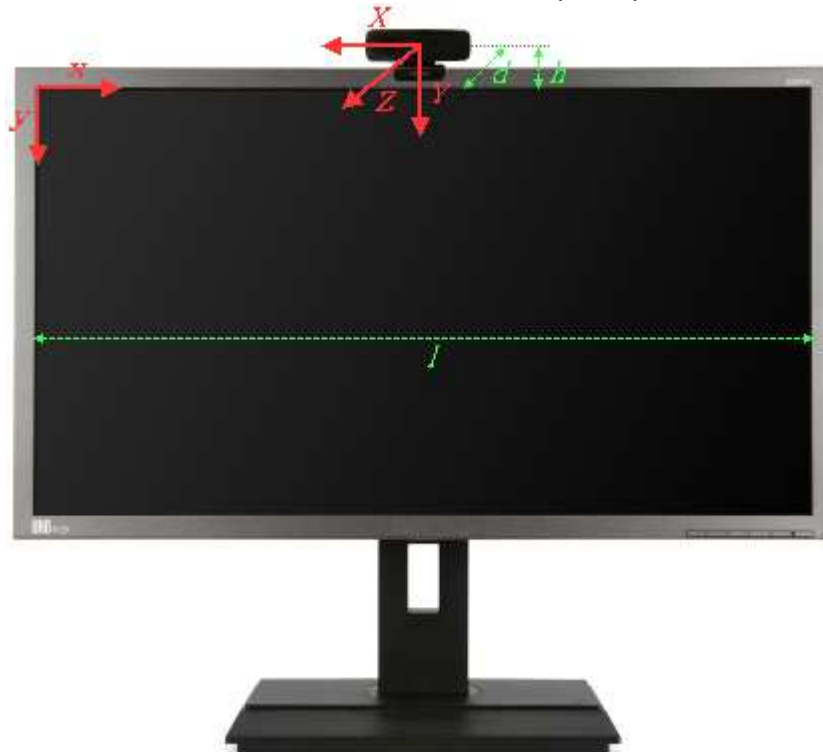


Figure 196 : Schématisation de l'installation de la caméra distante sur l'écran

Le changement d'origine de repère peut être effectué à l'aide du vecteur suivant :

$$T_{\text{écran}} = (x_{\text{écran}} \quad y_{\text{écran}} \quad z_{\text{écran}})^T = \left(\frac{l}{2} \quad h \quad d \right)^T$$

où l est la largeur de l'écran, h est la hauteur à laquelle la caméra est fixée au-dessus de l'écran et d est la distance entre le plan de l'écran et celui de la caméra (**Figure 196**).

Par rapport au repère général en 3D, les vecteurs de la nouvelle base 2D respectent les relations suivantes :

$$\begin{cases} x = -X \\ y = Y \end{cases}$$

Pour le passage à une unité de mesure en pixels, il faudra tenir compte de la résolution de l'écran qui est fréquemment exprimée en « pixels par pouce » et qui est spécifique au matériel utilisé : on la notera $res_{\text{écran}}$.

Le calibrage d'une caméra est, dans le domaine de la vision par ordinateur, une procédure visant à estimer la manière dont cette caméra projette un objet, qui appartient à l'environnement physique, sur le plan de l'image qu'elle enregistre (Zhang, 2004). Il est ainsi possible d'extraire, à partir d'images, des informations sur la métrique de l'environnement filmé (Isern González, Hernández Sosa, Domínguez Brito, & Naranjo Cabrera, 2005). Les techniques de calibrage peuvent reposer sur l'observation d'objets 3D, 2D, 1D ou encore sur un processus d'auto-calibrage (Zhang, 2004). Néanmoins, on notera que l'auto-calibrage est complexe et moins précis que les techniques qui utilisent un objet. Bien que les techniques les plus populaires soient celles qui reposent sur les objets 2D, les techniques s'appuyant sur des objets 3D semblent être les plus précises. Le modèle couramment adopté pour décrire le processus de formation d'image est le modèle du sténopé, également appelé modèle « pinhole » (**Figure 197**). Dans ce modèle, l'image m d'un point M de l'espace 3D est formée à l'intersection d'un rayon optique passant par M et par le centre optique o de la caméra, et du plan de l'image.

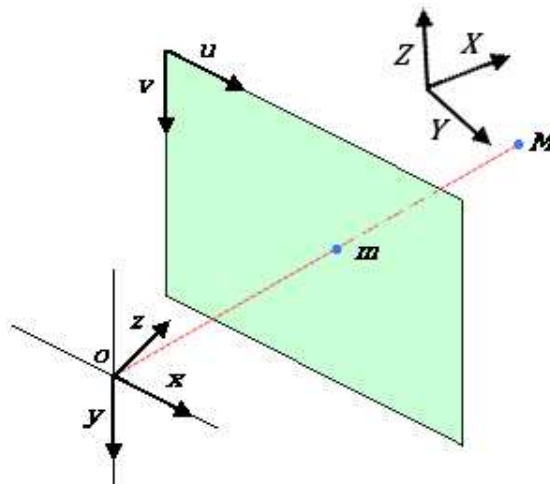


Figure 197 : Schéma des relations entre repères dans le modèle pinhole

En coordonnées homogènes, la relation entre M et m est la suivante (Zhang, 2004) :

$$km = AHM = PM$$

avec

$$A = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{m} = (u_m \quad v_m \quad 1)^T \quad \mathbf{M} = (X_M \quad Y_M \quad Z_M \quad 1)^T$$

$$H = (R \quad t) \quad P = AH \quad k \text{ est un facteur d'échelle arbitraire}$$

La matrice H contient les paramètres extrinsèques, c'est-à-dire la rotation R et la translation t qui permettent de passer du repère général de base vectorielle (X, Y, Z) au repère de la caméra de base vectorielle (x, y, z) . La matrice A contient les paramètres intrinsèques de la caméra, à savoir :

- (u_0, v_0) les coordonnées du point principale, c'est-à-dire la projection de \mathbf{o} sur le plan de l'image suivant le vecteur z .
- α et β les facteurs d'échelles des axes u et v de l'image.
- γ l'inclinaison de ces deux axes. Dans le cas de pixels rectangulaires, on a $\gamma = 0$.

La matrice P est appelée « matrice de projection de la caméra » et elle intègre l'ensemble des paramètres précités. Ainsi, il existe un lien fort entre paramètres intrinsèques et extrinsèques, ce qui rend difficile leurs calibrages conjoints (Isern González et al., 2005). Néanmoins, il est possible d'estimer P d'une manière linéaire, d'en extraire les paramètres puis d'affiner ces derniers jusqu'à l'obtention d'une solution acceptable. On remarquera qu'il peut également être nécessaire, en particulier avec des caméras bon marché, de tenir compte de la distorsion de la lentille. Nous ne détaillerons pas davantage ces procédures de calibrage de la caméra (pour plus de détails techniques, consulter (Zhang, 2004)).

Le dernier calibrage qui doit être réalisé est propre à l'anatomie de l'œil de l'utilisateur du système de suivi oculaire. Plus précisément, il s'agit de déterminer l'angle α entre son axe visuel et l'axe optique, que l'on peut géométriquement établir (Villanueva et al., 2007). Il sera alors ultérieurement possible, à partir du seul vecteur de l'axe optique de l'œil, de déterminer le vecteur du regard porté par l'axe visuel. Pour cela, il faut commencer par déterminer le centre de la courbure cornéenne, noté O_α , où s'intersecte l'axe optique et l'axe visuel (Chi et al., 2009; Nagamatsu et al., 2008; Nouredin et al., 2005)(Figure 198).

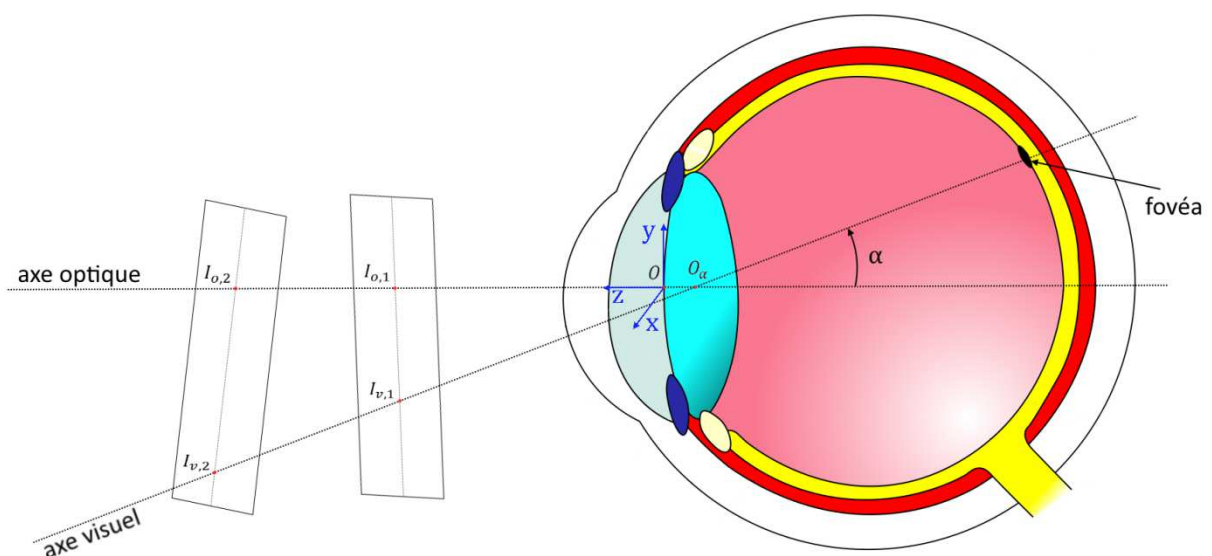


Figure 198 : Schéma de la détermination géométrique de l'angle α

Pour établir la position relative de O_α par rapport à O , ainsi que la direction réelle du vecteur de regard, il est possible d'utiliser seulement deux points de calibrage. Ainsi, l'utilisateur va être invité à regarder un premier point puis un deuxième dont la position par rapport à l'œil de l'utilisateur sera différente, soit en l'affichant sur l'écran à une position différente du premier soit en modifiant la position de l'observateur par rapport à l'écran entre les deux observations. Lors de chacune de ces observations, deux valeurs sont accessibles : l'intersection observée entre l'axe optique et l'écran, notée $I_{o,i}$, et l'intersection supposée de l'axe visuel et du plan de l'écran, notée $I_{v,i}$, qui est le point affiché (avec $i \in [1,2]$).

Dans le repère de l'œil, le problème peut alors être modélisé comme représenté sur la **Figure 198**. Les intersections entre l'axe optique et l'écran n'ont pas besoin d'être calculées : seule la normale à la pupille, dirigée par le vecteur unitaire z , nous intéresse car elle conditionne les coordonnées de l'ensemble des points manipulés. Cette dernière est obtenue implicitement à l'aide de la méthode d'extraction des caractéristiques oculaires que nous avons précédemment décrite (voir la sous-sous-sous-section **L'extraction des caractéristiques oculaires**). Les intersections entre l'axe visuel et le plan de l'écran vont, quant à elles, nous permettre de calculer le vecteur directeur du regard, noté \vec{V}_α , de la manière suivante :

$$\vec{V}_\alpha = (x_{\vec{V}_\alpha} \quad y_{\vec{V}_\alpha} \quad z_{\vec{V}_\alpha})^T = \begin{pmatrix} x_{v,2} - x_{v,1} \\ y_{v,2} - y_{v,1} \\ z_{v,2} - z_{v,1} \end{pmatrix}$$

avec (pour $i \in [1,2]$ et $x_{v,2} > x_{v,1}$)

$$I_{v,i} = (x_{v,i} \quad y_{v,i} \quad z_{v,i})^T$$

On peut déjà remarquer que, dans le repère de l'œil, le centre de la courbure cornéenne O_α exhibe certaines particularités. Effectivement, par définition :

$$O_\alpha = (x_{O_\alpha} \quad y_{O_\alpha} \quad z_{O_\alpha})^T = (0 \quad 0 \quad z_{O_\alpha})^T$$

En projetant l'axe visuel dans le plan défini par la base vectorielle (y, z) , il va être possible de calculer sa dernière coordonnée non nulle, z_{O_α} . En effet, l'axe visuel de vecteur directeur \vec{V}_α passe par O_α , donc la projection de cet axe passe également par la projection dans ce point. On peut alors poser le coefficient directeur de la projection de l'axe visuel :

$$a = \frac{y_{\vec{V}_\alpha}}{z_{\vec{V}_\alpha}}$$

Ainsi que son ordonnée à l'origine (pour $i \in [1,2]$) :

$$y_{v,i} = az_{v,i} + b \Leftrightarrow b = y_{v,i} - az_{v,i}$$

Etant donné que $y_{O_\alpha} = 0$, il est alors possible de calculer directement z_{O_α} :

$$y_{O_\alpha} = az_{O_\alpha} + b \Leftrightarrow z_{O_\alpha} = -\frac{b}{a} \Leftrightarrow z_{O_\alpha} = z_{v,i} - y_{v,i} \frac{z_{\vec{V}_\alpha}}{y_{\vec{V}_\alpha}}$$

Il est finalement possible de déterminer les coordonnées de O_α dans le repère de l'œil :

$$O_\alpha = \begin{pmatrix} 0 \\ 0 \\ z_{v,1} - y_{v,1} \frac{z_{v,2} - z_{v,1}}{y_{v,2} - y_{v,1}} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ z_{v,2} - y_{v,2} \frac{z_{v,2} - z_{v,1}}{y_{v,2} - y_{v,1}} \end{pmatrix}$$

Pendant l'enregistrement, il sera possible, via une translation $\overrightarrow{OO_\alpha}$ et en prenant comme nouveau vecteur directeur $\overrightarrow{V_\alpha}$, de passer de l'axe optique observé géométriquement à l'axe visuel estimé.

Pour finir, on signalera que les calibrages de la géométrie du système et des caméras peuvent être effectués une unique fois après la mise en place du dispositif. Le calibrage spécifique à un utilisateur devra, quant à lui, être effectué avant chaque session d'enregistrement impliquant ce dernier.

b. La détermination du point de regard

A l'aide de changements de repères (**Figure 184**), la direction du regard de l'utilisateur par rapport à l'écran va pouvoir être estimée. Il sera alors possible de déterminer les coordonnées de son point de regard sur l'écran. Pour cela, nous allons passer en revue les différentes transformations qui permettent d'exprimer le vecteur de regard dans le repère de l'écran, c'est-à-dire dans le repère général, puis finalement nous pourrions calculer l'intersection de ce vecteur avec le plan de l'écran.

En coordonnées homogènes, les transformations qui permettent de passer d'un repère à l'autre peuvent facilement être exprimées (Allison et al., 1996; Duchowski, 2007). Dans le cas du passage du repère de la tête à celui de l'œil, la transformation $H_{\text{œil}}$ peut être formulée ainsi :

$$H_{\text{œil}} = \begin{pmatrix} R_{\text{œil}} & T_{\text{pupille}} \\ (0 & 0 & 0) & 1 \end{pmatrix}$$

où $R_{\text{œil}}$ et T_{pupille} ont été calculées précédemment lors de l'extraction des caractéristiques de l'œil (voir la sous-sous-sous-section **L'extraction des caractéristiques oculaires**) et sont définis, à l'aide des notations que nous avons posées, de la sorte :

$$R_{\text{œil}} = (r_{11} \quad r_{12} \quad r_{13})$$

$$T_{\text{pupille}} = (x_0 \quad y_0 \quad z_0)^T = t_1$$

Dans le cas du passage du repère de la caméra, qui est liée à l'écran, à celui de la tête, la transformation $H_{\text{tête}}$, qui est mesurée à l'aide du Intel® Perceptual Computing SDK (Intel, 2013)(voir la sous-sous-section **La détection de la tête de l'utilisateur**), peut simplement être formulée de la sorte :

$$H_{\text{tête}} = \begin{pmatrix} R_{\alpha_{\text{yaw}}, \beta_{\text{pitch}}, \gamma_{\text{roll}}} & T_{\text{nasion}} \\ (0 & 0 & 0) & 1 \end{pmatrix}$$

où

$$R_{\alpha_{\text{yaw}}, \beta_{\text{pitch}}, \gamma_{\text{roll}}} = R_X(\alpha_{\text{yaw}})R_Y(\beta_{\text{pitch}})R_Z(\gamma_{\text{roll}})$$

$$= \begin{pmatrix} \cos \beta_{\text{pitch}} \cos \gamma_{\text{roll}} & -\cos \beta_{\text{pitch}} \sin \gamma_{\text{roll}} & \sin \beta_{\text{pitch}} \\ \cos \alpha_{\text{yaw}} \sin \gamma_{\text{roll}} + \cos \gamma_{\text{roll}} \sin \alpha_{\text{yaw}} \sin \beta_{\text{pitch}} & \cos \alpha_{\text{yaw}} \cos \gamma_{\text{roll}} - \sin \alpha_{\text{yaw}} \sin \beta_{\text{pitch}} \sin \gamma_{\text{roll}} & -\cos \beta_{\text{pitch}} \sin \alpha_{\text{yaw}} \\ \sin \alpha_{\text{yaw}} \sin \gamma_{\text{roll}} - \cos \alpha_{\text{yaw}} \cos \gamma_{\text{roll}} \sin \beta_{\text{pitch}} & \cos \gamma_{\text{roll}} \sin \alpha_{\text{yaw}} + \cos \alpha_{\text{yaw}} \sin \beta_{\text{pitch}} \sin \gamma_{\text{roll}} & \cos \alpha_{\text{yaw}} \cos \beta_{\text{pitch}} \end{pmatrix}$$

$$T_{\text{nasion}} = (X_N^W \quad Y_N^W \quad Z_N^W)^T$$

En se reposant sur la configuration géométrique des éléments du système (Villanueva et al., 2006), le vecteur de regard estimé \vec{V}_α' dans le repère général, peut alors directement être exprimé de la manière suivante :

$$\vec{V}_\alpha' = (x_{\vec{V}_\alpha'} \quad y_{\vec{V}_\alpha'} \quad z_{\vec{V}_\alpha'})^T = R_{\alpha_{yaw}, \beta_{pitch}, \gamma_{roll}} R_{stereo} \vec{V}_\alpha$$

Son origine $O_{\alpha'}$, dans ce même repère, peut être calculée ainsi :

$$O_{\alpha'} = T_{nasion} + R_{\alpha_{yaw}, \beta_{pitch}, \gamma_{roll}} (T_{centre\ optique} + R_{stereo} (T_{pupille} + R_{\alpha_{\text{œil}}} \overrightarrow{OO_{\alpha'}}))$$

Le point de regard dans le plan de l'écran, qui est lié à celui de la caméra qui détermine l'orientation et la position de la tête de l'observateur, peut alors être simplement calculé (Allison et al., 1996; Koons & Thorisson, 1993) :

$$\begin{cases} x = \frac{x_{\vec{V}_\alpha'}}{z_{\vec{V}_\alpha'}} (z_{\text{écran}} - z_{O_{\alpha'}}) + x_{O_{\alpha'}} \\ y = \frac{y_{\vec{V}_\alpha'}}{z_{\vec{V}_\alpha'}} (z_{\text{écran}} - z_{O_{\alpha'}}) + y_{O_{\alpha'}} \\ z = d \end{cases}$$

Ces coordonnées peuvent alors finalement être converties, en pixels, dans les coordonnées de l'écran :

$$\begin{cases} x' = res_{\text{écran}}(x_{\text{écran}} - x) \\ y' = res_{\text{écran}}(y - y_{\text{écran}}) \end{cases}$$

En résumé, l'estimation du point de regard de l'observateur va pouvoir algorithmiquement être estimé ainsi :

1. Calculer \vec{V} et O par stéréoscopie dans le repère d'une caméra proche de l'œil.
2. En déduire \vec{V}_α' et $O_{\alpha'}$ dans le repère général.
3. Calculer l'intersection (x, y) avec le plan de l'écran dans le repère général.
4. Convertir ces coordonnées en pixels, puis enregistrer le couple (x', y') obtenu.

On pourra finalement signaler que, si les deux yeux sont utilisés pour déterminer le point de regard, les positions déterminées à l'aide de chaque œil pourront être ensuite moyennées afin de n'obtenir qu'un unique point de regard (Nagamatsu et al., 2009). Cela sous réserve que ces positions ne soient pas trop distantes l'une de l'autre, car, effectivement, une divergence notable de ces positions pourrait être caractéristique d'une pathologie ou d'un problème de calibrage, ce qui nécessiterait alors de plus amples investigations.

c) Le traitement des données

Comme nous l'avions initialement indiqué, le traitement des données consiste ici à déterminer quels mouvements oculaires sont effectués par l'observateur. Cette identification des mouvements oculaires est une étape nécessaire pour pouvoir les analyser. En particulier, l'analyse de fixations et de saccades nécessite de pouvoir identifier les fixations, c'est-à-dire déterminer les positions, dans le plan de l'affichage, des fixations à

partir des données brutes échantillonnées suite à l'enregistrement de l'activité motrice de l'œil (Salvucci & Goldberg, 2000). A travers cette identification des fixations et saccades, il va être possible de réduire le volume de données manipulées. En effet, cette identification revient à agréger une très large partie des données relatives aux mouvements oculaires observés, ce qui, selon Salvucci et Goldberg (2000), est utile pour au moins deux raisons :

- Pendant les saccades, aucun traitement visuel complexe ne peut être effectué (Rayner, 1998), il est donc possible de s'affranchir des points échantillonnés pendant les saccades car ils ne sont révélateurs d'aucun traitement cognitif en particulier.
- Au cours des fixations, les petits mouvements tels les tremblements, micro-saccades et micro-dérives ne sont pas non plus révélateurs de traitements cognitifs de haut niveau (Irwin, 2004), ce qui rend possible le filtrage et le moyennage de points échantillonnés sans perdre les caractéristiques essentielles des mouvements identifiés (Špakov, 2012).

Nous allons ainsi mettre en place un certain nombre de filtres et d'agrégateurs afin de décomplexifier les données brutes du suivi oculaire et de procéder efficacement à l'identification des mouvements oculaires nous intéressant, à savoir les fixations et saccades. Une méthode simple d'identification des saccades peut être la mesure de la vitesse entre deux mesures transmise par le système de suivi oculaire (Stampe, 1993) : si la vitesse du déplacement angulaire excède, par exemple, les 30°/s, alors on peut estimer qu'il s'agit d'une saccade. On peut alors supposer que les valeurs entre des saccades appartiennent à des fixations. Or, on remarque que cette approche, bien que particulièrement simple, ne tient pas compte du troisième type de mouvement oculaire que nous souhaitons pouvoir identifier : la poursuite lisse, qui survient si le stimulus proposé n'est pas statique. Effectivement, le besoin d'identification de ce type de mouvements ne permet pas d'appliquer une méthode d'identification binaire par seuillage unidimensionnel. Il est ainsi difficile d'appliquer une méthode donnant de bons résultats pour détecter automatiquement les mouvements oculaires, et on peut constater qu'il existe une variabilité non négligeable entre les différents algorithmes qui existent (Cuba Gyllensten, 2014). Effectivement, il est délicat de trouver une méthode permettant de distinguer les saccades des poursuites lisses avec un haut niveau de fiabilité, par rapport à une identification réalisée par un expert du domaine ou attendue conformément aux stimuli proposés. De la même manière, les algorithmes actuellement disponibles ne semblent pas en mesure de distinguer les poursuites lisses de très faible amplitude angulaire, de l'ordre de 0.5-1°, des fixations.

En résumé, ce traitement de données va nous permettre de convertir le flux de données brutes des positions de l'œil en événements discrets. De la sorte, il va nous être possible de nous approcher d'une représentation des intentions et comportements de l'utilisateur, dans la mesure où nous allons pouvoir nous appuyer sur des modèles des mouvements oculaires pour filtrer les données directement recueillies (Jacob, 1991). Pour cela, ce traitement doit procéder (Cuba Gyllensten, 2014; Salvucci & Goldberg, 2000) :

- Au filtrage du bruit sur les enregistrements de l'évolution des positions du point de regard.
- A l'identification des mouvements oculaires à partir de ces enregistrements filtrés.

Nous allons ainsi détailler les algorithmes et méthodes impliqués dans ces opérations.

1. Le filtrage du bruit

Lors de ses expériences sur l'utilisation des données issues du suivi oculaire en tant qu'entrée d'une IHM, Jacob (1991) a observé ce qui semblait être un comportement erratique dans l'interface utilisateur, même lorsque l'utilisateur pensait qu'il fixait un élément de l'interface d'une façon parfaitement immobile. Cela était visiblement dû à des causes naturelles et artificielles : les mouvements normaux de micro-saccades ou de clignement de l'œil au cours de fixations, ainsi qu'un bruit gaussien de haute fréquence introduit par le matériel d'acquisition vidéo et/ou les algorithmes utilisés (Olsson, 2007). Afin de pallier à de possibles mises en défaut de notre système, il est primordial de filtrer les données avant de tenter de procéder à leur analyse en tant que telle (Stampe, 1993). Une méthode efficace, pour débarrasser le signal des imperfections de mesure, consiste à simplement lisser l'évolution des coordonnées brutes du point de regard (Špakov, 2012). Cette méthode peut évidemment n'être appliquée, si besoin, qu'à une des dimensions du système de coordonnées.

En 1991, Jacob a considéré que les écarts, entre des relevés de positions du point de regard distantes de 100 ms, ce qui est a fortiori le délai minimum pour détecter une fixation, de l'ordre de 0.5-1°, n'avaient pas à être considérés comme consécutifs à une saccade. Ainsi, durant une fixation entière, la position du point de fixation conservé est la moyenne des positions des points de regards obtenus consécutivement aux micro-saccades. Ce délai a d'ailleurs été utilisé avec succès dans des applications visant à analyser les comportements humains en condition de lecture (Campbell & Maglio, 2001). Effectivement, les mouvements oculaires de faible amplitude angulaire qui surviennent lors d'une fixation sont du même ordre de grandeur que le bruit de mesure et leur caractère involontaire les rend d'intérêt moindre. Dans le cas d'un matériel qui ne disposerait pas d'une haute fréquence d'échantillonnage ainsi que d'une bonne définition d'image, il est judicieux de filtrer ces mouvements liés aux fixations (Špakov, 2012). D'autre part, il est important de signaler qu'un filtre efficace doit dynamiquement s'adapter à l'état du signal qu'il filtre afin de respecter les caractéristiques locales de ce dernier (San Agustin Lopez, 2009). Dans notre contexte d'application, cela revient à modifier le comportement du filtre selon que le signal correspond à une fixation, une saccade ou une poursuite lisse. Assurément, il y aurait peu de sens à intégrer dans une même moyenne des valeurs appartenant, par exemple, à une fixation et d'autres provenant d'une saccade.

Parmi les techniques disponibles, une qui semble donner les meilleurs résultats est le filtre à réponse impulsionnelle finie à noyau gaussien (Jimenez et al., 2008). Cette méthode consiste à lisser le signal par l'intermédiaire du calcul d'une moyenne pondérée. Selon une seule dimension, chaque valeur p_x , correspondant au $x^{\text{ième}}$ échantillon parmi les N qui constituent l'enregistrement, est alors recalculée à l'aide de sa valeur mesurée et des $n - 1$ valeurs qui la précèdent. On pourra dire que nous appliquons un filtre sur une fenêtre de taille n . La nouvelle valeur, p'_x , est exactement obtenue de la sorte :

$$\forall x \in [1, N], \forall n \in [2, N], p'_x = \sum_{i=1}^{\min(x,n)} w_i p_{x-i+1}$$

avec la définition suivante des poids :

$$\forall i \in [1, n], w_i = \frac{f(i)}{\sum_{i=1}^n f(i)}$$

et à l'aide la fonction gaussienne suivante :

$$\forall i \in [1, n], f(i) = e^{-\frac{i^2}{2\sigma^2}}$$

Pour tenir compte du caractère bi ou tridimensionnel du point de regard, on pourra noter les nouvelles coordonnées spatiales, à l'aide d'une base vectorielle (X, Y, Z) de la manière suivante :

$$\forall x \in [1, N], x = (p'_{x,x}, p'_{y,x}, p'_{z,x})$$

En outre, afin de se prévenir des interférences causées par un clignement d'œil, c'est-à-dire une occlusion de l'œil par la paupière, il faut introduire dans l'algorithme d'identification des mouvements oculaires une tolérance à l'absence de données. En effet, dans le cas de l'impossibilité d'une détection de la pupille, ce dernier ne fournira aucune donnée relative à la direction du regard, ou alors il s'agira de données erronées. Il faut donc s'assurer que l'absence passagère de ces données ne génère pas d'artefacts parmi les mouvements oculaires détectés. Dans ce but, Jacob avait décidé que son système pourrait ignorer, sans plus de conséquences, l'absence de position du point de regard pendant un maximum de 200 ms (Jacob, 1991). Une telle suppression pure et simple des données correspondant à un clignement d'yeux est assurément la méthode la plus commune (Cuba Gyllensten, 2014). Elle peut, par exemple, être obtenue par un seuillage au niveau de la vitesse de déplacement du point de regard de façon à détecter simplement les valeurs aberrantes (Duchowski, 2007). On peut, par exemple, utiliser une vitesse angulaire de 100°/s (Larsson, Nyström, Andersson, & Stridh, 2015).

2. L'identification des mouvements oculaires

Il existe plusieurs types d'algorithmes d'identification des mouvements oculaires (Salvucci & Goldberg, 2000). Ils peuvent d'ailleurs être classés selon les caractéristiques qu'ils exploitent. A ce titre, on peut les répartir en différentes catégories qui utilisent :

- Les caractéristiques spatiales, c'est-à-dire que ces algorithmes peuvent être basés sur la vitesse, sur la dispersion ou sur les régions d'intérêt.
- Les caractéristiques temporelles, c'est-à-dire que ces algorithmes s'appuient sur des informations de durée des mouvements oculaires ou qu'ils analysent la proximité temporelle des échantillons.

On signalera qu'il est également possible d'identifier les mouvements oculaires à l'aide d'un modèle d'apprentissage (Vidal, Bulling, & Gellersen, 2012). Quelle que soit l'approche adoptée, ces algorithmes cherchent à identifier les mêmes mouvements oculaires. On peut d'ailleurs relever que les mouvements oculaires les plus fins, tels que les tremblements observés pendant les fixations, sont d'une amplitude tellement faible qu'il est délicat de les détecter. En effet, la marge d'erreur des appareils de mesure est souvent supérieure à cette amplitude (Martinez-Conde et al., 2004). Selon Jacob (1991), dans le cas des IHM, les mouvements oculaires autres que les saccades, fixations ou poursuites sont à ignorer car ils ne relèvent pas de la volonté de l'utilisateur. Ainsi, étant donné les limitations fonctionnelles

et techniques de notre contexte applicatif, notre dispositif ne pourra raisonnablement servir qu'à la détection des mouvements oculaires les plus évidents c'est-à-dire les fixations, les saccades et les poursuites. Naturellement, il est plus aisé de distinguer les différents mouvements oculaires si l'on connaît les caractéristiques des stimuli les provoquant ainsi que leurs moments d'occurrence (Komogortsev & Karpov, 2013). Cependant, dans le cas d'un système fonctionnant en temps réel dans un contexte d'IHM, il ne paraît pas réaliste d'avoir accès à de telles informations. On remarquera néanmoins qu'il existe des méthodes pour identifier les mouvements relativement subtils que nous préférons ignorer dans notre cas (comme, par exemple, les glissades (Nyström & Holmqvist, 2010)).

Dans le cas de l'algorithme que nous allons présenter, nous considérons que le filtrage du bruit, que nous avons précédemment décrit, est suffisant pour que les évaluations de critères, pour l'identification et la classification des mouvements oculaires, soient menées sur un signal supposé maintenant relativement exempt de bruit. En outre, nous comptons proposer un algorithme d'identification des mouvements basé sur l'évaluation par seuillage de critères simples, qui ne subissent aucune variabilité inter-individus notable. Ce type d'approche, qui est fréquemment basé en partie sur la vitesse oculaire, a d'ailleurs prouvé son efficacité (Komogortsev & Karpov, 2013). Afin de déterminer cette vitesse, qui peut être utilisée comme un indicateur facile à calculer et fiable (Nyström & Holmqvist, 2010), la distance entre les échantillons doit être considérée (Olsson, 2007). Cette distance peut également servir à évaluer la dispersion des échantillons, ce qui permet efficacement de distinguer les saccades des fixations (Salvucci & Goldberg, 2000). Si l'on considère que le point de regard reste toujours circonscrit dans un même plan, qui est muni d'une base vectorielle (X, Y) dans laquelle les coordonnées filtrées de ce point sont exprimées, alors cette distance peut être exprimée ainsi :

$$\forall x \in [2, N], l'(x) = \sqrt{(p'_{x,x} - p'_{x,x-1})^2 + (p'_{y,x} - p'_{y,x-1})^2}$$

On notera que, de la même manière, la distance peut également être calculée avec des coordonnées non filtrées : elle est alors notée $l(x)$. Le critère de dispersion prenant en compte les changements de distance, de direction et de colinéarité des vecteurs formés par les échantillons, il faut également évaluer la propension qu'ont trois échantillons consécutifs à être alignés :

$$\forall x \in [3, N], r(x) = \frac{\sqrt{(p'_{x,x} - p'_{x,x-2})^2 + (p'_{y,x} - p'_{y,x-2})^2}}{l'(x) + l'(x-1)}$$

Si les trois points, dans l'intervalle $[x-2, x]$ ($\forall x \in [3, N]$), tendent à s'aligner, c'est-à-dire qu'ils se dispersent dans une direction donnée, alors cette valeur tend vers 1. A contrario, si le troisième point repart dans la direction du premier alors ce critère tend vers 0. Le critère de dispersion peut alors être calculé ainsi :

$$\forall x \in [3, N], d(x) = l'(x) \times r(x)$$

Ce premier critère prend donc en compte la distance entre les échantillons ainsi que leur distribution spatiale. A partir de la distance entre les positions d'échantillons successifs, la vitesse, c'est-à-dire le second critère que nous évaluons, peut également être trivialement obtenue :

$$\forall x \in [2, N], v(x) = \frac{l'(x)}{t_x - t_{x-1}} \quad \text{où } t_x \text{ correspond à la valeur de l'échantillon } x \text{ dans le domaine temporel}$$

Chaque pic de cette fonction, c'est-à-dire chacune des augmentations soudaines des valeurs qu'elle prend, correspond alors à une saccade ou à une poursuite lisse. Les fixations sont quant à elles les plages de valeurs pour lesquelles la vitesse est quasiment nulle. Logiquement, il resterait maintenant à trouver le moyen de distinguer les poursuites lisses des saccades. En réalité, la vitesse des poursuites lisses peut être très faible et donc le problème serait, après atténuation du bruit, de distinguer les poursuites lisses des fixations, ce qui n'est en rien évident (Komogortsev & Karpov, 2013; Larsson et al., 2015; San Agustin Lopez, 2009). Il faut donc introduire des nouveaux critères afin de procéder à une telle discrimination. En particulier, il peut être intéressant de considérer l'accélération de déplacement du point de regard. Effectivement, dans le cas de la poursuite lisse, si le mouvement est uniforme alors l'accélération moyenne est supposée nulle en l'absence de bruit, comme elle l'est pour les fixations (Behrens, MacKeben, & Schröder-Preikschat, 2010). Cette accélération peut être simplement calculée ainsi :

$$\forall x \in [3, N], a(x) = \frac{v(x) - v(x-1)}{t_x - t_{x-1}}$$

Le **Tableau 32** présente les ordres de grandeur des valeurs prises par les trois critères, que nous avons retenus pour notre algorithme de classification des mouvements oculaires, pour chacun des types de mouvements que nous souhaitons pouvoir identifier.

Mouvements oculaires	Dispersion	Vélocités	Accélérations
Fixation	Approximativement nulle	Approximativement nulle	Approximativement nulle
Poursuite lisse	Variable	Légèrement positive	Approximativement nulle
Saccade	Importante	Fortement positive	Fortement positive puis fortement négative

Tableau 32 : Critères retenus pour la classification des mouvements oculaires

Nous allons utiliser trois fonctions d'état pour représenter les résultats d'évaluations des critères retenus pour la classification des mouvements oculaires (**Tableau 32**). La première de ces fonctions, $D(x)$, est relative à l'évaluation du critère dispersion et est définie ainsi :

$$\forall x \in [3, N], \begin{cases} D(x) = 0, \text{ si } d(x) < \varepsilon_D \\ D(x) = 1, \text{ sinon} \end{cases} \quad \text{où } \varepsilon_D \text{ est le seuil en dessous duquel la dispersion peut être considérée nulle}$$

La seconde fonction, $V(x)$, est relative à l'évaluation du critère de vitesse et est définie de la manière suivante :

$$\forall x \in [3, N], \begin{cases} V(x) = 0, \text{ si } v(x) < \varepsilon_V \\ V(x) = 1, \text{ sinon} \end{cases} \quad \text{où } \varepsilon_V \text{ est le seuil en dessous duquel la vitesse peut être considérée nulle}$$

Le dernière fonction, $A(x)$, est relative à l'évaluation du critère d'accélération et est définie de la sorte :

$$\forall x \in [3, N], \begin{cases} A(x) = 0, \text{ si } |a(x)| < \varepsilon_A \\ A(x) = 1, \text{ sinon} \end{cases} \quad \text{où } \varepsilon_A \text{ est le seuil en dessous duquel} \\ \text{l'accélération peut être considérée nulle}$$

On notera également que, par définition, ces fonctions d'état ne prendront des valeurs exploitables pour l'identification des mouvements oculaires qu'à partir du troisième échantillon. Le **Tableau 33** résume les valeurs que devraient prendre ces deux fonctions d'états pour les différents types de mouvements oculaires que nous souhaitons identifier. On pourrait remarquer que, théoriquement, seules $D(x)$ et $A(x)$ sont requises pour discriminer les mouvements oculaires que nous considérons. Néanmoins, le calcul de la vitesse étant requis pour obtenir l'accélération, la prise en compte de $V(x)$ ne représente pas réellement de coût supplémentaire en termes de calculs, et nous avons donc décidé de conserver ce critère, qui permet tout de même de faire une distinction entre les fixations et les saccades.

Mouvements oculaires	$D(x)$	$V(x)$	$A(x)$
Fixation	0	0	0
Poursuite lisse	1	0 ou 1	0
Saccade	1	1	1

Tableau 33 : Valeurs attendues des fonctions d'état en fonction des mouvements oculaires

Les plupart des algorithmes utilisant des caractéristiques spatiales reposent sur des mesures angulaires. Les mesures de déplacement angulaire, de vitesse angulaire et d'accélération angulaire sont donc fonction de leurs homologues au niveau du point de regard ainsi que de la distance entre l'observateur et la cible de son regard. Or, dans notre contexte applicatif, on peut considérer que, lors d'une session, la distance entre l'observateur et son observation est relativement fixe. A ce titre, il est possible de ne considérer que les caractéristiques spatiales des mouvements du point de regard pour un individu donné pendant une session donnée, comme nous le proposons au travers des critères que nous avons retenus (**Tableau 32**). Il faut néanmoins tenir compte du changement de distance, d'une session à l'autre, qui induit une dilatation ou une contraction de ces grandeurs proportionnellement à la variation de distance. Il reste donc à établir un moyen de déterminer des valeurs efficaces pour les trois seuils, ε_D , ε_V et ε_A , afin qu'ils tiennent compte de cette variabilité de l'ordre de grandeur des valeurs de ces critères. D'une manière générale, le niveau de bruit augmente avec cette distance. Nous souhaitons donc évaluer cette grandeur avant filtrage afin de l'utiliser pour déterminer dynamiquement les valeurs de ces seuils, qui deviennent alors les fonctions $\varepsilon_D(x)$, $\varepsilon_V(x)$ et $\varepsilon_A(x)$.

Pour rappel, n est la taille de la fenêtre du filtre employé. On signalera que, pour adapter dynamiquement la taille de ce filtre au mouvement en cours, il sera possible de faire varier la taille et les bornes de cette fenêtre, que nous noterons f . Ainsi, la fenêtre sera redéfinie de façon à débuter avec le dernier échantillon avant l'identification du mouvement en cours, noté o , pour finir à l'échantillon courant x avec un taille comprise entre 1 et n . Il sera ainsi

possible, en temps réel, de limiter l'influence d'échantillons considérés comme appartenant à des mouvements précédents. La fenêtre peut alors être représentée de la façon suivante :

$$\forall x, o \in [3, N] | x > o, f = [\max(o, x - n), x]$$

Pour déterminer le premier de ces seuils, $\varepsilon_D(x)$, il faut prendre en considération l'erreur moyenne de calcul de distance induite par le bruit que nous filtrons :

$$\forall x, o \in [2, N], \mu_L(x) = \frac{1}{x - \max(o, x - n)} \sum_{i=\max(o, x-n)}^x |l'(i) - l(i)|$$

Il est ainsi possible d'obtenir une estimation de la dispersion moyenne causée par le bruit :

$$\forall x \in [3, N], \mu_D(x) = \frac{1}{x - \max(o, x - n)} \sum_{i=\max(o, x-n)}^x \mu_L(x) \times r(i)$$

L'écart type observé peut alors être calculé :

$$\forall x \in [3, N], \sigma_D(x) = \sqrt{\frac{1}{x - \max(o, x - n)} \sum_{i=\max(o, x-n)}^x (\mu_L(x) \times r(i) - \mu_D(x))^2}$$

Finalement, le seuil peut être défini de manière à rendre le bruit négligeable dans l'évaluation du critère de dispersion :

$$\forall x \in [2, N], \varepsilon_D(x) = \lambda_D \times \sigma_D(x)$$

On remarquera que le paramètre λ_D permet de régler la sensibilité de seuil. Dans la littérature, une valeur entre 3 et 6 est fréquemment adoptée dans ce genre de situation.

D'une manière similaire, il est maintenant possible de déterminer une estimation de l'erreur moyenne de calcul de vitesse induite par le bruit :

$$\forall x \in [2, N], \mu_V(x) = \frac{1}{x - (\max(o, x - n) + 1)} \sum_{i=\max(o, x-n)+1}^x \frac{\mu_L(x)}{t_i - t_{i-1}}$$

L'écart type peut ainsi être calculé :

$$\forall x \in [2, N], \sigma_V(x) = \sqrt{\frac{1}{x - (\max(o, x - n) + 1)} \sum_{i=2}^x \left(\frac{\mu_L(x)}{t_i - t_{i-1}} - \mu_V(x) \right)^2}$$

Et le seuil peut être défini de manière à ignorer l'effet du bruit sur l'évaluation du critère de vitesse :

$$\forall x \in [2, N], \varepsilon_V(x) = \lambda_V \times \sigma_V(x)$$

Cette fois encore, la valeur du paramètre λ_D devrait être réglée entre 3 et 6. Ce réglage sera de moindre importance dans la mesure où ce seuil ne doit permettre de distinguer que les fixations des saccades entre lesquelles l'écart de vitesse est important.

Pour finir, le calcul de l'estimation de l'erreur moyenne de calcul de l'accélération induite par le bruit adopte le même schéma :

$$\forall x \in [3, N], \mu_A(x) = \frac{1}{x - (\max(o, x - n) + 2)} \sum_{i=\max(o, x-n)+2}^x \frac{\mu_V(x) - \mu_V(x-1)}{t_i - t_{i-1}}$$

L'écart type peut alors être calculé :

$\forall x \in [3, N]$,

$$\sigma_A(x) = \sqrt{\frac{1}{x - (\max(o, x - n) + 2)} \sum_{i=\max(o, x-n)+2}^x \left(\frac{\mu_V(x) - \mu_V(x-1)}{t_i - t_{i-1}} - \mu_A(x) \right)^2}$$

Le seuil peut être formulé de manière à ignorer l'effet du bruit sur l'évaluation du critère de d'accélération :

$$\forall x \in [3, N], \varepsilon_A(x) = \lambda_A \times \sigma_A(x)$$

Le paramètre λ_A pourra également être réglé pour prendre une valeur entre 3 et 6. D'une manière globale, seule l'expérimentation permettra d'établir les valeurs optimales des trois paramètres. On pourra également signaler que ce seuil, tel qu'actuellement défini, permet l'identification de poursuites lisses dans le cas du suivi d'un objet dont le mouvement est uniforme. A ce titre, il pourrait être judicieux de proposer un autre calcul de ce seuil afin qu'il permette également la discrimination des poursuites lisses à l'accélération non nulle.

Pour assurer la continuité de la détection des mouvements oculaires, nous allons utiliser des compteurs afin de comptabiliser le nombre d'échantillons à considérer dans notre fenêtre d'analyse. Ainsi, seront définis quatre compteurs cp_f , cp_s , cp_pl et cp_x , respectivement, alloués aux fixations, saccades, poursuites lisses et mouvements indéterminés. La valeur du compteur du mouvement couramment identifié doit être égale à $(x - o)$, alors que les valeurs des autres doivent être nulles. Par la même occasion, afin de ne pas avoir à stocker la valeur de o , la valeur d'un compteur cp pourra être utilisée dans les calculs des indicateurs basés sur des sommes pour lesquelles la fenêtre de calcul f devient alors $[x - \min(cp, n), x]$ (avec potentiellement une translation de 1 ou 2 unités de la borne inférieure selon le cas). En outre, pour minimiser l'erreur moyenne de détection, lors de l'analyse d'un échantillon x , de début et/ou de fin des mouvements oculaires imputable à la fréquence d'échantillonnage de l'appareil utilisé, la valeur temporelle t_x peut être remplacée par une valeur médiane entre cet échantillon et celui qui le précède (Andersson, Nyström, & Holmquist, 2010). Ainsi, le début et/ou la fin d'un mouvement oculaire, qui est détecté à t_x , sera enregistré comme se produisant en réalité à t , avec:

$$t = t_x - \frac{t_x - t_{x-1}}{2}$$

A l'aide de tous les éléments que nous avons définis, l'algorithme d'identification des mouvements oculaires peut être synthétisé. Il se présente sous la forme suivante :

1. On procède à l'initialisation :
 - a. Pour $x = 1$:
 - i. $p'_{X,x} = p_{X,x}$ et $p'_{Y,x} = p_{Y,x}$.
 - ii. Noter le **début de l'indétermination** à $t = 0$.
 - b. Pour $x = 2$:
 - i. Calculer $l(x)$.

- ii. Calculer la valeur filtrée $x = (p'_{x,x}, p'_{y,x})$ sur la fenêtre $f = [1,2]$.
 - iii. Calculer $l'(x)$.
 - iv. Calculer $v(x)$.
- c. Pour $x = 3$:
- i. $cp_f = 0$, $cp_s = 0$ et $cp_pl = 0$.
 - ii. $cp_x = 2$.
2. Pour l'échantillon x à partir du troisième ($x \in [3, N]$), on détermine les valeurs des fonctions d'état de la manière suivante :
- a. Calculer $l(x)$.
 - b. Calculer la valeur filtrée $x = (p'_{x,x}, p'_{y,x})$ sur la fenêtre $f = [x - \min(cp, n), x]$ avec $cp = \max(cp_f, cp_s, cp_pl, cp_x)$.
 - c. Calculer $l'(x)$ et $r(x)$.
 - d. Calculer $d(x)$, $\mu_D(x)$, $\sigma_D(x)$ et $\varepsilon_D(x)$.
 - e. Si $l'(x) < \varepsilon_D$ alors $D(x) = 0$ sinon $D(x) = 1$.
 - f. Calculer $v(x)$, $\mu_V(x)$, $\sigma_V(x)$ et $\varepsilon_V(x)$.
 - g. Si $v(x) < \varepsilon_V$ alors $V(x) = 0$ sinon $V(x) = 1$.
 - h. Calculer $a(x)$, $\mu_A(x)$, $\sigma_A(x)$ et $\varepsilon_A(x)$.
 - i. Si $|a(x)| < \varepsilon_A$ alors $A(x) = 0$ sinon $A(x) = 1$.
3. On évalue les valeurs prises par les fonctions d'état pour identifier les mouvements oculaires ainsi :
- a. Si $D(x) + V(x) + A(x) = 0$ alors une fixation est détectée :
 - i. $cp_f + +$.
 - ii. Si $cp_f = 1$ alors :
 - 1. Noter le **début de fixation** à $t = t_x - \frac{t_x - t_{x-1}}{2}$.
 - 2. Noter la **fin du mouvement précédent** à t pour lequel le compteur est non nul.
 - iii. $cp_s = 0$, $cp_pl = 0$ et $cp_x = 0$.
 - b. Sinon, si $D(x) + V(x) + A(x) = 1$ alors une saccade est détectée :
 - i. $cp_s + +$.
 - ii. Si $cp_s = 1$ alors :
 - 1. Noter le **début de saccade** à $t = t_x - \frac{t_x - t_{x-1}}{2}$.
 - 2. Noter la **fin du mouvement précédent** à t pour lequel le compteur est non nul.
 - iii. $cp_f = 0$, $cp_pl = 0$ et $cp_x = 0$.

- c. Sinon, si $D(x) = 1$ et $A(x) = 0$ alors une poursuite lisse est détectée :
 - i. $cp_{pl} + +$.
 - ii. Si $cp_{pl} = 1$ alors :
 - 1. Noter le **début de poursuite lisse** à $t = t_x - \frac{t_x - t_{x-1}}{2}$.
 - 2. Noter la **fin du mouvement précédent** à t pour lequel le compteur est non nul.
 - iii. $cp_f = 0$, $cp_s = 0$ et $cp_x = 0$.
 - d. Sinon, le mouvement est indéterminé :
 - i. $cp_x + +$.
 - ii. Si $cp_x = 1$ alors :
 - 1. Noter le **début de l'indétermination** à $t = t_x - \frac{t_x - t_{x-1}}{2}$.
 - 2. Noter la **fin du mouvement précédent** à t pour lequel le compteur est non nul.
 - iii. $cp_f = 0$, $cp_{pl} = 0$ et $cp_s = 0$.
4. $x + +$, puis retour à l'étape 2 si cette valeur existe.

Nous précisons que cet algorithme peut bénéficier de nombreuses optimisations assez simples qui ne sont pas détaillées ici. Par exemple, pour l'immense majorité des dispositifs, on remarque que $\forall x \in [2, N], t_x - t_{x-1}$ est une constante, ou encore il est possible de calculer incrémentalement les moyennes des critères considérés. Il est également important de conserver toutes les valeurs déjà évaluées pour les trois derniers échantillons afin de ne pas avoir à les recalculer. En outre, il sera aussi possible de procéder à un post traitement afin de tenter de désambigüiser les mouvements indéterminés. Effectivement, les mouvements indéterminés traduisent seulement l'impossibilité pour notre système soit de déterminer en temps réel la séparation nette entre deux mouvements oculaires différents, soit de conserver un niveau de précision suffisant pour identifier tous les échantillons d'un même mouvement oculaire comme lui appartenant. Pour cela, il est, par exemple, possible de prendre en compte les durées moyennes des mouvements oculaires que nous souhaitons identifier.

On notera néanmoins que le matériel que nous utilisons, c'est-à-dire des caméras bon marché, ne permettent pas nécessairement d'obtenir un nombre d'échantillons suffisants pour obtenir un taux élevé d'identification correcte des mouvements oculaires ainsi qu'une erreur temporelle de détection acceptable (Andersson et al., 2010). Effectivement, il faut un taux d'échantillonnage d'environ 200 Hz pour passer outre cet impact négatif du matériel sur la qualité de l'identification temporelle des mouvements oculaires. Avec notre approche, qui consiste à identifier un mouvement oculaire à partir d'un minimum de trois points, une fréquence d'échantillonnage minimum de 100 Hz est requise pour ne pas être amené à ignorer les saccades les plus rapides, c'est-à-dire qui sont de l'ordre de 20 ms. A contrario, si le taux d'échantillonnage augmente alors la quantité de calculs nécessaires augmente également et les performances de l'algorithme, pour une utilisation en temps réel, pourront d'autant en être amoindries.

d) La cartographie, l'analyse et l'interprétation des données

Bien que la cartographie, l'analyse et l'interprétation soient hors de notre périmètre de recherche, car elles relèvent de l'étude et l'explication des comportements oculaires, nous souhaitons évoquer les possibilités et limites des dispositifs de suivi oculaire dans des contextes applicatifs similaires au nôtre.

En premier, nous rappelons que le nombre et la précision des mouvements oculaires, ainsi que leur durée et leur durée de latence, peuvent être utilisés comme des métriques objectives reflétant certaines capacités neuropsychologiques (par exemple, utilisées pour l'étude de l'intégration entre perception auditive et visuelle (Xiao et al., 2007)). Ainsi, ces métriques peuvent aussi bien avoir un intérêt dans le domaine des études en psychologie (Clifton et al., 2007; Rayner, 1998) que dans celui de l'interaction homme-machine (Campbell & Maglio, 2001; Miyoshi & Murata, 2001b). Dans ces deux domaines, le but de la cartographie de ces données est de faire mettre en relation des stimuli à des mouvements oculaires, celui de l'analyse consiste ensuite à classifier ces relations comme étant différents types de comportements, et finalement celui de l'interprétation classe l'individu ou son intention au regard des comportements qu'il aura affichés.

L'analyse des données peut aller jusqu'à l'identification de comportements tels que la lecture (Campbell & Maglio, 2001). D'une manière générale, cette analyse va avoir pour but d'extraire les caractéristiques d'un comportement oculaire au regard de la tâche qui est proposée à l'observateur (Zeman, 2013). Cependant, la relative imprécision des systèmes de suivi oculaire est un problème pour établir une cartographie sur laquelle cette analyse va pouvoir rigoureusement s'appuyer (Hyrskykari, 2006). C'est principalement imputable aux performances du dispositif de mesure, à une dérive des paramètres de calibrage en cours de session et, finalement, aux limites même de la technique pour déterminer le point où se porte l'attention visuelle au regard des caractéristiques biologiques de l'œil humain (par exemple, dans le cas de la lecture, la perception visuelle est pour partie parafovéale (Clifton et al., 2007), ce qui ne se traduira pas nécessairement par des fixations). Dans le but de remédier à ces limitations, des algorithmes de corrections, s'appuyant sur les caractéristiques spatiales et temporelles des stimuli proposés, peuvent être mis en œuvre (par exemple, comme c'est le cas dans le logiciel *iDict* (Hyrskykari, 2006) qui tient compte du découpage en lignes du texte présenté pour corriger constamment l'erreur de mesure verticale du suivi oculaire). Effectivement, ce type d'approches permet de prendre en compte le conditionnement du comportement oculaire engendré par le stimulus présenté. Par exemple, lorsqu'on connaît le comportement d'un stimulus en mouvement, il est possible de déterminer si les mouvements oculaires d'une personne correspondent au suivi de ce stimulus (Vidal, 2014). Néanmoins, ce genre de correction n'est pas nécessairement souhaitable lorsqu'on compte analyser les comportements de sujets pathologiques chez lesquels, a fortiori, les mouvements oculaires peuvent être fortement divergents de ce à quoi on peut normalement s'attendre. Ces méthodes de correction seront donc plutôt à réserver au domaine des IHM pour augmenter la robustesse de tels dispositifs.

Il a été montré qu'un certain nombre de mesures de suivi de l'œil sont représentatives des processus cognitifs essentiels à la lecture (Rayner, 1998). Dans notre contexte applicatif, qui est celui du diagnostic et de la remédiation de la dyslexie, cela est primordial pour l'analyse et l'interprétation des données. En particulier, la durée des fixations oculaires semble

correspondre à la difficulté de traitement de l'information. Ainsi, cette durée augmente avec la quantité d'activité cognitive impliquée dans le processus d'interprétation (Irwin, 2004). En conditions normales de lecture, les fixations ne durent que 200-250 ms et la taille moyenne des saccades est de 7-9 lettres (Rayner, 1998). Mais ces performances sont influencées par les variables textuelles (sémantiques, orthographiques, etc.) et typographiques (qualité d'impression, longueur de ligne, espacement des lettres, etc.) ainsi que des caractéristiques propres au lecteur lui-même (Clifton et al., 2007). D'autre part, on peut noter que les mouvements oculaires diffèrent entre la lecture silencieuse et la lecture à haute voix. En effet, les durées moyennes de fixation sont plus longues lors de la lecture à haute voix, ou en écoutant une voix lisant le même texte, que dans la lecture silencieuse (Rayner, 1998). On signale également que, lorsque les compétences en lecture augmentent, la durée et le nombre de fixations diminuent, la longueur des saccades augmente et la fréquence des anti-saccades diminue. Des anti-saccades, c'est-à-dire des saccades dans le sens opposé à celui de la lecture normale, peuvent apparaître quand un mot long et/ou difficile est rencontré. Cela traduit le besoin pour le lecteur de revenir sur sa lecture, faute d'une compréhension directe. Chez les normaux lecteurs, ces anti-saccades représentent 10-15% des saccades totales effectuées (Rayner, 1998). Consécutivement à cette variabilité de la complexité, orthographique ou sémantique, au sein d'un même texte, la durée des fixations peut également osciller entre 100 et 500 ms et la taille des saccades entre 1 et 15 lettres. De telles métriques peuvent donc permettre d'établir, par rapport à un stimulus ou une série de stimuli donnés, différents profils de lecteurs allant, par exemple, des très bons lecteurs aux très mauvais lecteurs.

Afin d'étudier ces différents types de comportements observés pendant la lecture, divers algorithmes ont d'ailleurs été développés pour cartographier, analyser et interpréter les fixations et les saccades (par exemple consulter (Beymer & Russell, 2005; Biedert et al., 2009; Campbell & Maglio, 2001)). Généralement, ils consistent à identifier les fixations, et par complémentarité les saccades (étant donné qu'il n'y pas stimulus dynamique, il n'y a pas de poursuite lisse), puis à classifier les saccades en fonction de leur longueur et de leur direction. Les proportions de différents types de saccades, ainsi mises en évidence, permettent de déterminer le comportement de l'utilisateur vis-à-vis du texte et, par exemple, s'assurer qu'il le lit vraiment (Olsson, 2007). Il est également possible d'obtenir d'autres informations pertinentes telles que la vitesse de lecture ou le taux d'anti-saccades (Beymer & Russell, 2005).

Dans le cas de la dyslexie, ce type d'approche peut permettre de distinguer des enfants dyslexiques d'enfants normaux lecteurs. Selon Bosse et al. (2009), le nombre d'éléments visuels pouvant être traités en parallèle est, chez les dyslexiques, plus limité que chez le reste de la population. Chez ces personnes, il a d'ailleurs été observé, au cours de la lecture et de la recherche visuelle, beaucoup plus de fixations ainsi qu'un temps de fixation de plus longue durée que chez des sujets de contrôle (Bucci et al., 2012), ce qui est également cohérent avec une étude de Prado et al. (2007). A ce titre, dans leur outil destiné à dépister les dyslexiques, Al-Edaily et al. (2013) ont utilisé les mesures suivantes :

- La durée de chaque fixation pour chaque zone d'intérêt.
- La durée moyenne des fixations.
- Le nombre de fixations dans chaque zone d'intérêt.
- Le nombre total de fixations.
- Le nombre de rétro-saccades.

Ces indicateurs semblent leur avoir permis de classer avec succès des enfants dyslexiques et des normaux lecteurs. Nous pouvons donc être confiants quant aux possibilités offertes par les dispositifs de suivi oculaire : ces derniers devraient pouvoir s'intégrer efficacement à notre système de prise en charge de la dyslexie.

4) Vue d'ensemble du processus de suivi oculaire

En guise de conclusion concernant notre dispositif de suivi oculaire, nous allons présenter une vue d'ensemble du processus d'acquisition et de traitement de ces données. Nous allons également préciser quelques détails concernant les sortes d'agents impliquées dans ce processus de traitement d'image (**Tableau 34**). Pour cela, nous les décrirons et spécifierons leurs niveaux hiérarchiques ainsi que leurs entrées et sorties. On pourra d'ailleurs signaler que le paradigme MA a déjà été utilisé avec succès pour des applications de traitement d'images médicales (par exemple, consulter (Richard, Dojat, & Garbay, 2004)). L'utilisation d'agents offre la possibilité de modifier rapidement et seulement certaines parties du processus, ce qui en facilite l'évolution et/ou l'adaptation dynamique. La **Figure 199** résume les étapes du processus jusqu'à l'identification des mouvements oculaires. En l'état, ce processus est applicable à n'importe quelle application nécessitant une identification des mouvements oculaires et il n'est donc pas spécifique à la prise en charge de la dyslexie. Cette spécification pourrait néanmoins apparaître lorsque seront rajoutés des agents en charge de la cartographie, l'analyse et l'interprétation des données. Effectivement, ces agents seront spécialement conçus pour une tâche et l'identification des différents comportements qu'elle peut induire, en particulier, ceux observables spécifiquement chez les enfants dyslexiques. Finalement, il faudra aussi relever que le choix du matériel utilisé conditionne fortement le type d'application ainsi que les traitements algorithmiques sous-jacents (Hansen & Ji, 2010). A ce titre, une partie de la population d'agents devra être remplacée en cas de changement de type ou de nombre de capteurs, ou de leur géométrie.

Nom de l'agent	Niveau	Description	Entrée(s)	Sortie(s)
Agent Humain (AH)	0	Il interagit avec le matériel (AM).	- NA -	Comportement et expressions physiologiques
Agent Matériel (AM)	0	Il est constitué de caméras filmant l'AH selon différents points de vue.	Comportement et expressions physiologiques de l'AH	Données brutes
Agent Logiciel de Traitement (ALT)	0	Il assure le suivi oculaire en laissant la liberté des mouvements de la tête.	Données brutes	Mouvement oculaire en cours
Agent Logiciel de Suivi Oculaire (ALSO)	1	Il assure la fonction de suivi oculaire à proximité de l'œil.	Données formatées (4 RGB)	Position et orientation de l'œil par rapport à la tête
Agent Logiciel de Suivi de la Tête	1	Il réalise la fonction de suivi de la tête.	Données formatées	Position et orientation de la

(ALST)			(1 RGB + 1 IR)	tête par rapport à l'écran
Agent Logiciel de Projection du Regard (ALPR)	1	Il projette le regard sur le plan de l'écran (Allison et al., 1996).	Sortie d' ALSO + sortie d' ALST	Position du point de regard
Agent Logiciel d'Identification des Mouvements oculaires (ALIM)	1	Il détermine les mouvements oculaires de l'utilisateur.	Sortie d' ALPR	Sortie d' ALT
Agent Logiciel de Formatage d'Images (ALFI)	2	Il formate les données brutes.	Données brutes	Données formatées pour les traitements
Agent Logiciel d'Identification Oculaire (ALIO)	2	Il détecte la pupille de l'œil par un seuillage multi niveaux (Liao et al., 2001) suivi d'une segmentation par level set (Shi & Karl, 2008).	Sortie RGB d' ALFI	Contour de la pupille oculaire
Agent Logiciel des Caractéristiques Oculaires 3D (ALCO)	2	Il procède stéréoscopiquement à l'extraction des axes optiques de chaque œil en 3D (Kohlbecher et al., 2008).	Sortie d' ALIO	Position de référence de l'œil + direction du regard
Agent Logiciel de Projection de l'Œil (ALPO)	2	Il calcule le changement de repère entre les repères de l'œil et de la tête de l'utilisateur.	Sortie d' ALCO	Sortie d' ALSO
Agent Logiciel des points d'Intérêt de la Tête (ALIT)	2	Il évalue les positions des points d'intérêt du visage de l'utilisateur et calcule la position du nasion (Li et al., 2011; Li & Zhang, 2013).	Sortie RGB d' ALFI	Positions des points d'intérêt du visage en particulier du nasion
Agent Logiciel d'Orientation 3D de la Tête (ALOT)	2	Il estime l'orientation spatiale du visage de l'utilisateur par rapport à la caméra (Bradski, 1998).	Sortie RGB d' ALFI	Orientation spatiale de la tête (= angles d'Euler)
Agent Logiciel de Position 3D du Nasion (ALPN)	2	Il fusionne les données des analyses 2D et celles de profondeur en corrigeant la parallaxe.	Sortie IR d' ALFI + sortie d' ALIT	Position de référence de la tête de l'utilisateur
Agent Logiciel de Projection de la Tête (ALPT)	2	Il calcule le changement de repère entre les repères de la	Sortie d' ALPN + sortie d' ALOT	Sortie d' ALST

	tête de l'utilisateur et du dispositif (AM).		
--	--	--	--

Tableau 34 : Description de tous les agents impliqués dans le suivi oculaire

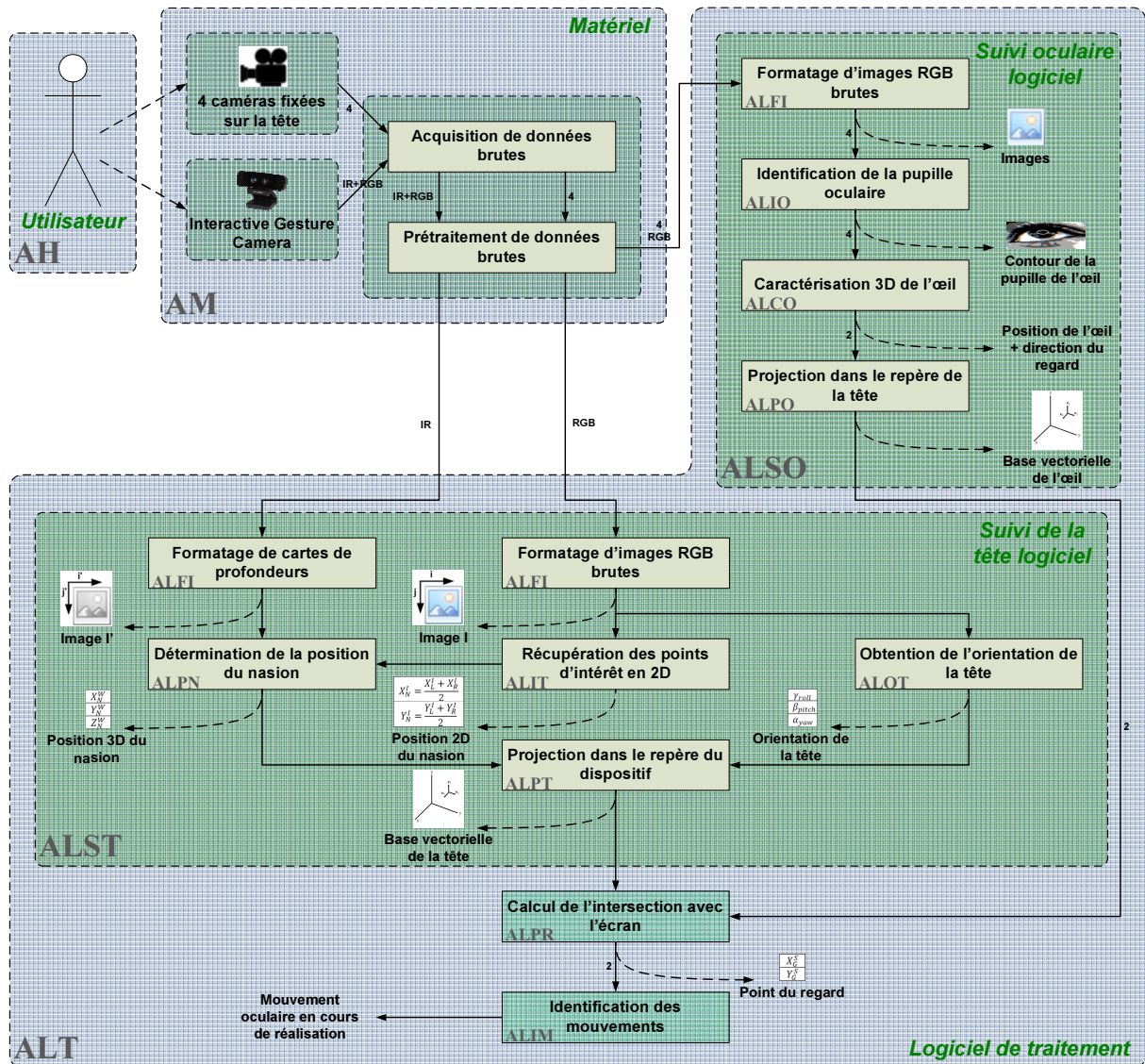


Figure 199 : Schéma de l'ensemble des traitements multi-agents permettant le suivi oculaire

B) L'électroencéphalographie

L'analyse de signaux EEG consiste à étudier la dynamique de l'activité cérébrale afin d'apprécier l'exécution de processus cognitifs (Luck, 2005). En effet, il est admis que les manifestations d'une activité électrophysiologique cérébrale, distribuée spatialement et temporellement, sont révélatrices et caractéristiques de processus cognitifs spécifiques. De la sorte, dans une situation donnée, on peut supposer que si l'on peut enregistrer ces manifestations de l'activité cérébrale chez un individu, alors il est possible d'apprécier le type et la qualité des traitements cognitifs que ce dernier met en œuvre. Une méthode d'imagerie médicale rapide et peu invasive pour observer la dynamique de l'activité cérébrale est de disposer des électrodes sur le scalp de la personne afin d'enregistrer les signaux, en provenance de l'encéphale, qui traversent la boîte crânienne (Teplan, 2002). Actuellement, ces techniques, relevant de l'EEG, sont couramment utilisées en milieu clinique et permettent de déceler rapidement certains dysfonctionnements neurologiques. Notre but ici n'est pas de corrélérer des observations avec des déficits cognitifs mais de simplement discuter d'une manière, pour un dispositif d'enregistrement EEG, de procéder à de telles observations.

1) Le principe de fonctionnement

Le fonctionnement cérébral est particulièrement complexe et loin d'être entièrement expliqué par la science actuelle. Afin de préciser les enjeux fonctionnels et techniques de l'imagerie fonctionnelle cérébrale, nous présenterons sommairement l'activité électrique observable par EEG, puis nous évoquerons les techniques modernes employées pour son enregistrement et son traitement.

a) L'activité électrophysiologique

L'EEG permet d'observer l'activité électrique cohérente d'assemblées de neurones du cerveau (Luck, 2005). On suppose que lorsque différentes assemblées de neurones communiquent entre elles, elles le font de manière synchronisée. D'ailleurs, on identifie différents rythmes cérébraux présumément liés aux communications impliquées dans les activités cognitives. On remarquera qu'un rythme cérébral désigne une oscillation électromagnétique, composée d'ondes, dans une bande de fréquences donnée. Les activités électriques cérébrales rythmiques sont ainsi classées en fonction de leur fréquence (Pammer, 2014; Teplan, 2002) :

- Les ondes delta (1-4 Hz)
- Les ondes thêta (4-8 Hz)
- Les ondes alpha (8-12 Hz)
- Les ondes beta (13-24 Hz)
- Les ondes gamma (25-50 Hz)
- Les hautes ondes gamma (> 50 Hz)

La dynamique des fréquences corticales n'a pour l'instant que très peu été utilisée dans le cas de l'étude de la dyslexie (Pammer, 2014). Néanmoins, les enregistrements EEG peuvent

être utilisés par la technique de **Potentiels Evoqués** (Luck, 2005). Un PE désigne la modification du potentiel électrique produite par le système nerveux en réponse à une stimulation interne ou externe et enregistrée grâce à des techniques comme l'EEG ou la MEG (Teplan, 2002). Ces PE sont caractérisés graphiquement, sur un tracé EEG, par leurs composantes identifiables par la présence de pics ou de creux (**Figure 200**). Ils sont adaptés à l'étude de l'aspect des processus cognitifs de nature normale et pathologique, et ils permettent la mise en évidence de l'évolution des activations cérébrales impliquées dans les processus mentaux. En particulier, leur utilité a été montrée, par exemple, pour juger de l'efficacité sur le fonctionnement cérébrale d'une thérapie ciblant certains processus déficitaires chez des enfants dyslexiques (Santos, Joly-Pottuz, Moreno, Habib, & Besson, 2007). Avant de discuter plus en détail des aspects techniques de la méthode EEG, on note que, couramment, sont distingués quatre types de PE :

- Les PE visuels, qui peuvent être déclenchés, par exemple, par une stimulation par flashes lumineux. Ces PE permettent l'étude des voies visuelles de la rétine jusqu'aux neurones occipitaux.
- Les PE auditifs, qui sont consécutifs à une stimulation sonore. Ils permettent l'étude des fibres nerveuses des voies auditives partant des oreilles jusqu'aux projections cérébrales qui leur sont associées.
- Les PE somesthésiques, qui sont obtenus par brèves stimulations électriques des nerfs périphériques des membres. Ces PE servent à étudier les voies sensibles des membres jusqu'à l'encéphale.
- Les PE moteurs, qui sont produits à l'aide d'une stimulation par TMS des zones motrices cérébrales. Ces PE permettent l'étude des voies motrices entre les régions corticales cérébrales et les membres.

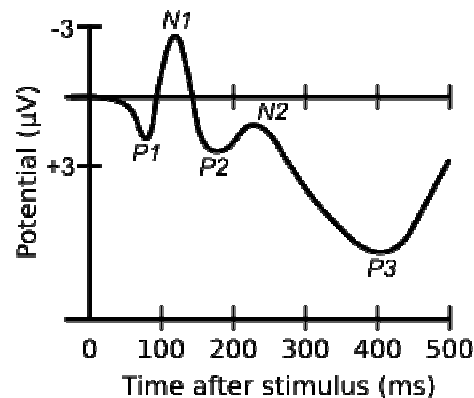


Figure 200 : Exemple de tracé mettant en évidence plusieurs composantes des PE (source (Wikipedia, 2015))

b) Les fondamentaux de la technique

Tout d'abord, il faut préciser que l'EEG a une moins bonne résolution spatiale que d'autres techniques d'imagerie fonctionnelle, telles que l'IRM fonctionnelle ou la MEG, mais a une meilleure résolution temporelle que celles-ci (Luck, 2005; Vion-Dury, 2010). Ainsi, ces différentes techniques d'imagerie sont fréquemment utilisées de manière complémentaire,

de manière multimodale. Généralement, un système de mesure EEG est constitué des éléments suivants (Teplan, 2002) :

- Des électrodes. Il existe différents types d'électrodes, à usage unique ou réutilisables, qui disposent d'une partie métallique et qui peuvent nécessiter ou non l'ajout d'un liquide ou gel conducteur. Les électrodes peuvent être maintenues à l'aide d'un bonnet, d'un casque ou d'une bande.
- Des amplificateurs de signal disposant de filtres. Le gain de tels amplificateurs est généralement compris entre 100 et 100000.
- Un convertisseur analogique-numérique, dans le cas d'un enregistrement numérique. La fréquence d'échantillonnage d'un tel appareil est habituellement comprise entre 128 Hz et 1024 Hz.
- Un enregistreur. Les données numériques peuvent rapidement occuper un espace mémoire important, en particulier si de nombreux canaux sont enregistrés et si la précision de mesure est importante. A ce titre, la capacité de stockage de l'enregistreur utilisé doit être adaptée aux différents paramètres du contexte de l'enregistrement.

L'activité électrique des assemblées de neurones est mesurée au travers d'électrodes placées sur le scalp du patient. Les signaux captés par ces électrodes sont alors massivement amplifiés avant d'être enregistrés sur papier ou sur un composant électronique de stockage de données. Il existe différentes manières de calculer les potentiels électriques au niveau de chaque électrode, selon le choix de l'électrode de référence retenue pour chacune d'entre elles. Nous ne détaillerons pas les différents montages possibles.

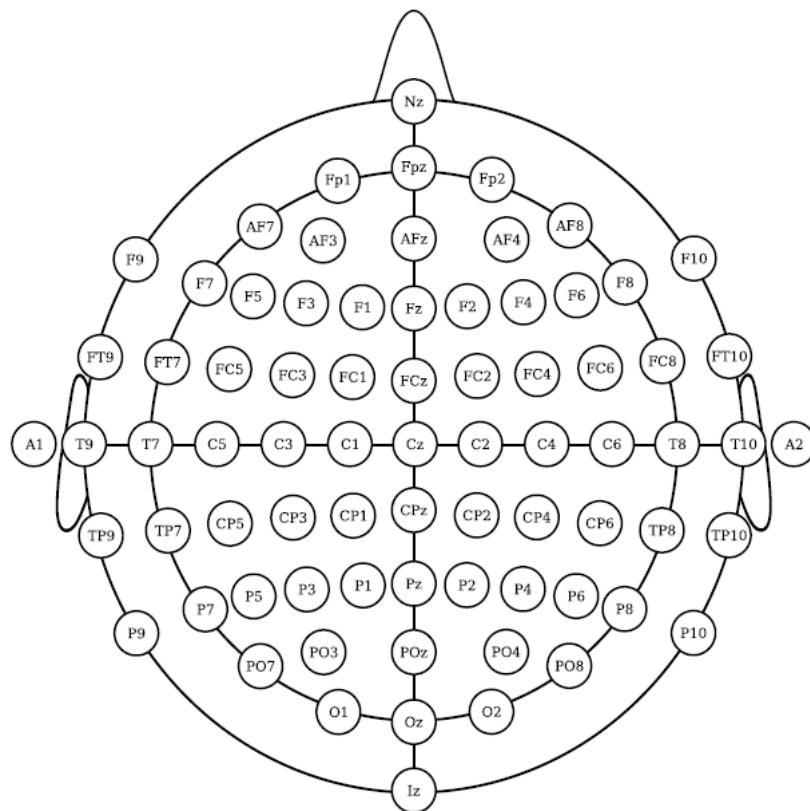


Figure 201 : Emplacement et nomenclature des électrodes intermédiaires à 10%, tel que normalisé par l'American Clinical Neurophysiology Society (American Clinical Neurophysiology Society, 2015)

Afin de normaliser cette acquisition de mesures, la disposition de ces électrodes respecte généralement le standard utilisé en contexte clinique : le système international 10-20 de positionnement des électrodes en EEG (Klem, Lüders, Jasper, & Elger, 1999). L'American Clinical Neurophysiology Society a également proposé une série d'améliorations pour ce standard (American Clinical Neurophysiology Society, 2006) donnant naissance au système 10-10 (**Figure 201**). Ce type de nommage renvoie aux valeurs des distances entre électrodes adjacentes qui représentent 10% ou 20% des distances totales entre l'avant et l'arrière du crâne, plus précisément entre le nasion et l'inion, ou entre ses côtés droit et gauche, plus précisément entre les points pré-auriculaires droit et gauche. Ainsi, dans le système 10-10, le nombre d'électrodes ainsi que la définition de l'image cérébrale acquise sont supérieurs à ceux du système 10-20. Il existe même un système standard d'encore plus haute définition appelé système 10-5 ou système à 5% (Oostenveld & Praamstra, 2001). Néanmoins, plus la densité des électrodes est élevée, plus il est probable que ces électrodes mesurent un potentiel identique. On signalera qu'il est également délicat de respecter exactement le placement, sur le crâne, des électrodes de façon à ce qu'il soit comparable/valide entre différents enregistrements (Jurcak, Tsuzuki, & Dan, 2007). Effectivement, la reproductivité des mesures EEG reste ambiguë et réduit la précision de ce type de système. Les trois standards 10-20, 10-10 et 10-5 définissent, respectivement, le placement de 21, 74 et jusqu'à 345 électrodes. Ces électrodes mesurent des signaux électriques normalement d'une amplitude de 0.5 μ V à 100 μ V. Les différences de potentiels sont mesurées entre les électrodes et une électrode de référence.

Il est notable que les appareils de recueil de signaux EEG sont particulièrement sensibles et qu'ils peuvent facilement être parasités, en particulier, par les clignements d'yeux ou les mouvements de la tête du patient qui génèrent des artefacts (O'Regan & Marnane, 2013). De tels mouvements peuvent également provoquer un déplacement ou une perte de contact, même temporaire, d'une ou plusieurs électrodes, ce qui se traduit également par une altération conséquente de la qualité des signaux enregistrés. Les perturbations électriques environnementales sont également sources de bruit et, dans un contexte de recherche, il peut être nécessaire d'utiliser une cage de Faraday. L'identification de bruit ou d'artefacts sur des enregistrements EEG est fréquemment attribuée à un expert du domaine. Pour finir, on signalera également que l'intérêt des PE est de procéder au calcul d'un signal moyen, ce qui permet d'en retirer toute la composante aléatoire correspondant au bruit enregistré sur le signal brut (Luck, 2005).

2) Les utilisations possibles

Naturellement, les dispositifs d'EEG sont conçus et utilisés pour l'exploration cérébrale (Vion-Dury, 2010). Ils permettent à la fois l'étude de l'activité cérébrale spontanée et celle conditionnée par stimulation. L'EEG est ainsi utilisé pour, par exemple, monitorer l'état global de fonctionnement du cerveau, détecter des lésions locales, étudier la dynamique des processus mentaux ou encore tester l'effet de médicaments psychoactifs (Teplan, 2002). Néanmoins, on signalera que, quelles que soient la définition et la précision des mesures EEG effectuées, elles ne pourront jamais donner qu'un aperçu de certains aspects du fonctionnement cérébral (Vion-Dury, 2010). L'utilisation de cette technique à des fins médicales devra donc toujours être soumise au regard critique d'un praticien. En particulier,

il faut noter que les caractéristiques des PE présentent des différences entre les individus et même entre différentes séries de mesures chez un seul individu (Luck, 2005). A ce titre, il est délicat d'établir avec certitude ce que devrait être un PE normal.

A l'heure actuelle, les dispositifs d'EEG ne sont plus exclusivement utilisés à des fins médicales/paramédicales mais aussi pour évaluer, par exemple, des stratégies commerciales ou pour être utilisés comme dispositifs d'IHM. Ainsi, on constate l'apparition de plus en plus de solutions de BCI basées sur l'exploitation de signaux EEG (par exemple, consulter (Renard et al., 2010)).

Dès à présent, nous pouvons proposer, à l'aide du méta-modèle que nous avons retenu pour représenter les modalités d'IHM (Figure 15)(Obrenović & Starčević, 2004, 2006), une modélisation UML générique décrivant les éléments pertinents de la modalité EEG (Figure 202). Nous signalons que nous avons ajouté quelques stéréotypes au méta-modèle originel afin qu'il puisse supporter les implications sous-tendues par les BCI.

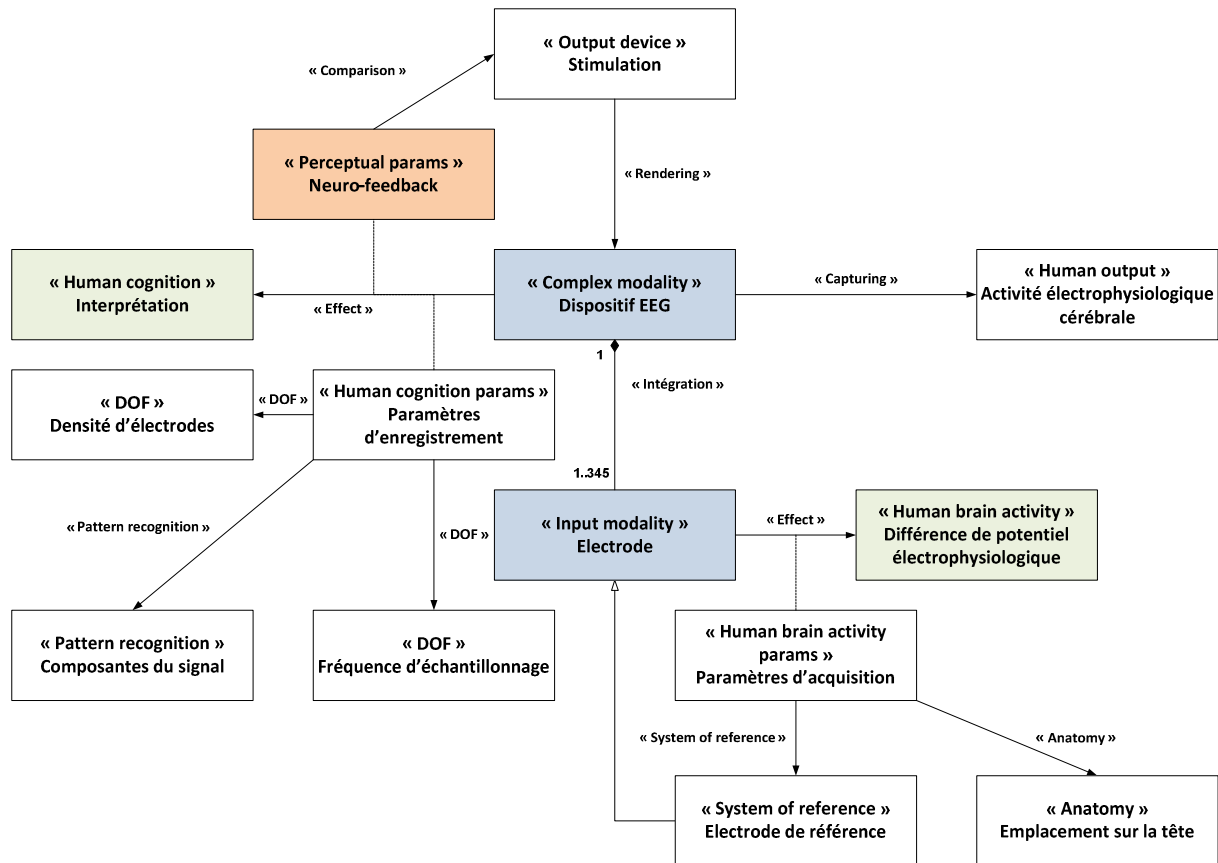


Figure 202 : Diagramme de classe de la description de la modalité EEG

Dans cette section, nous allons, dans un premier temps, nous restreindre à présenter l'utilisation qui est faite de l'EEG pour le calcul de PE dans le cadre des recherches sur les fondements neurologiques de la dyslexie. Ensuite, nous présenterons très brièvement l'usage qui en est fait en tant que dispositif d'IHM.

a) Un appareil d'analyse de l'activité cognitive des dyslexiques

Dès 1998, Karmiloff-Smith a fait valoir que les études longitudinales sur l'imagerie cérébrale devraient être utilisées pour les troubles du développement. En effet, même si à un niveau comportemental on observe des performances dans la norme dans un domaine donné, ces performances peuvent être obtenues grâce à des processus cognitifs différents (Karmiloff-Smith, 1998). L'imagerie cérébrale doit donc nous permettre d'observer des manifestations physiologiques spécifiques des différents processus cognitifs alternatifs impliqués dans ces performances. Ainsi, afin d'utiliser la mesure de PE et l'identification de leurs caractéristiques comme outil d'aide diagnostique, les mêmes événements électrophysiologiques majeurs peuvent faire l'objet d'observation, lors de tâches linguistiques mais pas uniquement, chez des sujets dyslexiques et chez des sujets témoins avant d'être finalement comparés (Karapetsas & Zygouris, 2011; Shaul, 2008). Les études menées relèvent fréquemment d'importantes différences de latences et d'amplitudes entre les PE de ces deux groupes de sujets (par exemple, consulter (Martin, 2005)). Il est alors possible de comparer statistiquement les principales composantes des PE entre ces groupes afin d'établir des indicateurs de la pathologie. Des études ont même tenté d'utiliser les PE pour déterminer à quel sous-groupe de dyslexiques appartiennent les enfants ainsi diagnostiqués (par exemple, consulter (Lachmann, Berti, Kujala, & Schröger, 2005)). Une telle comparaison peut également intervenir entre différentes mesures effectuées sur un même sujet, par exemple, avant et après remédiation afin de juger de l'efficacité de cette dernière au travers des modifications observées au niveau de certains de ces potentiels (par exemple, consulter (Jucla, Nenert, Chaix, & Demonet, 2010; Santos et al., 2007)).

En particulier, une composante fréquemment utilisée en tant qu'indicateur, dans le cas de la dyslexie, est la « négativité de discordance » ou **MisMatch Negativity**. Cette MMN est une onde cérébrale qui traduit un changement du stimulus, par exemple, lorsqu'un nouveau stimulus apparaît dans une séquence de stimuli identiques. On trouve, par exemple, une MMN atténuée en réponse à des stimuli vocaux chez des enfants dyslexiques (Schulte-Körne, Deimel, Bartling, & Remschmidt, 1998). La MMN, utilisée comme moyen de dépistage, peut d'ailleurs prédire robustement les futures capacités de lecture (Maurer et al., 2009). D'une manière générale, le cerveau des dyslexiques semble exhiber des différences neurophysiologiques observables à l'aide des PE tels que la MMN, la P300 ou encore la N400 (Habib, 2000).

Finalement, nous signalons que, selon Goswami et al. (2010), les méthodes d'EEG ne sont ni rapides ni faciles à utiliser sur des enfants. Cependant, comme nous l'avons vu, l'utilisation de cette technique permettrait d'améliorer le diagnostic de la dyslexie, ainsi que potentiellement d'autres maladies neuropsychologiques développementales. Ainsi, avec notre système, nous voulons pallier à ce défaut d'usage afin de rendre de telles méthodes plus rapides et simples à mettre en place avec des enfants.

b) Un dispositif d'IHM

L'EEG peut être utilisé comme interface neuronale directe afin de transmettre des commandes à un système par l'interprétation de l'activité cérébrale de son utilisateur (Turk

& Kölsch, 2003). Les BCI sont des interfaces qui identifient des commandes de l'utilisateur à partir, par exemple, de caractéristiques de ses ondes cérébrales ou de certains PE (par exemple, consulter (Ramírez-Cortés, Alarcon-Aquino, Rosas-Cholula, Gomez-Gil, & Escamilla-Ambrosio, 2010)). D'après Turk (1998), les interactions homme-machine basées sur les interfaces graphiques, en particulier ayant une dépendance au clavier et à la souris, ne peuvent satisfaire les besoins des futurs types d'interactions. Les BCI s'inscrivent ainsi dans le prolongement des PUI (voir la section **Les interfaces utilisateur**) afin de supporter ces nouveaux besoins en termes d'IHM. A ce titre, on pourra, par exemple, signaler l'existence de la plate-forme gratuite et open-source nommée OpenViBE (Renard et al., 2010), qui permet la conception, l'évaluation et l'exécution de solutions de BCI et dont la dernière version a été rendue disponible très récemment (Inria, 2015). Cet outil a, entre autres, l'avantage de proposer un langage de conception graphique afin d'être utilisable par des utilisateurs sans compétence particulière en informatique.

Bien que nous ne rentrions pas dans les détails de ce domaine d'application, il est à noter que ce genre de technologie peut s'avérer précieux pour les personnes souffrant de handicap, en particulier s'il s'agit d'un handicap moteur. En effet, la mise en œuvre de BCI permet de s'affranchir de mouvements qui, généralement, nécessitent une motricité fine. En outre, ces interfaces peuvent servir à fournir un neuro-feedback dans le cadre du suivi de programme de remédiation (pour un exemple d'application, consulter (Shindo et al., 2011)). Néanmoins, on remarque qu'en tant que dispositif de sélection, il a été constaté, par exemple, que l'utilisation des interfaces neuronales était moins flexible et donnait moins de satisfaction à ses utilisateurs (Pasqualotto, Federici, & Olivetti Belardinelli, 2012) que celles basées sur le suivi oculaire (Mele & Federici, 2012a). Il semble donc judicieux d'utiliser ces différentes technologies de manière complémentaire, de manière multimodale, afin de proposer une interface plus flexible et plus facilement appréhendable. Ce constat nous conforte dans notre approche multimodale et nous apprend que l'acquisition de données EEG ne devrait, si possible, pas être utilisée à des fins sélectives mais plutôt pour procéder à un recueil passif de l'état électrophysiologique de l'utilisateur.

3) La conception de notre dispositif d'analyse de données EEG

Contrairement à notre étude du suivi oculaire, nous ne rentrons pas ici dans tous les détails des traitements algorithmiques mis en œuvre. Ainsi, nous allons nous restreindre à présenter globalement le processus dans sa globalité et nous ne discuterons pas des alternatives possibles à chacune des tâches le constituant.



Figure 203 : Photographie de l'Emotiv Epoc® (source (Emotiv Inc., 2015))

Tout d'abord, nous avons choisi d'utiliser un casque EEG bon marché et « plug-and-play », ce qui le rend facile d'accès et d'utilisation. Notre choix s'est ainsi porté sur l'Emotiv Epoc® (**Figure 203**) avec lequel nous avons développé un prototype d'acquisition et de traitement de données EEG (voir la section **L'objectif PROTO2**). Effectivement, ce matériel remplit les contraintes de coût et d'utilisabilité que nous venons d'énoncer et il dispose également d'un SDK en permettant l'intégration à notre système (Emotiv Inc., 2013). Bien que la qualité des mesures effectuées par ce matériel ne soit pas aussi bonne qu'avec des appareils à usage clinique, celle-ci reste suffisante pour calculer des PE exploitables (pour des exemples d'utilisation, consulter (Badcock et al., 2013; Ramírez-Cortes et al., 2010)). En outre, il a également prouvé son intérêt et son efficacité dans le domaine des IHM (par exemple, consulter (Gonzalez-Sanchez et al., 2011)). Nous allons maintenant expliciter la manière dont nous avons procédé pour calculer des PE à l'aide de ce matériel, puis nous évoquerons comment ces derniers peuvent être utilisés par un agent d'aide à la décision.

Pour débiter nos traitements, à l'aide de ce matériel, nous avons recueilli M enregistrements R_i (où $i \in [1, M]$). Chaque enregistrement R_i est réalisé dans l'intervalle $[t_{i,1}, t_{i,X_i}]$ et comprend X_i échantillonnages auxquels on procède sur N canaux simultanément. Nous définissons matriciellement ces enregistrements ainsi :

$$\forall i \in [1, M], R_i = \begin{pmatrix} x_{i,1,t_{i,1}} & \cdots & x_{i,1,t_{i,X_i}} \\ \vdots & \ddots & \vdots \\ x_{i,N,t_{i,1}} & \cdots & x_{i,N,t_{i,X_i}} \end{pmatrix}$$

Dans cette matrice, chaque ligne correspond ainsi à un canal, qui est associé à une électrode, et chaque colonne correspond à un échantillonnage. Il peut y avoir des chevauchements de ces enregistrements, de telle sorte que les intersections entre les différents intervalles d'échantillonnage ne sont pas toutes nécessairement vides. On fixe cependant l'ordre des bornes inférieures de ces intervalles tel que :

$$\forall i \in [2, M - 1], 0 \leq t_{1,1} \leq t_{i,1} \leq t_{i+1,1}$$

Pour bien faire, il faut ensuite effectuer un prétraitement du signal afin d'éliminer le bruit causé par l'alimentation électrique de l'appareil de mesure. Pour cela, un filtre est appliqué afin de retirer la composante principale à 50/60 Hz du signal (Teplan, 2002). Les artefacts de mesure peuvent également faire l'objet d'une suppression manuelle ou automatique (par exemple, par détection des mouvements de la tête du patient (O'Regan & Marnane, 2013)).

Notre but étant de calculer des PE, nous devons pouvoir moyenner ces différents enregistrements EEG. A ce titre, il faut que nos signaux disposent au moins d'une fréquence d'échantillonnage comparable, ou à défaut d'un peigne d'échantillonnage comparable. Dans ce deuxième cas, nous parlerons de « pseudo-fréquence » d'échantillonnage. Pour déterminer cette pseudo-fréquence, on propose donc de redéfinir l'ensemble des valeurs temporelles normalisées des occurrences d'échantillonnage effectuées pour l'ensemble des enregistrements. Nous nommons cet ensemble Ω_E . Pour cela, nous allons normaliser nos échantillons de façon à ce que leur nouvel intervalle d'échantillonnage soit le suivant :

$$[T_1, T_E] = \bigcup_{i=1}^M [0, t_{i,X_i} - t_{i,1}]$$

Le nombre d'échantillons pour chaque enregistrement va alors devenir $Card(\Omega_E) = X_E$, et respecter la contrainte suivante :

$$\forall i \in [1, M], \max_{i \in [1, M]} (X_i) \leq X_E \leq \sum_{i=1}^M X_i$$

On peut même expliciter l'ensemble Ω_E des échantillons de la manière suivante :

$$\Omega_E = \{t_{i,j} - t_{i,1} \mid (i \in [1, M]) \wedge (j \in [1, X_i])\}$$

Naturellement, on notera que le nombre d'échantillons constituant chaque enregistrement est voué à augmenter. Pour chaque enregistrement, afin de déterminer les valeurs EEG correspondant à ces nouvelles valeurs d'échantillonnage, nous procédons de la sorte :

$\forall i \in [1, M], \forall t \in [T_1, T_E]$:

- Si cette valeur d'échantillonnage t se trouve dans la plage de l'enregistrement R_i , c'est-à-dire si $t \in [0, t_{i,X_i} - t_{i,1}]$, alors on teste si une valeur a été mesurée à l'instant t , c'est-à-dire si $\exists j \in [1, X_i] \mid t = t_{i,j} - t_{i,1}$. Si l'enregistrement R_i ne dispose pas d'une telle valeur alors on l'interpole, par exemple, à l'aide de splines cubiques.
- Sinon, si la valeur d'échantillonnage est hors de la plage de l'enregistrement, alors on fixe la valeur de l'enregistrement à ω , ce qui signifie « pas de valeur ».

Cette méthode nous permet d'obtenir des enregistrements normalisés, disposant chacun de X_E valeurs par canaux, tels que :

$$\forall i \in [1, M], R_i^* = \begin{pmatrix} x_{i,1,T_1} & \cdots & x_{i,1,T_E} \\ \vdots & \ddots & \vdots \\ x_{i,N,T_1} & \cdots & x_{i,N,T_E} \end{pmatrix}$$

Il reste ensuite à moyenner ces différents enregistrements afin de calculer les PE. Pour ce calcul, on considère que lorsqu'un échantillon n'a pas de valeur alors sa valeur est considérée comme nulle dans le cadre du calcul de cette moyenne, c'est-à-dire $\omega = 0$. Ainsi, la matrice Pe , qui doit accueillir le calcul des PE et qui est dimensionnellement comparable à celle des enregistrements précédents, peut être calculée de la sorte :

$$\forall k \in [1, N], \forall j \in [1, E], Pe = \begin{pmatrix} \cdots \\ \vdots & \frac{\sum_{i=1}^M x_{i,k,T_j}}{M_j^*} & \vdots \\ \cdots \end{pmatrix}$$

Nous pouvons d'ailleurs préciser comment est déterminé M_j^* , qui représente le nombre d'échantillons, indifféremment mesurés ou interpolés, réellement porteurs d'une valeur :

$$\forall k \in [1, N], \forall j \in [1, E], M_j^* = M - \sum_{i=1}^M (x_{i,k,T_j} = \omega)$$

On signalera également que M_j^* est indépendant du canal considéré.

Les PE ainsi calculés peuvent maintenant être utilisés pour fournir une aide à la décision. Effectivement, les caractéristiques de ces potentiels étant caractéristiques de processus cognitifs, il paraît logique de les utiliser pour discriminer les personnes chez lesquelles ces processus sont déficitaires. Pour cela, il est possible d'utiliser la technique de distorsion temporelle dynamique qui consiste à comparer les PE d'un individu spécifique à un gabarit représentant les PE moyens constatés chez des individus sains (Asseondi et al., 2009; Casarotto, Cerutti, Bianchi, & Chiarenza, 2003). Ainsi, on peut automatiquement aligner les ondes cérébrales composant les enregistrements des PE ainsi que quantifier les caractéristiques morphologiques de ces PE. Les divergences entre la référence et les PE de l'individu permettent alors de proposer une aide à la décision.

4) Vue d'ensemble du processus de traitement des données EEG

Finalement, comme nous l'avons fait avec le suivi oculaire, nous souhaitons donner une vue d'ensemble du processus de traitement des données EEG. Néanmoins, pour ne pas alourdir cette synthèse, nous ne préciserons pas les agents éventuellement impliqués et nous nous contenterons de proposer une représentation graphique de ce processus de traitement qui est linéaire. Ainsi, la **Figure 204** résume ce que devrait être un processus d'analyse de signaux EEG menant à une aide diagnostique avancée. Cette figure intègre à la fois ce que nous avons implémenté mais aussi une partie relative à de la classification de signaux afin de proposer une aide à la décision allant jusqu'à l'identification potentielle d'une pathologie donnée (Asseondi et al., 2009; Casarotto et al., 2003). En effet, en disposant d'une base de signaux EEG de référence appartenant à des individus sains et à d'autres souffrant de pathologies spécifiques, nous souhaitons pouvoir, à terme, indiquer si un individu testé souffre probablement d'une pathologie et, si c'est le cas, pouvoir suggérer laquelle. Naturellement, on remarque qu'un tel procédé peut être appliqué à un large spectre de pathologies, ce qui étend les perspectives du dispositif bien au-delà de la seule sphère de la prise en charge de la dyslexie.

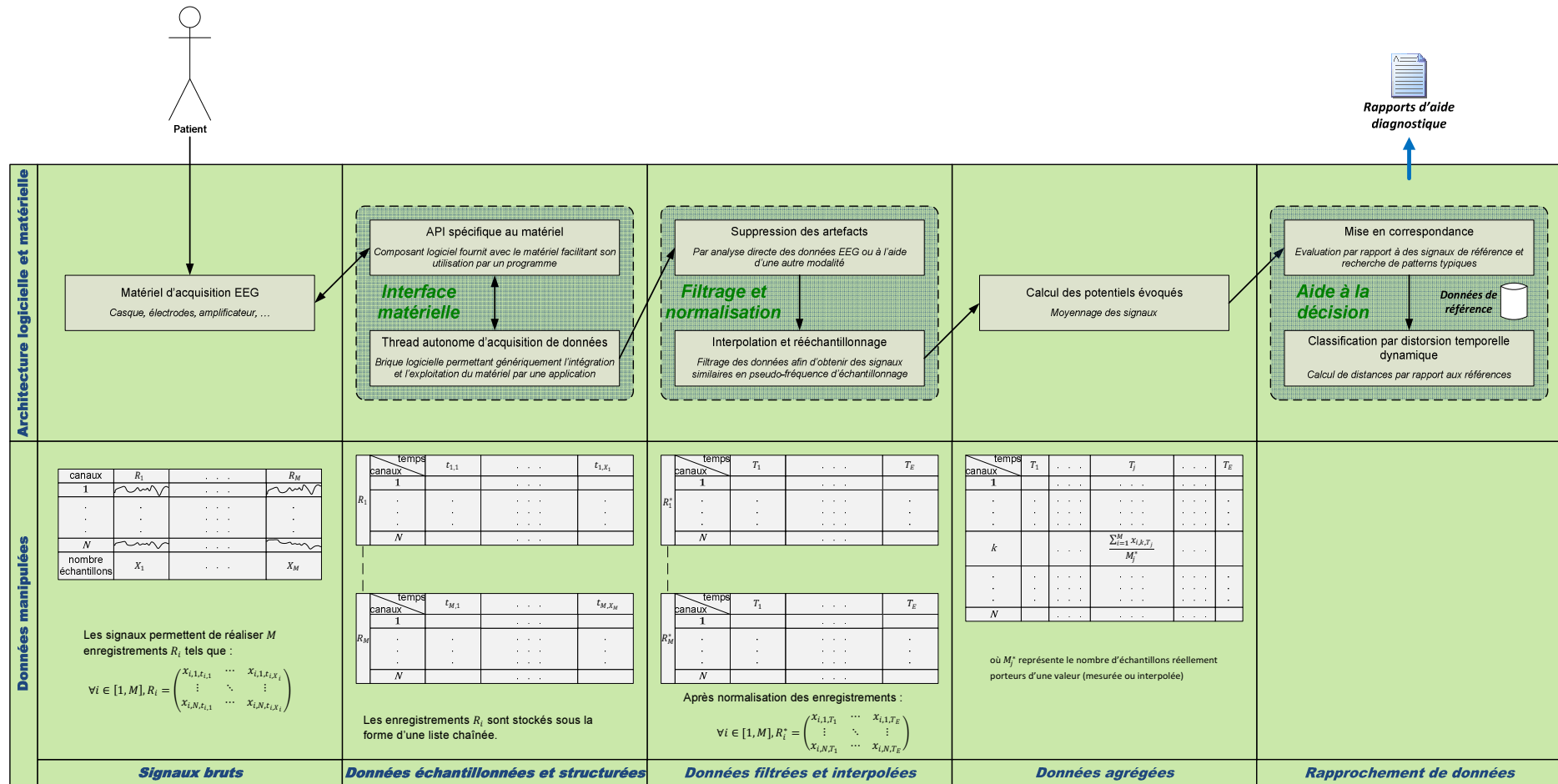


Figure 204 : Schéma du processus de traitement des données issues de l'EEG

C) L'intégration de données multimodales

Dans notre contexte applicatif, l'intégration de données multimodales passe par une fusion de ces données afin de pouvoir proposer une aide à la décision à des cliniciens. Pour conclure cette partie, nous allons examiner brièvement ces deux domaines de manipulation des données et expliquer en quoi nos choix techniques s'avèrent pertinents pour le développement d'applications sur la PAMMA les mettant en œuvre.

1) La fusion de données

La fusion de données est capable de combiner des informations partielles provenant de différentes modalités (Oviatt & Cohen, 2000; Turk, 2014). Le processus de fusion doit tenir compte de la temporalité dans l'intégration des données par le système (Nigay & Coutaz, 1993, 1995; Nigay et al., 2008) ainsi que des liens logiques et structurels entre elles (Nigay, 1994). Les mécanismes de fusion de données s'appliquent spécifiquement à des données provenant de modalités d'entrée (pour une revue, consulter (Lalanne et al., 2009)). La fusion multimodale est une étape critique et cruciale du traitement intelligent de données qu'est la combinaison et l'interprétation de différentes modalités entrantes (Zhen Zhu & Wang, 2006). Chacune de ces modalités est potentiellement porteuse d'informations partielles et/ou complémentaires qui peuvent être utiles pour le système. De la sorte, cette étape de fusion, qui doit prendre en compte la temporalité ainsi que les contraintes d'utilisation des différentes modalités entrantes, peut être de différentes natures (Lalanne et al., 2009; Oviatt, 2007; Srivastava & Dev, 2013; Tapia, de la Prieta, Rodríguez González, Bajo, & Corchado, 2011; Turk, 2014) :

- La fusion de données, qui consiste à combiner des données brutes afin de produire de nouvelles données brutes.
- La fusion de caractéristiques, qui consiste à fusionner les caractéristiques analysées à partir de modalités synchronisées.
- La fusion sémantique, qui consiste à unifier des événements et indicateurs issus de différentes modalités.

Ces différents types de fusion interviennent à différents niveaux de l'architecture logicielle, conformément à leur degré d'abstraction par rapport au matériel, et peuvent être réalisés par l'intermédiaire d'agents spécifiques (Nigay et al., 2008). Effectivement, les agents (voir la sous-sous-section **Les différents types d'agents et d'organisations d'agents**) sont adaptés pour la fusion de données car ils peuvent représenter des entités autonomes de fusion au travers de la modélisation de leurs capacités, de leur expertise et de leurs intentions (Tapia et al., 2011). La fusion de données multimodales devrait être assurée par des agents relativement intelligents (Wooldridge & Jennings, 1995b), c'est-à-dire qui seraient capables d'évaluer ces données et de prendre des décisions les concernant. Selon la nature des informations considérées, un agent responsable de la fusion de données peut être amené à prendre différentes mesures (Bouchet, 2006; Oviatt, 2007; Zhen Zhu & Wang, 2006) :

- Dans le cas d'informations concomitantes/redondantes, il peut réaliser l'action.
- Dans le cas d'informations incompatibles, il doit refuser l'action.
- Dans le cas d'informations complémentaires, il peut réaliser l'action résultante.

L'utilisation d'agents permet de décentraliser la fusion des données (pour un exemple de telle application dans le domaine de l'imagerie médicale, consulter (Richard et al., 2004)) et offre ainsi la possibilité au système d'évoluer dynamiquement avec des changements de son infrastructure (Tapia et al., 2011). En outre, les SMA prennent intrinsèquement en charge les traitements en parallèle et peuvent également fournir des moyens de calcul distribué. Cela s'avère particulièrement important dans le cas de la PAMMA. On signale aussi que notre méthode de suivi oculaire est déjà en soi une méthode de fusion de données multimodales, étant donné qu'elle associe des données relatives à l'œil et à la tête de l'utilisateur pour déterminer l'endroit où ce dernier regarde (voir la sous-section **La conception de notre dispositif de suivi oculaire**). De la même manière, l'identification des caractéristiques de la tête de l'utilisateur pratique déjà la fusion de données issues de différentes modalités. En effet, cette analyse combine des données provenant d'un capteur photographique RGB et d'un capteur de profondeur IR. Il s'agit ici de deux exemples, respectivement, de fusion de caractéristiques et de données brutes. On pourra aussi signaler que le casque EEG, que nous avons décidé d'utiliser (voir la sous-section **La conception de notre dispositif d'analyse de données EEG**), dispose de capteurs qui permettent d'estimer directement les mouvements de la tête de l'utilisateur et qui peuvent, par exemple, être utilisés pour la suppression d'artefacts sur les signaux EEG (O'Regan & Marnane, 2013).

Finalement, nous ajoutons qu'un système s'appuyant sur l'acquisition de données multimodales est moins sensible au bruit qui peut parasiter chacune des sources de données qu'il utilise. En effet, en fusionnant différentes sources de données redondantes et/ou complémentaires, un tel type de système sera plus robuste et plus précis (par exemple, consulter (Gurban & Thiran, 2010)). Ce dernier aspect est primordial pour un système, tel que le nôtre, qui doit procéder à des mesures en contexte clinique. Effectivement, la qualité de l'aide à la décision, qu'il est alors possible de proposer, est directement dépendante de l'exactitude des indicateurs considérés, qui peuvent être consolidés efficacement par fusion multimodale. L'intégration de données cliniques provenant de différents capteurs peut, par exemple, permettre l'élimination d'occurrences d'alarme qu'on pourrait qualifier de « faux positifs » (Arney et al., 2009).

2) L'aide à la décision

L'aide à la décision informatisée est une forme d'intelligence artificielle (Greenes, 2014). A ce titre, vouloir l'assurer au travers d'un SMA est particulièrement judicieux (Ferber, 1995; Wooldridge & Jennings, 1995b) (voir la sous-section **Les systèmes multi-agents**). Ainsi, il est logique de constater l'existence de SMA employés pour fournir de l'aide à la décision en milieu clinique (par exemple, consulter (Czibula et al., 2008)) ainsi que celle de méthodes d'ingénierie utilisées pour leur conception (par exemple, consulter (Wilk et al., 2008)).

L'aide à la décision consiste à proposer une interaction à l'utilisateur au travers de laquelle ce dernier peut introduire un problème auquel le système va tenter d'apporter des solutions (Greenes, 2014). L'utilisateur peut ensuite analyser les solutions proposées avant de prendre une décision finale (Bottrighi et al., 2013). Ainsi, un système d'aide à la décision est conçu pour assister un processus décisionnel (Sainfort et al., 2007). D'une manière générale, il est primordial que cette aide informatisée soit interactive et adaptée au contexte dans lequel elle est invoquée. En particulier, une telle aide doit permettre à la personne en charge de la

prise de décision d'accéder aux bonnes informations au bon moment indépendamment de la potentielle dissémination de ces dernières (Jennings, 1996). Dans le contexte médical, ce type d'approche est de plus en plus indispensable pour assurer la sûreté des patients (Bates & Gawande, 2003; Bates et al., 2001; Das et al., 1997). Pour notre système, l'aide à la décision s'appuie sur les différentes données auxquelles le système a accès et elle peut être divisée en deux parties (Isern et al., 2010) :

- Les tâches réalisées automatiquement par le système au travers de l'action d'agents.
- Les tâches s'exécutant sous la supervision d'un ou plusieurs utilisateurs.

La littérature des SMA précise d'ailleurs plusieurs moyens de réaliser ces différents types de tâches en milieu clinique (Isern et al., 2010) :

- Les assistants intelligents d'aide au diagnostic, qui vont pouvoir extraire des agrégats d'informations utiles aux cliniciens, à un moment donné dans le cadre de leur pratiques, à partir de multiples dossiers de patients et/ou de bases de connaissances (pour des exemples d'utilisation, consulter (Mateo, Cervantes, Yang, & Lee, 2007; Singh et al., 2005)).
- Les services de santé, permettant de réaliser le suivi dynamique par les patients de directives/recommandations cliniques, de processus de soins, et de proposer aux cliniciens des ajustements le cas échéant (Isern & Moreno, 2008).

Il existe également, indépendamment de tout choix technique, d'autres classifications des types de systèmes d'aide à la décision en milieu médical (par exemple, consulter (Berlin, Sorani, & Sim, 2006; Wright, Goldberg, Hongsermeier, & Middleton, 2007)). D'un point de vue fonctionnel, on y retrouve globalement les types d'aides relevant des primitives utilisées pour la modélisation des tâches de workflows (voir la sous-sous-section **Le niveau 3 de l'éditeur**). Ces aides peuvent, par ailleurs, s'adresser aux praticiens et/ou aux patients. Par exemple, des programmes d'aide à la décision peuvent être utilisés pour aider les cliniciens dans le choix du diagnostic et du traitement (Garg et al., 2005). En particulier, une telle assistance est fréquemment impliquée dans l'établissement de diagnostic différentiel (Greenes, 2014). Nous pensons donc qu'ils peuvent être utilisés efficacement dans le cadre du diagnostic de la dyslexie (par exemple, sous la forme d'un modèle multivarié (Le Jan et al., 2011)). De tels programmes disposent de procédures explicitement normalisées pour le processus de diagnostic, comprenant (Blonk et al., 1996) :

- Des connaissances récentes sur les problèmes de fond
- Des instruments d'évaluation
- Des alternatives de traitement

Au sujet des assistants intelligents, on peut signaler que différents types de critères sont susceptibles d'être évalués selon l'ambition du dispositif d'aide à la décision (**Figure 205**). En effet, la complexité des traitements n'est pas la même selon que l'on souhaite évaluer un seul ou plusieurs critères simultanément ou que l'analyse est ponctuelle ou longitudinale.

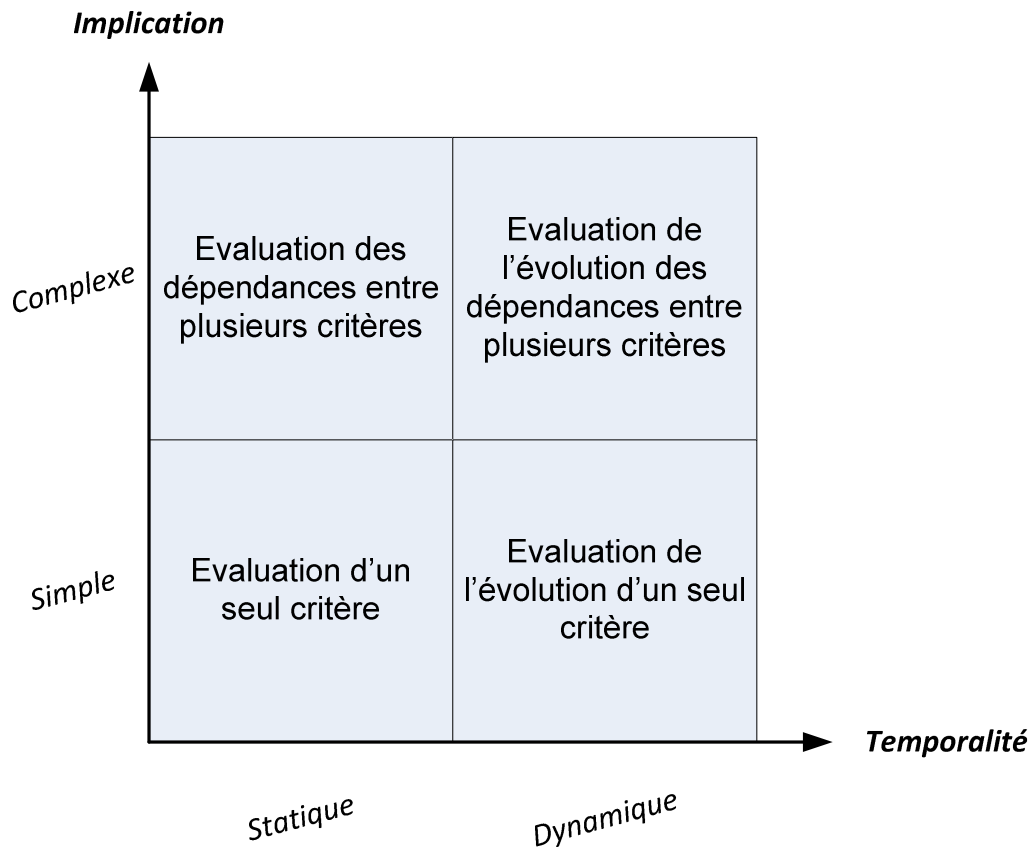


Figure 205 : Schématisation de la classification des critères d'évaluation impliqués dans la prise de décision

Ces différents types d'évaluations reposent sur des accès et/ou des recueils de données ainsi que sur le traitement de ces dernières. D'une manière générale, ces méthodes cherchent, à la suite d'un ou plusieurs examens cliniques, à classer des individus en différents groupes, ceux dont les membres souffrent probablement d'une pathologie et ceux dont les membres pourront être qualifiés de « sains ». Ainsi, pour permettre l'aide diagnostique ainsi que l'estimation d'un pronostic, il existe différents types de méthodes de détection et de classification comme, par exemple, les méthodes suivantes que nous nous contenterons d'évoquer (Greenes, 2014) :

- Les statistiques bayésiennes
- Les approches algorithmiques
- Les raisonnements heuristiques
- Les pondérations de résultats
- Les méthodes de reconnaissance de patterns

Dans le domaine de l'aide au diagnostic de la dyslexie, il existe, par exemple, des approches de reconnaissance de patterns basées sur les réseaux de neurones et/ou l'application de la logique floue (Costa et al., 2013; Jain, Manghirmalani, Dongardive, & Abraham, 2009). En particulier, on remarque que des mesures des mouvements oculaires sont fréquemment utilisées comme données d'entrée pour ce type d'aide diagnostique (pour des exemples de tels outils, consulter (Gomathi & Nasira, 2015; Macaš, 2005; Novák et al., 2004)). Ce genre d'approche est éminemment efficace, grâce à leur excellente précision pour un temps

d'exécution acceptable, dans d'autres domaines médicaux (Mateo et al., 2007)(pour un exemple d'application en cardiologie, consulter (Czibula et al., 2008)). Naturellement, il existe des SMA à usage clinique qui permettent d'effectuer de telles classifications (par exemple, consulter (González-Vélez et al., 2009)).

Pour un système d'aide à la décision, la capacité de pouvoir faire des suggestions, relevant de l'application d'un certain nombre de règles, est importante pour que l'utilisation du système soit couronnée de succès (Wright et al., 2009). Dans le cas du triage, il a été montré que les règles ainsi définies peuvent permettre de classer efficacement les patients (Blaszczynski et al., 2005). En outre, les performances des praticiens semblent être plus significativement améliorées lorsque l'aide à la décision est active et ne requiert pas qu'il y soit explicitement fait appel, c'est-à-dire lorsqu'elle relève d'un plus haut niveau d'intégration (Garg et al., 2005). Aussi est-il important de rappeler que tout système est faillible et que les systèmes d'aide à la décision ne font pas exception à cette règle. Ils doivent donc être considérés pour ce qu'ils sont, c'est-à-dire des aides, des outils qui ne sont pas nécessairement à même de gérer toutes les situations possibles. En aucun cas, ils ne peuvent remplacer le jugement d'un utilisateur expert qui pourra, le cas échéant, ne pas suivre les recommandations proposées (Lenz et al., 2012)(comme, par exemple, dans (Tu & Musen, 2000; Wang et al., 2004)). De la sorte, doit toujours être offerte la possibilité pour l'utilisateur de passer outre les recommandations du système, sous réserve de justifications (Greenes, 2014). De telles décisions de l'utilisateur prennent alors la forme de modifications dynamiques du workflow en cours d'exécution (Dadam et al., 2000; Ehrler et al., 2005; Wang, Li, & Wang, 2005). Le formalisme MA s'avère particulièrement approprié pour gérer ce genre de fonctionnement (Guo, 2007).

On signalera finalement que, bien que les aides à la décision puissent ne pas toujours améliorer les processus de travail, un impact neutre sur l'efficacité organisationnel devrait être le résultat minimum acceptable consécutif à la mise en place de tels systèmes. Les systèmes d'aide à la décision les plus efficaces devront ajouter de la valeur au workflow par l'amélioration, par exemple, de la qualité des soins ou de l'accès à l'information en milieu clinique, sans en augmenter le temps d'exécution et peut-être même en l'améliorant (Levick & Osheroff, 2014). On relèvera également que, bien qu'il soit avéré que l'utilisation de ce type de système peut permettre d'améliorer les performances des praticiens, l'établissement quantifiable et formel d'effets bénéfiques pour les patients reste toujours à dresser (Garg et al., 2005; Hunt et al., 1998).

L'analyse des résultats et les perspectives

Notre travail s'est préliminairement concentré sur l'identification et la proposition de solutions à des problèmes (**Tableau 1**), que nous pourrions qualifier de problèmes techniques, rencontrés lors de la prise en charge de patients souffrant de dyslexie. Nous avons alors défini plusieurs objectifs (**Tableau 2**) que nos réalisations devaient viser à satisfaire. Dans la présente section, nous allons débattre des résultats que nous avons obtenus lors de nos efforts pour atteindre ces buts préfixés.

Dans un second temps, nous présenterons les principales perspectives de nos travaux en termes de développements potentiels et nous proposerons un grand nombre de pistes destinées à en maximiser les chances de succès. En particulier, nous discuterons des processus de conception, d'implémentation et d'évaluation qui pourront être mis en place autour d'un groupe constitué, entre autres, d'experts métiers et d'informaticiens.

A titre informatif, une liste des ressources dont nous avons disposé durant nos recherches et développement est fournie en annexe (voir l'annexe **Les ressources utilisées pour cette recherche**).

I) Le bilan des objectifs initialement fixés

Au terme de notre analyse du besoin, nous nous étions fixé plusieurs buts à atteindre (voir le chapitre **Les objectifs fixés**). Par la suite, notre travail a ainsi consisté à poursuivre ces différents buts au travers de nos recherches et développement. Le moment est maintenant venu de faire le bilan de l'accomplissement de ces tâches.

De la sorte, pour chacun de nos objectifs, nous allons proposer :

- Une présentation fonctionnelle de la réalisation effectuée
- Une description technique de notre apport
- Une analyse critique et une conclusion quant à l'accomplissement de la tâche
- Des perspectives qu'ouvre ce travail

Une fois ce bilan de nos objectifs effectué, nous présenterons les différentes perspectives qui s'ouvrent alors à nous.

A) L'objectif **MODEL1**

Notre premier objectif était de modéliser un SI permettant la prise en charge de la dyslexie d'une manière générique. En effet, les connaissances métiers dans ce domaine (voir le chapitre **La dyslexie**) sont diverses et variées et aucun réel consensus ni méthode réellement efficace n'apparaît nettement. Nous voulions ainsi disposer d'un système pouvant être rapidement agrémenté de fonctionnalités propres à une expertise métier afin de pouvoir l'adapter rapidement à une pratique spécifique. Pour mieux comprendre la problématique, nous avons essayé d'étudier en quoi consiste le travail des professionnels du domaine et avons procédé à des entretiens avec plusieurs d'entre eux. Ces professionnels appartenaient à des spécialités différentes et exerçaient diverses fonctions dans la prise en charge des enfants dyslexiques. Le but de ces entretiens étant de nous donner une vue d'ensemble des pratiques cliniques en vigueur afin d'établir concrètement le besoin de nos utilisateurs.

Sur le plan technique, notre travail s'est alors déroulé en deux étapes :

- Une phase de recherche de méthodes :
 - De développement, qui nous a conduits à retenir la méthode de développement UP7 (Gabay & Gabay, 2008)(voir le chapitre **Le choix d'un modèle de développement**).
 - D'analyse, qui a abouti avec notre proposition de framework MA-MVC (voir la section **La perspective du méta-système**).
- Une phase d'application de la méthode retenue qui a consisté à :
 - Appliquer relativement strictement UP7, pour modéliser notre application de prise en charge de la dyslexie (voir le chapitre **L'application de l'UP7: un modèle générique de prise en charge de la dyslexie**).
 - A détailler le fonctionnement avancé des éléments du noyau de la PAMMA, afin de compléter notre modélisation de notre plate-forme conformément au framework que nous avons défini(voir le chapitre **L'intégration horizontale : support des processus métiers collaboratifs**).

A la vue des nombreuses spécifications que nous avons produites et de l'état de l'art que nous avons pu faire du domaine, il paraît légitime de dire que notre objectif a été atteint.

L'étape logique suivante de ce travail sera d'implémenter et de tester le système (voir la section **L'implémentation et les tests**). A ce titre, il reste une importante charge de travail à apporter au projet pour qu'il se concrétise. En particulier, il faut séparer d'un côté le développement du noyau de la PAMMA et d'un autre celui de l'application pour la prise en charge de la dyslexie. Cependant, la route a été relativement et exhaustivement jalonnée et il est maintenant possible d'estimer précisément le nombre d'acteurs, le coût et le temps de développement requis pour poursuivre le projet. Nous reviendrons d'ailleurs sur ces points plus loin dans ce document (voir le chapitre **Les développements à venir**).

B) L'objectif *PROTO1*

Notre objectif suivant était d'implémenter un composant logiciel permettant le suivi oculaire. Dans l'optique finale de développer une interface multimodale accessible au plus grand nombre, ce composant devait être à même d'utiliser des caméras bon marché. Pour cela, nous avons développé, à partir d'une étude approfondie des techniques existantes (voir la section **Le suivi oculaire**), notre propre série de traitements.

Sur le plan technologique, l'implémentation de ce prototype a été réalisée sous Windows (Microsoft, 2010) à l'aide des MFC 9.0 (Microsoft, 2008a). L'architecture du prototype repose sur un système de threads et applique une séparation entre l'interface utilisateur et le noyau fonctionnel (**Figure 206**). Ce découpage est matérialisé physiquement par une compilation produisant un .EXE pour l'interface utilisateur et des DLL pour les autres composants architecturaux. Les principaux fichiers d'entêtes des composants du noyau fonctionnel ainsi développés sont présentés en annexes (voir la section **Quelques fichiers d'entête** de l'annexe **Éléments des différents prototypes**). On remarquera, d'autre part, que les deux prototypes de nos objectifs *PROTO1* et *PROTO2* reposent sur cette même architecture et seront intégrés à un unique thread qui gère l'interface graphique.

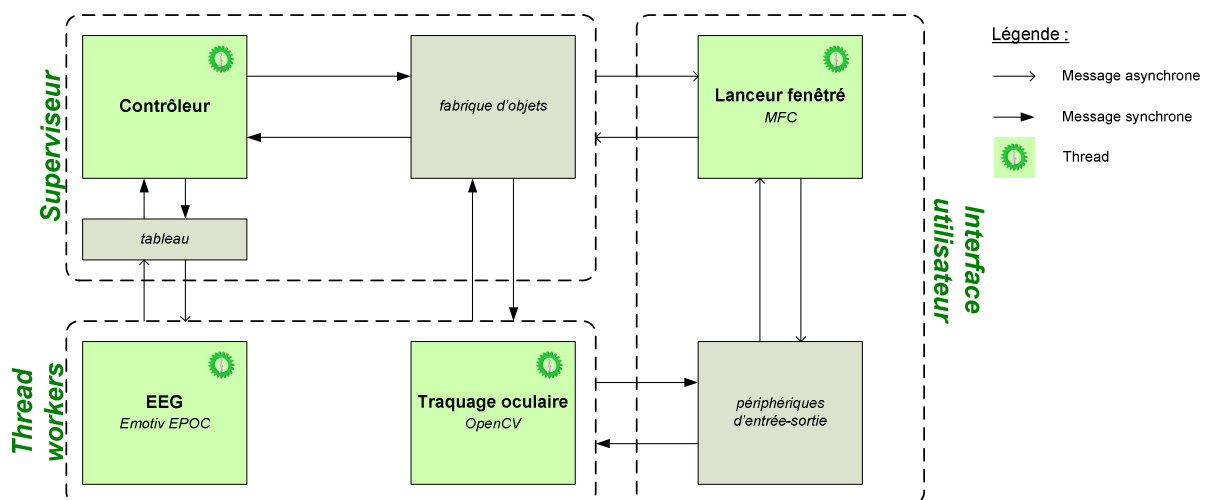


Figure 206 : Schéma de l'architecture logicielle des deux premiers prototypes

On peut aussi préciser que tous les agents de nos différents prototypes communiquent directement au travers de messages, qui, par commodité, ne se conforment pas nécessairement à un langage de communication entre agents, ou à l'aide d'un tableau noir qui centralise les informations sur lesquelles ils travaillent.

D'un point de vue fonctionnel, l'IHM de notre prototype de suivi oculaire se présente ainsi (**Figure 207**) :

- Une série de boutons permettant l'envoi des messages suivants :
 - o *MSG_EYE_START*, qui permet de charger la DDL de suivi oculaire.
 - o *MSG_EYE_STARTRECORD*, qui permet de débuter l'enregistrement vidéo et le suivi oculaire.
 - o *MSG_EYE_STOPANDSTARTRECORD*, qui permet d'arrêter l'enregistrement vidéo courant puis d'en débuter un autre.
 - o *MSG_EYE_STOPRECORD*, qui permet d'arrêter l'enregistrement vidéo en cours.
 - o *MSG_EYE_STOP*, qui permet de télécharger la DDL de suivi oculaire.
- Une zone dans laquelle s'affichent les images en, cours de traitement.

L'interface graphique dispose également d'autres éléments qui sont utilisés par le composant d'enregistrement et de traitement des signaux EEG (voir la section **L'objectif PROTO2**).

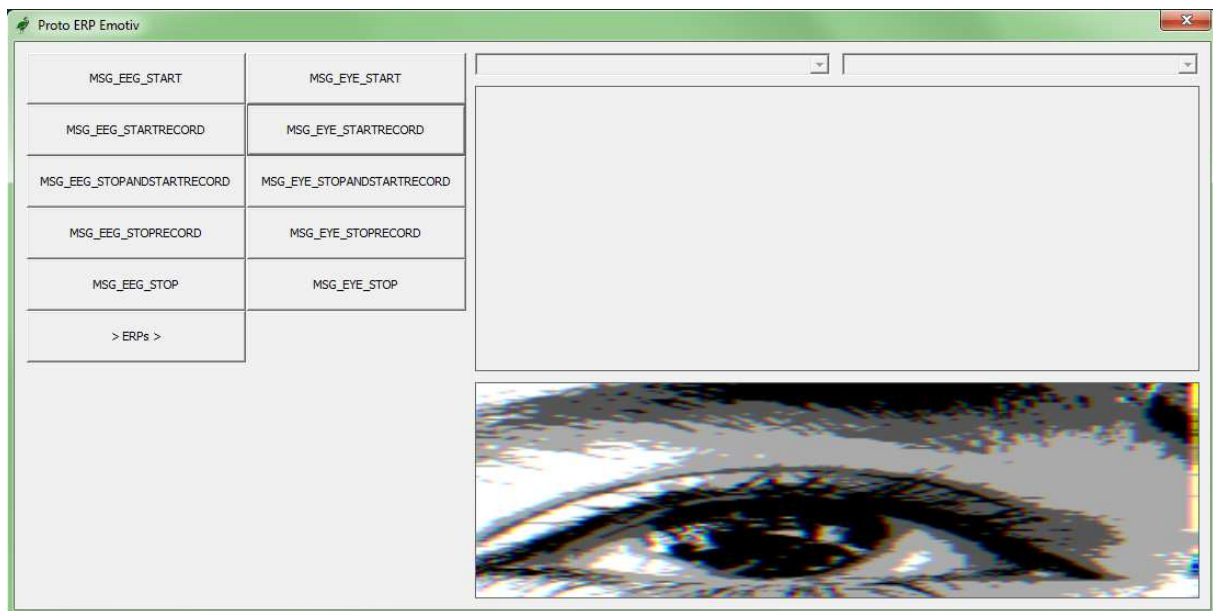


Figure 207 : Impression d'écran du prototype réalisé dans le cadre de l'objectif **PROTO1**

Malheureusement, l'entreprise partenaire du projet s'est retrouvée en redressement judiciaire et n'a donc pas financé le prototypage comme elle s'était initialement engagée à le faire. Il n'a donc pas été possible de poursuivre le développement du composant logiciel de suivi oculaire qui devenait alors impossible à tester et évaluer. Cet objectif ne peut pas, à ce titre, être considéré comme ayant été atteint. En effet, notre processus de traitement d'images est d'une telle complexité qu'il serait absolument irrecevable de le valider après s'être affranchi de son évaluation approfondie.

Bien que nous n'ayons pas pu complètement appliquer notre étude théorique de la problématique du suivi oculaire, nos recherches fondamentales n'en sont pas moins pertinentes. D'autre part, les résultats que nous avons tout de même obtenus concernant ceux de nos traitements que nous avons pu tester, nous incitent à continuer dans cette voie. Le prochain palier à franchir serait de poursuivre l'implémentation et la validation de nos développements à l'aide du prototype de casque pour le suivi oculaire qui a récemment pu être imprimé à l'aide d'une imprimante 3D (voir le chapitre **Premiers résultats** de l'annexe **Le prototypage de l'interface multimodale**). Il sera ensuite possible de faire passer une batterie de tests (comme, par exemple, ceux proposés dans (Figueira, de Azevedo, Russomano, & Lilienthal, 2008)) à notre système de suivi oculaire afin d'en garantir une précision suffisante pour son utilisation en milieu clinique, ainsi que ces performances en termes d'utilisabilité par rapport à d'autres dispositifs du même genre (San Agustin Lopez et al., 2010). D'autre part, une validation en situation réelle et pratiquée sur différents sujets humains sera également nécessaire à la validation de l'utilisabilité du dispositif final.

C) L'objectif *MODEL2*

Afin de procéder à l'acquisition des données qui doivent nous permettre de réaliser le suivi oculaire, nous avons été amenés à réfléchir au prototypage d'un nouveau matériel. En effet, notre étude et les traitements que nous avons mis en évidence requièrent d'utiliser certains types de capteurs agencés selon une géométrie spatiale spécifique (voir la section **Le suivi oculaire**).

Néanmoins, pour les raisons citées dans la section précédente, il ne nous a pas été possible d'aller plus loin que l'élaboration d'un cahier des charges fonctionnel (voir l'annexe **Le prototypage de l'interface multimodale**). Notre travail a été ainsi essentiellement théorique et a consisté à rassembler sur un même document l'ensemble de nos besoins fonctionnels.

D'un point de vue s'attachant strictement à notre objectif, notre tâche a, cependant, été entièrement accomplie. Un premier prototype physique du support du casque permettant le suivi oculaire a même été imprimé en imprimante 3D (comme présenté sur les photographies visibles en annexe dans la section **Premiers résultats**).

Cette première étape de conception de matériel s'inscrit dans l'élaboration d'un casque d'acquisition multimodale (voir le chapitre **La réalisation d'un casque « tout-en-un »**). Le travail sur cette modalité de suivi oculaire pourra ainsi être poursuivi, avant d'être progressivement enrichi par les modalités d'EEG et d'analyse de la parole.

D) L'objectif *PROTO2*

Cet objectif a consisté au développement de notre deuxième prototype. Ce prototype devait nous permettre de démontrer l'utilisabilité d'un matériel d'acquisition EEG bon marché et simple d'utilisation dans un contexte clinique et/ou de recherche. Nous avons pour cela utilisé le casque Emotiv EPOC® (Emotiv Inc., 2015) qui dispose d'une fréquence d'échantillonnage de 128 Hz. En nous appuyant sur une étude des spécifications techniques

du domaine, nous avons décidé de procéder au recueil de données EEG brutes ainsi qu'au calcul de potentiels évoqués (voir la section **L'électroencéphalographie**), la finalité de ce prototypage étant l'intégration des composants logiciels ainsi développés à notre interface multimodale.

Comme nous l'avons précédemment indiqué, ce prototype utilise la même interface graphique que celui de notre objectif **PROTO1** (voir la section **L'objectif PROTO1**). Les boutons spécifiques au composant d'enregistrement et de traitement des signaux EEG permettent l'envoi des messages suivants (**Figure 206**) :

- *MSG_EEG_START*, qui permet de charger la DDL pour l'EEG.
- *MSG_EEG_STARTRECORD*, qui permet de débuter l'enregistrement EEG.
- *MSG_EEG_STOPANDSTARTRECORD*, qui permet d'arrêter l'enregistrement EEG courant puis d'en débuter un autre.
- *MSG_EEG_STOPRECORD*, qui permet d'arrêter l'enregistrement EEG en cours.
- *MSG_EEG_STOP*, qui permet de décharger la DDL pour l'EEG.
- *> ERPs >*, qui permet d'interpoler puis de moyenniser les différents enregistrements qui ont été réalisés.

On notera également la présence de deux menus déroulants aux fonctions suivantes :

- Sélectionner un enregistrement. On distingue d'ailleurs trois types d'enregistrements (voir la section **L'électroencéphalographie**) :
 - o Les enregistrements de données brutes, qui correspondent à une suite de données transmises directement par les capteurs.
 - o Les enregistrements comprenant des données interpolées, qui sont calculées de sorte à ce que les différents enregistrements de données brutes se retrouvent avec des peignes d'échantillonnage identiques.
 - o Les enregistrements composés en moyennant des enregistrements de valeurs interpolées, qui permettent de calculer les potentiels évoqués.
- Sélectionner une électrode parmi les quatorze utilisées.

Pour finir, nous avons intégré un indicateur de confiance à notre affichage. Effectivement, étant donné que notre calcul de potentiels évoqués repose en partie sur des valeurs interpolées, il nous paraissait approprié de disposer d'un indicateur de fiabilité de ce calcul. L'indicateur, qui se présente sur la forme d'un frise hachurée de couleurs, permet de juger selon deux dimensions de la qualité de la moyenne obtenue à partir des signaux enregistrés (**Figure 206**) :

- La couleur permet d'estimer la proportion de données brutes intégrées dans le calcul de chaque valeur du signal moyen. Le but étant de pouvoir établir si cette moyenne repose essentiellement sur des valeurs réellement mesurées, qui sont codées par le **vert**, ou sur des valeurs interpolées, qui sont codées par le **rouge**.
- La hauteur permet d'estimer le nombre de valeurs utilisées pour moyenniser une valeur donnée du signal de résultat. Ainsi plus la bande est large plus le nombre d'échantillons utilisés, qu'ils soient interpolés ou non, est important. Une réduction est également effectuée pour normaliser la taille du rendu.

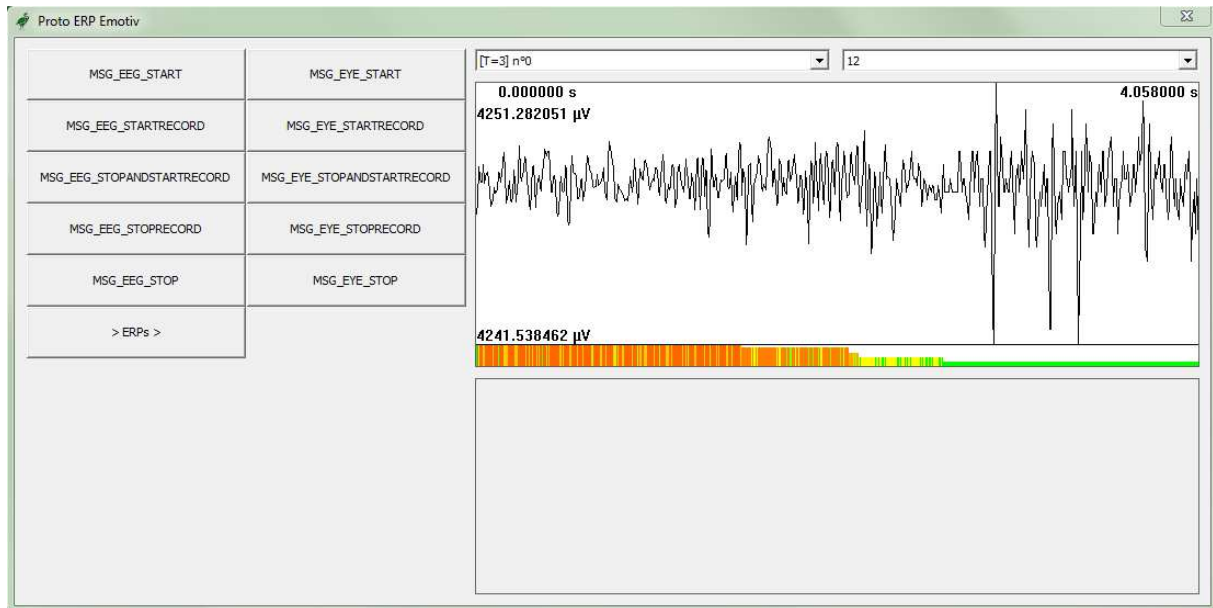


Figure 208 : Impression d'écran du prototype réalisé dans le cadre de l'objectif **PROTO2**

Cette fois encore, les choses ne sont pas déroulées pour le mieux. En effet, le professeur en médecine, qui était supposé détenir l'expertise neurophysiologique nécessaire pour savoir quoi mesurer, ainsi qu'ou et quand le mesurer, s'est désengagé du projet après avoir quitté son institution dès le début de ce travail de thèse. Cette personne était également sensé nous donner accès à du matériel homologué pour un usage clinique ainsi que, le cas échéant, à des patients. L'étude de la littérature pour un non-initié ne remplaçant pas une telle expertise ni un accès à du matériel de référence, il n'a pas été réellement possible de tester notre prototype en situation concrète. A ce titre, notre objectif ne peut pas être considéré comme achevé ni notre preuve de concept comme probante, même si le prototype semble pouvoir correctement enregistrer des signaux EEG et procéder au calcul des potentiels évoqués.

Ce composant logiciel est destiné, comme celui de suivi oculaire, à être intégré à notre interface multimodale. Il pourrait d'ailleurs s'avérer très important dans le contexte d'une avancée technologique pour le diagnostic de la dyslexie car l'EEG n'est, à l'heure actuelle, pas utilisé en milieu clinique alors que, selon la littérature, il pourrait être d'un intérêt majeur. Une expertise fiable dans ce domaine sera donc la prochaine nécessité pour pouvoir poursuivre efficacement ce travail.

E) L'objectif **PROTO3**

Notre dernier objectif concernait l'implémentation d'un test d'évaluation de la lecture utilisé couramment en milieu clinique. Nous avons choisi d'implémenter le test de lecture de mots fréquents et peu fréquents extrait de la méthode B.A.L.E. (Jacquier-Roux et al., 2010). Ce test consiste à présenter à l'utilisateur des mots selon des séries préétablies disposant chacune de caractéristiques propres (pour les listes de mots utilisés, consulter le chapitre **Les listes de mots du test de lecture de mots fréquents et peu fréquents** de l'annexe **Eléments des différents prototypes**). D'autre part, cette implémentation devait être réalisée de telle sorte

à ce que ce prototype puisse être utilisé en milieu immersif. Effectivement, les interactions sous-tendues par ce type d’environnements peuvent être d’un intérêt dans le cadre de la prise en charge clinique de troubles neuropsychologiques (Klinger, 2006; Rizzo et al., 2000).

Il est possible d’utiliser des plates-formes initialement destinées aux jeux vidéo pour concevoir des systèmes distribués à visées médicales (Cha, Wood, & Finkelstein, 2012). Ce genre de plates-formes logicielles, en grande partie par l’intermédiaire des consoles et matériels les exploitant, présente un intérêt au niveau de leur prix, de leur disponibilité, de leur simplicité d’utilisation et des possibilités qu’elles offrent en termes d’IHM. A ce titre, nous avons décidé d’utiliser l’environnement de développement d’Unity (pour un ouvrage sur l’utilisation concrète de cet environnement de développement, consulter par exemple (Goldstone, 2010)). Cet environnement a le mérite de permettre la compilation d’applications pour les plates-formes vidéoludiques les plus répandues. D’autre part, un énorme intérêt que peut avoir l’utilisation du logiciel Unity pour générer des interfaces graphiques est le fait que ce logiciel dispose de son propre moteur physique et graphique 3D. On signalera que l’utilisation de telles technologies peut également tirer parti de la prédisposition humaine au jeu (voir la sous-section **L’approche ludique et immersive**)).

L’exécution d’un projet sous Unity, consiste en l’instanciation de scènes qui contiennent les entités, c’est-à-dire les objets de l’application. La **Figure 209** présente succinctement les objets permettant d’instancier des scènes avec Unity.

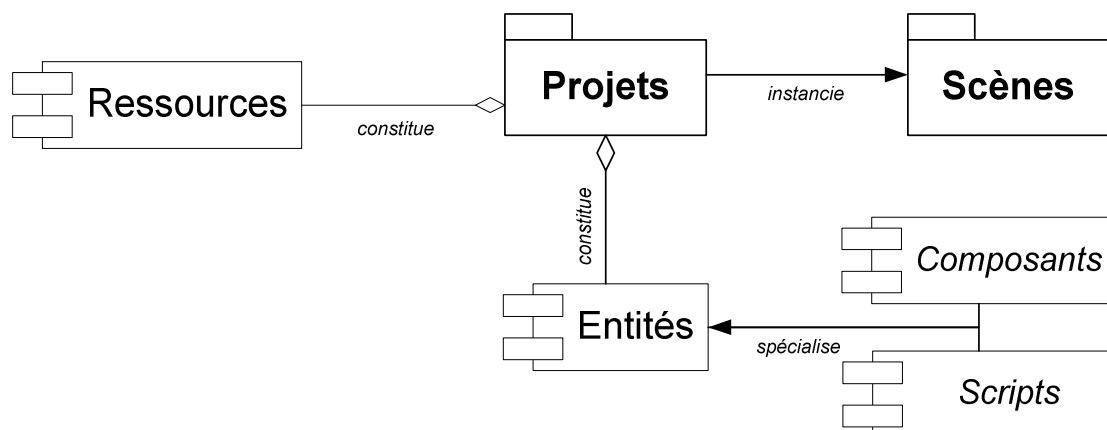


Figure 209 : Schéma des relations entre objets constitutifs des scènes sous Unity

Sur le plan fonctionnel, l’application consiste à présenter successivement des mots à l’utilisateur qui doit les lire. Il devient alors possible de mesurer ses temps de réponse entre le moment de la présentation d’un mot et sa réponse, ainsi que la qualité de ses réponses en termes d’exactitude de la lecture effectuée. Au lancement de l’application, l’utilisateur va pouvoir choisir quelle liste de mots va être présentée. Nous avons également fait en sorte que les couleurs des mots et du fond puissent être modifiées car il est reconnu que ces paramètres peuvent influencer les performances des dyslexiques en lecture (Altschuler et al., 2007). La **Figure 210** présent un exemple de l’interface du prototype pour lequel les caractères sont affichés en **vert** sur fond **rouge**.

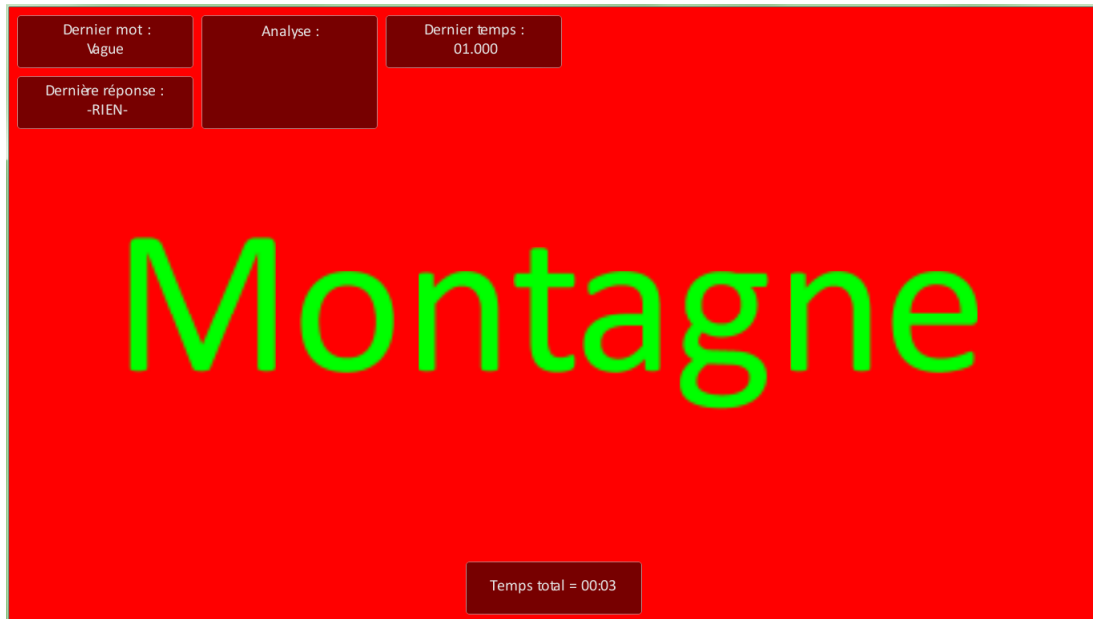


Figure 210 : Impression d'écran du prototype réalisé dans le cadre de l'objectif **PROTO3**

On signalera que ce prototype devait pouvoir, au final, servir à intégrer les différentes briques des deux précédents prototypes entre elles (**Figure 211**). Cependant, pour les raisons budgétaires précédemment énoncées, nous n'avons pas pu bénéficier d'une licence professionnelle d'Unity. Or, seul ce type de licence permet l'utilisation de DLL dans un projet réalisé avec ce logiciel. Il ne nous a donc pas été possible de procéder à l'intégration des différents prototypes que nous avons développés.

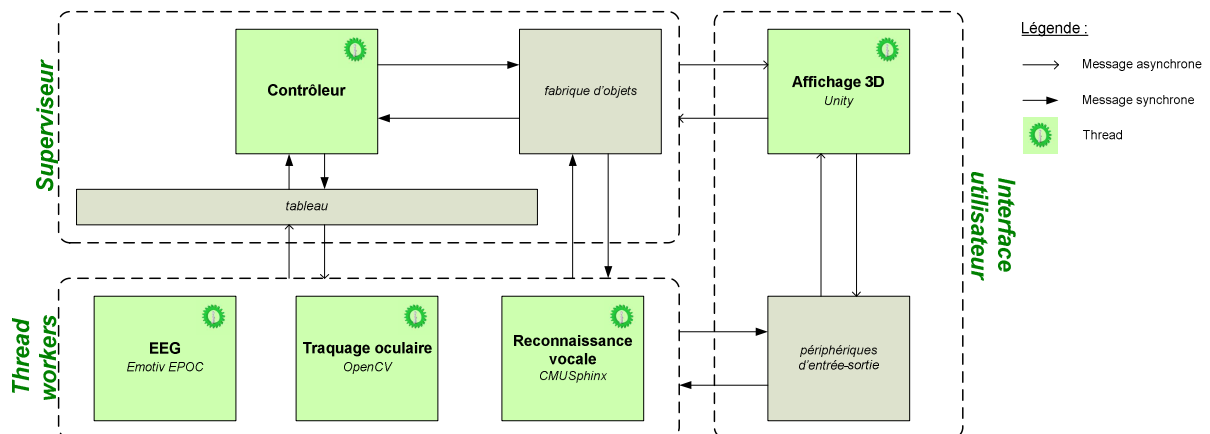


Figure 211 : Schéma de l'architecture logicielle envisagée pour l'intégration des trois prototypes

Pour les raisons que nous avons présentées, il ne nous a pas été possible d'automatiser complètement ce test comme nous l'aurions souhaité. A ce titre, notre objectif ne peut pas être considéré comme atteint. Néanmoins, nous sommes parvenus à implémenter à l'aide d'Unity une tâche de présentation de mots conforme à celle décrite dans la méthode B.A.L.E. (Jacquier-Roux et al., 2010). Nous l'avons également adjointe d'une partie de paramétrage, permettant d'aller plus loin que la description présentée par la méthode, ainsi que d'une séparation entre affichage et données, offrant des possibilités de modifications rapide, par exemple, des listes de mots. Notre objectif peut finalement être estimé partiellement rempli

dans la mesure où nous avons tout de même réussi à implémenter en environnement immersif un test d'évaluation de la lecture reconnu et extrait de la littérature.

D'un point de vue des perspectives de ce prototype, on signalera qu'au sein de l'environnement de développement proposé par Unity, il est également possible de définir des objets appelés « Prefabs » qui ne sont ni plus ni moins que des classes d'entités. Cela pourrait être particulièrement utile dans le cadre du framework que nous voulons réaliser. En effet, de telles classes pourront dynamiquement être paramétrées et instanciées afin de correspondre aux besoins spécifiques d'un processus donné. Pour provoquer l'instanciation dynamique de scènes, nous prévoyons ultérieurement d'ajouter un préprocesseur XML (World Wide Web Consortium, 2013), couramment appelé « parser », aux entités intégrées à notre projet. Le moteur, lors de son initialisation, pourra ainsi être en mesure d'exécuter un script utilisant ce parser afin de recevoir en entrée le descriptif du processus de soins à exécuter au format XML. Dès lors, notre projet Unity peut être utilisé comme une boîte à outils qui prendra en entrée le descriptif complet d'un processus et l'instanciera à l'aide des « Prefabs » dont elle dispose. Ce projet exhibe ainsi des capacités similaires à celles recherchées pour un service de workflows de notre PAMMA exploitant des fichiers/données encodés (voir la sous-section **L'encodage**). A ce titre, ce prototype est voué à évoluer pour réellement devenir une application de la PAMMA.

II) Les développements à venir

Le bilan de nos objectifs initiaux nous informe que notre travail théorique, constitué d'une partie de recherche et d'une autre de modélisation, a été productif. En effet, les différents axes d'amélioration de la prise en charge des individus dyslexiques que nous avons identifiés pendant notre analyse de la problématique (voir le chapitre **Les axes d'amélioration technique de la prise en charge**) ont été couverts par notre étude et des solutions conceptuelles ont été proposées. A contrario, le passage à des réalisations pratiques, incarnées par des prototypes, n'a quant à lui pas été un franc succès faute de moyens techniques et financiers. A ce titre, une partie importante des perspectives que nous pouvons identifier concerne les moyens à mettre en œuvre pour réaliser concrètement tout ou partie des développements que nous avons spécifiés dans ce manuscrit.

En particulier, une des perspectives directes de ce travail concerne l'implémentation du SI que nous avons modélisé à l'aide de la démarche UP7 (Gabay & Gabay, 2008). En effet, notre étude nous a conduits à produire une représentation théorique et générique d'un système de prise en charge de la dyslexie (voir le chapitre **L'application de l'UP7: un modèle générique de prise en charge de la dyslexie**). Il resterait maintenant à faire se concrétiser cette vision en réalisant un code exécutable. Pour cela, il faudrait poursuivre la démarche avec ses phases d'implémentation et de test. On pourra d'ailleurs rappeler que cette concrétisation du système passe par deux réalisations distinctes :

- Le framework, c'est-à-dire le noyau de la PAMMA et les outils qu'elle requiert.
- Le contenu spécifique, c'est-à-dire la modélisation de processus métiers spécifiques destinée à être intégrée par la PAMMA.

La prochaine étape à franchir pour pouvoir passer à l'implémentation de ce système serait l'obtention d'un financement. En effet, comme nous l'avons dit précédemment, la réalisation d'un tel système nécessite une grosse charge de travail. Il faudra donc mobiliser un certain nombre de développeurs autour de cette phase de programmation. Etant donné l'importance qu'à la documentation produite dans le cadre de notre processus de développement, il faut aussi prévoir des ressources spécifiques à sa réalisation (Priestley & Utt, 2000). D'ailleurs, un projet informatique se construit principalement autour de deux choses (Guérin, 2012) :

- Les buts du projet, c'est-à-dire la vision concrète du futur qui est soutenue par le projet.
- Les protagonistes du projet, qui sont les personnes constituant l'équipe du projet et partageant une vision unique de ce dernier.

Les buts de notre projet sont clairement établis :

- Implémenter une plate-forme d'applications médicales reposant sur notre framework (voir la sous-section **La perspective du méta-système**), c'est-à-dire la PAMMA.
- Concevoir des applications destinées à fonctionner sur la PAMMA, en particulier notre application de prise en charge de la dyslexie.

Ainsi, dans ce chapitre, nous allons présenter les acteurs de ces réalisations ainsi que des ébauches des processus coopératifs de développement les impliquant dans

l'accomplissement de ces deux buts. D'autre part, nous relèverons les difficultés qui pourront être rencontrées au cours de tels développements, auxquelles nous proposerons des pistes de solutions.

A) L'implémentation du noyau de la PAMMA

L'essentiel de notre travail a consisté à concevoir un système qui garantisse l'interopérabilité des informations, des traitements de données et des matériels dans le but de procéder efficacement à la prise en charge des individus souffrant de troubles neuropsychologiques, et plus particulièrement de la dyslexie, en permettant implémentation d'applications spécifiques. Cet apport s'inscrit dans un effort visant à fédérer les différents corps de métiers impliqués dans le traitement de ces pathologies autour d'un système reposant sur une architecture modulaire et sur l'utilisation de nombreux standards. Il y a d'ailleurs, pour ce type de technologies, un réel besoin clinique dont les enjeux financiers sont conséquents (Walker et al., 2005). Outre l'utilité évidente pour les patients et les professionnels de santé dont nous avons déjà discuté, cet état de fait nous conforte dans la viabilité économique d'une telle recherche et des développements qui pourront en découler.

Un tel système est destiné à s'intégrer à un SI existant ou à le remplacer. A ce propos, il faut rappeler que la mise en place d'un SI ne se traduit pas nécessairement par une augmentation de l'efficacité organisationnelle (Rahimi, Vimarlund, & Timpka, 2009). A ce titre, il faut prendre en compte un certain nombre de facteurs sociaux, organisationnels et économiques pour maximiser les chances de succès du système. Dans le cas de l'implémentation du noyau de la PAMMA, la plupart de ces considérations, hormis celles relevant de l'aspect financier, ne rentre pas directement en ligne de compte. Effectivement, les utilisateurs verront uniquement leurs pratiques influencées par les applications fonctionnant sur la PAMMA et non par le noyau de cette dernière. Nous reviendrons sur les facteurs liés à l'implémentation des applications sur la PAMMA un peu plus loin dans cette partie (voir la section **La conception d'applications pour la PAMMA**). L'implémentation du noyau de la PAMMA va se heurter principalement à des difficultés techniques et technologiques.

Le point le plus important de tous, car il conditionnera véritablement l'utilisabilité du système au sein des écosystèmes informatiques du milieu de la santé, est l'interopérabilité du système au travers des standards qu'il pourra supporter. Comme nous l'avons vu tout au long de ce travail, ces standards régissent les possibilités d'interaction/intégration entre différents systèmes et/ou composants de systèmes (voir la section **L'intégration des processus métiers**) ainsi que le niveau de confiance global qui peut être accordé au dispositif (voir la section **L'intégration des données**). En particulier, il faut que le service de workflows implémente les interfaces standard garantissant son interopérabilité avec d'autres systèmes du même genre (Hollingsworth, 1995, 2004)(voir la section **Le support au travail collaboratif**). De cette manière, les applications développées sur la PAMMA pourraient être portées vers un autre système à moindre frais et réciproquement.

Concrètement, l'implémentation du noyau de la PAMMA va se composer du développement des éléments suivants de notre système :

- Le framework (voir la sous-section **La perspective du méta-système**).
- Les principaux services de la plate-forme (voir le chapitre **L'application de l'UP7: un modèle générique de prise en charge de la dyslexie** et la sous-section **L'exécution**).
- L'éditeur (voir la sous-section **La spécification**).

Contrairement à notre travail de développement qui a, jusqu'à maintenant, essentiellement consisté à produire des spécifications, ces réalisations vont nécessiter une méthode adaptée à une implémentation réactive. En effet, ce système est complexe et tous ses aspects sont susceptibles de changer en fonction de nouveaux besoins que nous pourrions rencontrer pendant son codage. A ce titre, il semble judicieux d'avoir recourt à une méthode Agile, car une telle méthode vise à maximiser la satisfaction du client en focalisant continuellement l'effort de développement sur la réponse aux besoins de ce dernier. Plus précisément, nous pensons utiliser la méthode Scrum (Aubry, 2013) qui est caractérisée par un découpage en « sprints » de quelques semaines. Chaque sprint se termine par la production d'un livrable. La **Figure 212** propose une vision d'ensemble du processus de développement de cette méthode. On signalera l'existence de deux rôles importants au sein d'une équipe appliquant la méthode Scrum :

- Le « product owner », qui joue le rôle d'un gestionnaire de projet.
- Le « scrum master », qui joue le rôle du manager de l'équipe de développement.

Les développeurs occupant ces responsabilités auront, respectivement, en charge l'établissement du processus de développement et son suivi au jour le jour. A ce titre, durant chaque sprint, une réunion hebdomadaire sera organisée pour contrôler l'adaptation dynamique du processus (**Figure 213**).

On signalera finalement que le travail théorique, que nous avons présenté dans ce manuscrit, pourra servir de point de départ lors du sprint 0. En effet, à part le choix d'une technologie pour l'implémentation et la composition d'une équipe de développeur, nous avons couvert tous les autres éléments nécessaires à la mise en place d'un tel processus de développement.

Hormis pour le développement du noyau de la PAMMA, on pourra remarquer que la méthode Scrum peut également être mise en œuvre lors du développement d'application sous la PAMMA. Effectivement, cette méthode de développement est souple, ce qui permet d'utiliser des modèles provenant de différentes autres méthodes afin de couvrir les besoins spécifiques de notre contexte de développement d'applications médicales (Abouabdellah, Aguezoul, & Abdelghani, 2010).

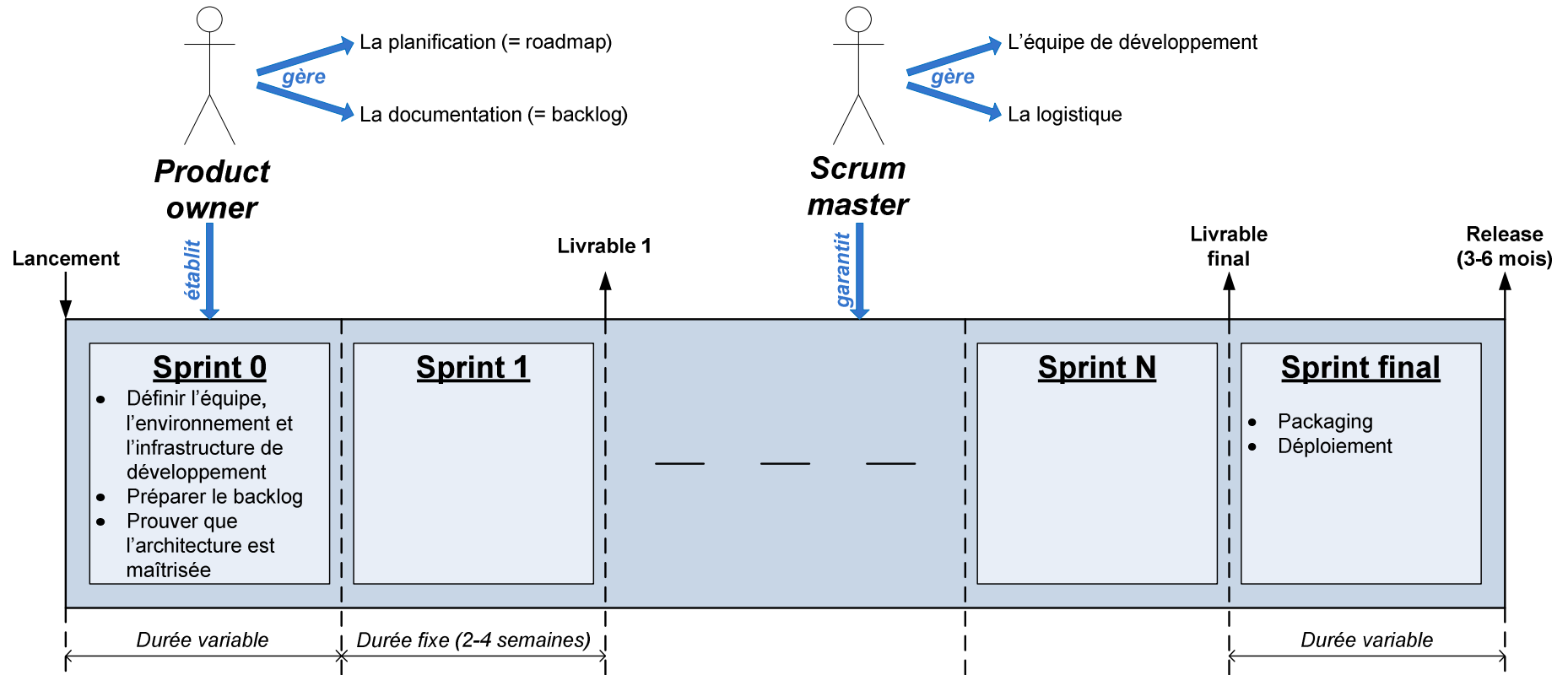


Figure 212 : Chronogramme des activités de développement de la méthode Scrum

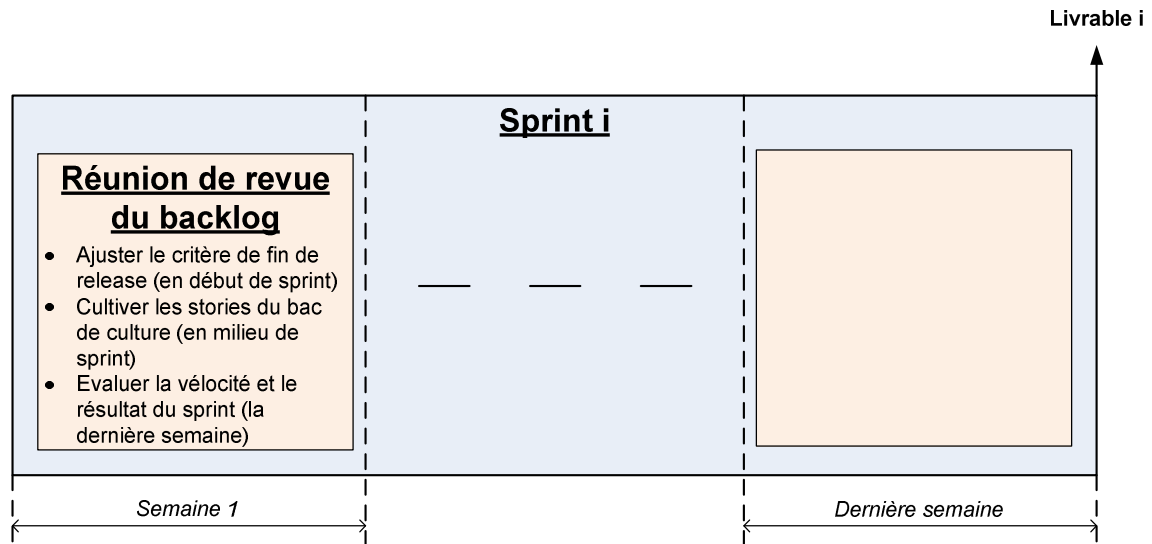


Figure 213 : Détail de l'activité de gestion d'un sprint

La dernière étape nécessaire à l'industrialisation du dispositif est la phase de test (pour un rappel sur les tests et leurs objectifs, voir la sous-section **Les tests**). Pour vérifier que les besoins fonctionnels sont couverts, c'est-à-dire établir si le système fait bien ce pour quoi il est conçu, il faudra probablement conduire des études cliniques. A ce titre, la PAMMA ne sera réellement testable sur ce plan qu'au travers de l'évaluation d'applications fonctionnant dessus. Dans le cas de l'application de prise en charge de la dyslexie, cela signifie qu'il faudra contrôler que l'application présente une efficacité quantifiable pour le diagnostic et le traitement de la pathologie. Pour cela, notre système se propose d'offrir une métrique portant sur la quantification de l'effet de la remédiation. Effectivement, il est envisagé qu'un modèle de simulation soit intégré au système (voir la section **Les exigences fonctionnelles**) pour reproduire le comportement de l'individu à partir de données recueillies à un instant choisi. Après remédiation, la comparaison entre les résultats fournis par le modèle, qui est paramétré à partir de performances constatées au préalable, et les performances obtenues par le patient, devrait nous permettre de quantifier l'impact de la remédiation sur ce dernier. De la même façon, les effets de la compensation pourront être évalués et comparés à ceux obtenus à l'aide d'autres approches.

Dans un registre concernant exclusivement les performances de la plate-forme, il faudra porter un soin particulier à l'attribution des ressources du système aux agents ainsi qu'aux temps de réponse du système qui sont des caractéristiques critiques d'un tel système distribué (Larson et al., 2012). Il est également primordial de prévoir que le système effectue des contrôles, à un niveau central et/ou local, afin de s'assurer que les ressources, en termes de matériels et de puissance de traitement nécessaires à l'exécution normale de chaque application, sont bien remplies. Pour évaluer le fonctionnement du système, il faudra aussi s'assurer de la bonne utilisabilité de ce dernier. Cette utilisabilité est une caractéristique multidimensionnelle qui consiste à juger de l'efficacité, de l'efficience et de la satisfaction que procure l'utilisation du système. Une telle évaluation peut intégrer des métriques objectives et/ou subjectives calculées pendant ou après une utilisation du système par un utilisateur. Par exemple, il est primordial de recueillir les avis des utilisateurs vis-à-vis de celui-ci (Lewis, 1995)(comme, par exemple, cela est pratiqué dans (Aigner & Miksch, 2006; Juristo et al., 2007; Kosara & Miksch, 2001; Purves et al., 1999)). Il devient ainsi plus facile de

procéder aux changements qui s'imposent afin de maximiser la satisfaction de ces utilisateurs par rapport au système et de ses fonctionnalités, comme l'exige un processus de développement Agile.

On pourra également rappeler que la réalisation de notre PAMMA va reposer sur des infrastructures et applications existantes. L'intérêt du recours à notre système est qu'il intègre nativement un gestionnaire de processus. Il va ainsi pouvoir être possible, au niveau de notre système, de représenter tout ou partie d'un autre système à l'aide d'un ou plusieurs agents. Ces agents vont ainsi pouvoir être utilisés, comme tout autre agent de notre paradigme, pour décrire des processus métiers. Il va donc être, par exemple, possible de combiner les fonctionnalités d'agents « externes » d'un autre système avec certaines autres spécifiquement développées pour notre système, ou encore d'organiser des processus ne faisant appel qu'à des agents « externes ». Une telle intégration, avec les systèmes déjà en production, est également un facteur favorisant la réussite de l'implantation du nouveau système (Rahimi et al., 2009). A ce titre, il sera possible d'utiliser les systèmes déjà utilisés dans l'environnement cible d'un futur déploiement ainsi que de s'appuyer sur des technologies « clefs en main » afin d'assurer rapidement et de manière fiable certaines fonctionnalités du nouveau système. Parmi les technologies existantes, on peut signaler que les solutions open sources peuvent d'ailleurs présenter un intérêt non négligeable en terme d'efficacité et de coût (Sayyad Shirabad et al., 2012). Par exemple, dans le domaine du suivi oculaire, on pourrait relever l'existence du projet openEyes (Babcock & Pelz, 2004; Li et al., 2006, 2005) ou encore de l'ITU Gaze Tracker (San Agustin Lopez, 2009; San Agustin Lopez et al., 2010), qui mettent à disposition des plans de montage et/ou des algorithmes relativement aboutis.

Pour conclure, il faudra prévoir que le système obtienne les agréments délivrés par des autorités compétentes et nécessaires à son utilisation en contexte réel. De tels agréments devront pouvoir être obtenus en évaluant seulement les composants un à un. En effet, pour des raisons pratiques, il n'est pas possible d'évaluer toutes les combinaisons possibles d'applications et de matériels rendues possibles par l'utilisation d'une PAM. Pour gérer les aspects sécuritaires d'un tel système, il faudrait alors procéder à l'agrément de la plateforme, puis pour chaque composant vérifier qu'il se conforme bien aux exigences techniques ainsi agréées (Vasserman & Hatcliff, 2014). D'ailleurs, Vasserman et Hatcliff (2014) estiment que les éléments, garantis par la maîtrise de la sécurité et de la sûreté du dispositif, sont :

- La non-mise en danger des patients même si les données sont trafiquées.
- Le contrôle de l'accès aux données des patients.
- La certitude que seuls des éléments agréés peuvent se connecter au système.
- La traçabilité des incidents.

Ainsi, la PAMMA devra bénéficier d'une telle homologation de la même manière que le devront les applications destinées à y être exécutées.

B) La conception d'applications pour la PAMMA

Bien que nous ayons précédemment évoqué succinctement le développement d'applications sur la PAMMA (voir la sous-section **Le processus de développement d'applications**), il nous

paraît nécessaire de démystifier cette tâche complexe. Assurément, pour de tels développements, il y a un grand nombre d'exigences à prendre en considération, de nombreux acteurs/utilisateurs et points de vue entre lesquels il faut savoir jongler et des processus de développement hautement collaboratifs et itératifs à mettre en œuvre. Dans cette section, nous allons nous efforcer de dresser un panorama de ces différents éléments.

Construire un réseau de communication interopérable, grâce à l'intégration de SI, implique des changements dans l'organisation des pratiques de soins et la façon dont les gens utilisent le système (Protti, 2009). Dans le cas de la PAMMA, ces changements seront provoqués par l'introduction d'applications se présentant comme de nouveaux outils mis à la disposition des professionnels de santé. Avec ce type d'applications, dans le cas d'un SI à visée médicale, on peut ainsi s'attendre à rencontrer certaines contraintes :

- Le système doit être adapté à la qualité et à la quantité de données transmises, en mode client-serveur, en fonction de la bande passante disponible (Porumb et al., 2010).
- Le système ne doit pas détourner l'attention que le praticien porte au patient, afin d'éviter les lacunes de communication, menant à des problèmes de suivi du traitement, entre ces derniers (Frankel et al., 2005; Weibel et al., 2013).
- Le système doit réellement faciliter le travail, c'est-à-dire qu'il doit s'intégrer parfaitement dans les pratiques des praticiens et offrir des interactions multimodales adaptées à de bons accès et usages de l'information. Dans le cas contraire, les professionnels de santé se débrouilleront pour trouver des moyens de contournements qui peuvent mener à sacrifier l'intérêt premier qu'est supposé avoir le système informatique (Chen, 2010; Heath & Luff, 1996).

Il faut également être précautionneux lorsqu'on souhaite mettre en œuvre un système informatique supportant différents groupes d'utilisateurs nourrissant chacun des buts différents (Ellingsen & Røed, 2010). Effectivement, la mise en production de ce genre de développement se heurte classiquement à trois problèmes majeurs (Star & Ruhleder, 1994) :

- L'implémentation et le déploiement.
- Les effets de bord du déploiement.
- L'acceptation du système par les utilisateurs.

Alors que ces deux premiers problèmes peuvent être réglés par l'enrôlement dans le projet de plus de ressources, le troisième est particulièrement complexe à gérer et est, malheureusement, rarement pris en compte (Cortés et al., 2008). En effet, il peut être délicat de faire accepter à des utilisateurs, qui se sentent déjà à l'aise dans leurs activités professionnelles avec les outils et routines qu'ils utilisent couramment, de travailler selon une nouvelle méthode (Greenes, 2014; Isern, 2008; Morley et al., 2011; Wangler et al., 2003). Ainsi, dans l'optique d'éviter un boycott du système par ses propres utilisateurs, il est primordial d'impliquer tout au long du processus de développement tous les groupes d'utilisateurs afin qu'ils soient parties prenantes du projet (Rahimi et al., 2009). Ceci va permettre entre autres d'arbitrer les éventuels conflits d'intérêt qu'il pourrait y avoir entre eux le plus tôt possible, de gérer les tensions entre stabilité et changement et faire de l'intégration un accomplissement collectif (Ellingsen & Røed, 2010). En outre, les processus de travail retenus devraient être assez flexibles pour permettre leur adaptation à l'échelle individuelle. Effectivement, les cliniciens peuvent avoir des appréhensions concernant l'utilisabilité, la compatibilité et la fiabilité du système ce, qui peut les inciter à ne pas

s'investir dans son acquisition (Bates & Gawande, 2003; Lakbala & Dindarloo, 2014; Mechanic, 2008). Il nous faut donc lever ces appréhensions en impliquant ces cliniciens dans un processus de développement centré-utilisateur afin qu'ils participent à la conception du système et qu'ils aient confiance dans ce dernier (Beaudouin-Lafon & Mackay, 2007; Sánchez et al., 2010). La confiance est assurément un point clef de l'acceptation d'un système médical par ses utilisateurs (Ferreira et al., 2009). Il faut aussi informer et former les futurs utilisateurs afin qu'ils aient les compétences requises pour utiliser correctement le système, et qu'ils comprennent à quel point son utilisation influe positivement sur leurs pratiques (Blatt et al., 2012; Levick & Osheroff, 2014; Rahimi et al., 2009). En effet, donner un accès aux informations relatives à un projet, ainsi que permettre à ses acteurs de communiquer à son propos, est une nécessité pour son bon déroulement (Bharadwaj & Reddy, 2003). L'acceptation et l'utilisation du système par les praticiens est nécessaire pour toute organisation de soins souhaitant un retour positif sur ses investissements en termes d'infrastructure informatique (Ilie et al., 2007). On notera aussi que la résistance des praticiens est une des principales raisons d'échec lors de la mise en place de tels systèmes en dépit de leurs avantages potentiels (Lakbala & Dindarloo, 2014). A contrario, les systèmes pour lesquels leur participation à la conception a été active semblent donner de bons résultats (Garg et al., 2005).

Un des aspects les plus sensibles pour l'acceptation des applications que nous voulons pouvoir développer sur la PAMMA concerne les modifications qui peuvent être produites au niveau de la relation praticien-patient. Une telle relation est nécessairement transformée lorsqu'un système se place comme un médiateur à la communication et/ou l'interaction entre humains. Dans notre contexte applicatif, cela est particulièrement flagrant dans le cas des applications dites de télémédecine (Wootton, 2001). Effectivement, la mise en place de solutions de télémédecine peut se heurter à plusieurs types de problèmes (Mandellos et al., 2009) :

- Juridiques, c'est-à-dire qu'il faut pouvoir établir à qui revient la responsabilité d'erreurs, médicales ou liées à une défaillance du matériel, survenues en utilisation du système. Il y a d'ailleurs fréquemment des vides juridiques entourant les pratiques de télémédecine.
- Financiers, c'est-à-dire que les équipements, le personnel mobilisé et l'utilisation de réseaux de télécommunication a un coût qu'il faut pouvoir évaluer afin de déterminer une facturation pour un tel service médical. Les actes de télémédecine ont d'ailleurs besoin d'être reconnus en tant qu'actes médicaux afin de pouvoir faire l'objet de remboursements par les assurances maladies.
- Technologiques, c'est-à-dire qu'il faut être capable de régler les soucis d'interopérabilité, d'adaptabilité, de respect des standards de qualité et de sécurité ou encore de disponibilité des réseaux de télécommunication. Ces différents problèmes techniques peuvent rendre l'utilisation d'un système de télémédecine inefficace ou simplement impossible.

Bien que nous ayons présenté tout au long de ce document des éléments permettant de résoudre des problèmes d'ordre technologique, la résolution des autres types de problèmes est complètement hors de notre spectre d'influence. Il faut donc espérer que les recherches sur la télémédecine permettront une reconnaissance de ce type d'actes ainsi que l'établissement d'un contexte légal clairement défini (Nealon & Moreno, 2003), levant ainsi toute ambiguïté en ce qui concerne l'efficacité de tels systèmes et la détermination des

responsabilités en cas d'incidents consécutifs à leur utilisation. D'un point de vue opérationnel, on pourra remarquer qu'il faut prendre en compte, pour maximiser les chances de succès d'un système de télémédecine, l'ensemble des considérations suivantes (Yellowlees, 2005) :

- Les sites et applications de télémédecine doivent être sélectionnés pragmatiquement et non philosophiquement, c'est-à-dire qu'il faut trouver avant tout les sites ayant les compétences et de l'intérêt pour les applications de télémédecine.
- Les cliniciens doivent conduire le projet et les autres types d'utilisateurs s'approprier le système. Effectivement, le meilleur moyen d'obtenir l'implication des praticiens est que ces derniers assimilent le projet comme étant le leur pour qu'ils en soient fiers et le promeuvent.
- La direction et le support de l'application de télémédecine devraient suivre de bonnes pratiques de travail en entreprise. En particulier, les cliniciens devraient pouvoir eux-mêmes assurer la gestion du projet de manière à que ce dernier ne perde pas de vue le contexte clinique pour lequel il est développé.
- La technologie devrait être aussi facile à utiliser que possible, c'est-à-dire que les applications de télémédecine devraient pouvoir s'intégrer dans un environnement classique de travail et s'interfacer avec d'autres applications de façon à être simples d'usage.
- Les utilisateurs de la télémédecine doivent être bien formés et assistés, aussi bien techniquement que fonctionnellement. Assurément, des séances d'entraînement doivent être mises en place avant et après l'introduction du système, à travers une sensibilisation et de réelles formations, de façon à ce que les cliniciens voient le dispositif comme faisant partie de leur environnement normal de travail.
- Les applications de télémédecine devraient être évaluées et soutenues d'une manière économiquement et cliniquement adaptée ainsi que facile à mettre en place. Effectivement, il faut prévoir d'évaluer l'efficacité de la solution au travers d'un business plan permettant d'anticiper le coût des équipements nécessaires, du personnel et du support technique mais également les retombées d'un point de vue clinique.
- L'information à propos du développement devrait être partagée. Pour que la télémédecine s'installe durablement dans les pratiques, il faut qu'elle puisse faire l'objet d'évaluations basées sur les expériences passées, qu'elles soient positives ou négatives. A ce titre, des retours d'expériences cliniques devront être formulés pour chaque application mise en œuvre.

Toutes ces considérations relèvent d'une approche de pilotage de projet qui s'attache à garder au premier plan l'intérêt fonctionnel que représente l'intégration du système informatique. On pourra également signaler qu'en France la **Haute Autorité de Santé** (Haute Autorité de Santé, 2015) propose une grille de pilotage pour accompagner le déploiement des applications de télémédecine (Haute Autorité de Santé, 2013a).

Nous venons de présenter les exigences des professionnels de santé vis-à-vis des systèmes de télémédecine, mais il ne faut pas oublier les autres utilisateurs très importants du système, à savoir les patients. Dans le cas de la mise en place de solutions de télémédecine, on remarque que la satisfaction des patients est généralement positive (Allen & Hayes, 1994). Ce type d'interventions permet, en particulier, d'augmenter l'offre de services médicaux/paramédicaux dans les « déserts de soins » et d'éviter ainsi aux patients le

fardeau que peuvent être les trajets à destination d'établissements de santé (Wootton, 2001). Dans le contexte général de l'intégration des processus de soins, qui dans notre cas va bénéficier d'une implémentation informatique, le ressenti des patients est malheureusement rarement étudié et, bien qu'une telle pratique semble donner de bons résultats, il n'est pas évident de déterminer quelles sont les véritables attentes des malades vis-à-vis de ce type d'approche (Spragins & Lorenzetti, 2008). L'implication des patients dans le continuum de soins mériterait donc de faire l'objet de plus amples investigations. Pour finir, on pourra noter qu'il est évidemment critique de garantir la confidentialité des données des patients, pour éviter l'échec de la mise en œuvre d'un tel système (Protti, 2009).

Pour être adopté en milieu clinique, notre système va devoir être « user-friendly » et permettre une utilisation « plug-and-play » du matériel qu'il supportera. Les performances du système sont aussi un élément clef de son acceptation par les praticiens. En effet, un système ne mettant pas tout en œuvre pour atteindre un haut niveau de performances risque d'être délaissé (Arney et al., 2014). A la vue de ces considérations une version aboutie du système devra disposer d'une brique logicielle capable d'introspecter la solution afin de la renseigner sur les matériels qui lui sont connectés. Cette brique devra, à cette même occasion, déterminer les caractéristiques opérationnelles de ces matériels.

Pour conclure, rappelons qu'il est primordial de s'assurer que notre solution, basée sur des agents distribués, est commercialement viable (comme, par exemple, cela a été étudié dans le contexte du commerce en ligne (Mahmoud & Yu, 2005)). En effet, il est nécessaire qu'un modèle économique équitable régie les échanges de services entre différentes PAMMA appartenant à des institutions différentes.

Après ces nombreuses considérations d'ordre général, nous souhaitons approfondir deux points propres au développement d'applications médicales permis par notre approche :

- La réalisation d'applications via la modélisation des processus métiers.
- La réalisation d'applications pour les enfants dyslexique basées sur le jeu.

1) La modélisation des processus métiers

Dans le contexte de notre travail, s'intéresser à la modélisation des processus métiers en milieu clinique c'est étudier le processus de développement d'applications pour la PAMMA. Or le développement des modèles de workflow a besoin, comme d'autres processus de développement (voir le chapitre **Le choix d'un modèle de développement**), de suivre un processus cyclique (Hollingsworth, 2004) qui débute par l'analyse des besoins des utilisateurs. L'identification et la modélisation des processus doit ainsi s'effectuer de manière collégiale et itérative avec les différents utilisateurs du futur système (Gooch & Roudsari, 2011). Ce travail va requérir, en plus des développeurs, l'adhésion des futurs utilisateurs, à savoir des praticiens, des patients et des chercheurs. Il paraît donc indispensable de monter des collaborations avec différents regroupements professionnels pour procéder, dans un cadre formel, à la matérialisation et maturation du système. En outre, afin que le système puisse être utilisé en pratique, il est important de s'assurer que les utilisateurs ont un certain niveau de maîtrise de ce dernier, ce qui est rarement le cas (Breen & Matusitz, 2010). A ce titre, il faudra prévoir de les former. On souhaite ainsi

efficacement pouvoir prendre en compte leurs différentes approches et points de vue (Framinan et al., 2005). Il peut d'ailleurs être pertinent de pouvoir visualiser, au travers de différentes représentations des workflows, ces différents points de vue de manière exclusive afin d'en maximiser la compréhension par ces acteurs (Caetano, Pereira, & Sousa, 2012). D'autre part, chaque organisation a ses propres routines, terminologies et outils qui complexifient d'autant l'intégration globale des processus (Wangler et al., 2003). Morley et al. (2011) proposent une démarche générale en trois étapes pour recueillir et traduire ces éléments informatiquement :

- L'identification des processus, en réaction à des événements déclencheurs internes ou externes à l'organisation.
- La modélisation des processus, sous la forme de chaînes chronologiques d'activités visant à produire les sorties attendues.
- La mise en correspondance entre activités humaines et éléments informatiques.

La première étape revient à recueillir précisément le besoin des utilisateurs, ce qui est de la plus haute importance afin de fournir, après développement, un produit adéquat (Neill & Laplante, 2003). Pour cela, il est prépondérant d'aller observer les utilisateurs sur leur lieu de travail et de réaliser des entretiens, qui seront un minimum structurés (Dieste et al., 2008), visant à mieux comprendre comment ils travaillent, comment ils procèdent et ce qu'ils utilisent à cette fin (Anderson et al., 2001). Les deux suivantes, quant à elles, vont principalement s'appuyer sur l'éditeur de workflows, dont nous avons précédemment discuté (voir la section **L'intégration des processus métiers**). En effet, ce dernier va, dans un premier temps, permettre la modélisation des processus de travail (Niveau 3 de l'éditeur) puis, dans un second temps, offrir l'assistance à leur mise en correspondance avec des agents (Niveau 2 de l'éditeur). Les implémentations ainsi réalisées doivent ensuite faire l'objet d'une validation/diagnostic. Pour cela, on peut enrichir la démarche précédemment proposée et mettre en place un cycle constitué des étapes suivantes (**Figure 214**) :

- La modélisation des processus, c'est-à-dire l'informatisation des processus existants sous la forme de workflows.
- La configuration du système, correspondant à l'enregistrement dans le système de gestion des ressources impliquées dans les processus.
- L'adoption des processus, à savoir le déploiement des workflows.
- Le diagnostic, c'est-à-dire l'amélioration des processus basée sur l'analyse et l'observation du fonctionnement des workflows.

Le caractère itératif de processus est propice à la réalisation de prototypes permettant aux utilisateurs d'expérimenter les futures applications avant leur finalisation et leur mise en production (Beaudouin-Lafon & Mackay, 2007). D'autre part, cette représentation a le mérite d'être simple et pertinente (Ko, 2009). On pourrait même la qualifier de « process pattern » (Ambler, 1998). Une telle approche a par ailleurs déjà été envisagée dans le cadre du développement de SMA (Occello, 2003). L'utilisation de notre éditeur vise d'ailleurs à assister et à systématiser ce type de démarche.

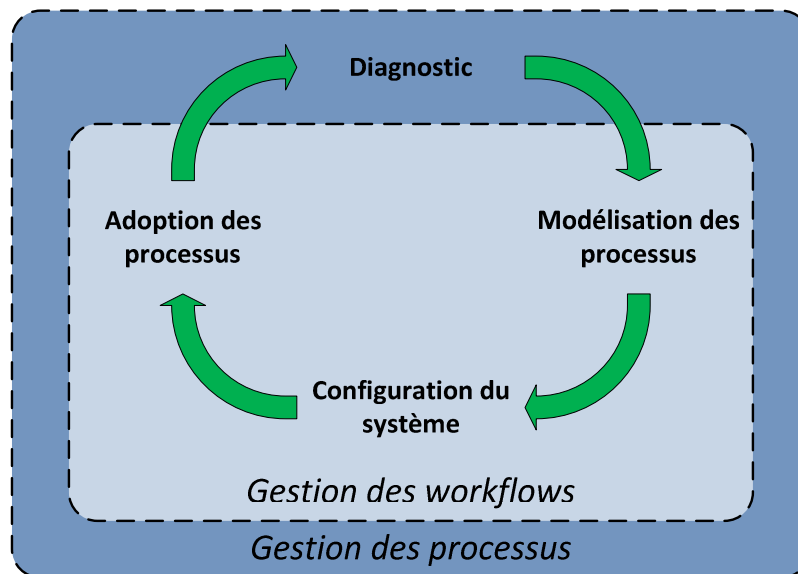


Figure 214 : Schéma de la conception d'une gestion de processus métiers (d'après (van der Aalst, ter Hofstede, & Weske, 2003))

On peut retrouver au sein d'une organisation médicale une implémentation des processus de soins relativement similaire (**Figure 215**) et qui s'appuie idéalement sur des directives/recommandations cliniques édictées par des instances centrales (Isern, 2008)(telles que la HAS en France (Haute Autorité de Santé, 2015) qui propose des recommandations de bonnes pratiques (Haute Autorité de Santé, 2014) et des protocoles nationaux de diagnostic et de soins (Haute Autorité de Santé, 2013b), ou encore le GIN à un niveau international (Guidelines International Network, 2014)). En particulier, l'idée générale est de pouvoir adapter des directives/recommandations d'ordre général au contexte local de l'organisation de santé souhaitant intégrer informatiquement son activité (Boxwala et al., 2001; Fridsma, Gennari, & Musen, 1996)(pour un exemple d'application, consulter (Gillois et al., 2001)). Cette manière de procéder se retrouve d'ailleurs dans la description de l'utilisation du formalisme PROforma (Fox et al., 1998). Il existe des moyens pour convertir systématiquement les descriptions textuelles des directives/recommandations cliniques en structures plus formelles, voire en algorithmes (Gerber, Brown, & Harrison, 2012; Lobach & Kerner, 2000)(GMT (Votruba et al., 2004) est un exemple de tel outil). Néanmoins, il est important de pouvoir identifier les intentions, sous-tendues dans des descriptions narratives de processus métiers, afin de correctement les intégrer aux applications à développer (Yu, 1995) et un traitement automatique n'est pas nécessairement capable d'un tel discernement. Une intervention manuelle reste donc indispensable à la modélisation des directives/recommandations cliniques sous la forme de workflows. Cette formalisation passe idéalement par une étroite collaboration entre des experts métiers – des praticiens expérimentés – et des experts techniques – des ingénieurs et des analystes fonctionnels, dont l'objectif est de faire la jonction entre la technique et l'expertise métier (Huser et al., 2011; Moser et al., 2001; Peleg & González-Ferrer, 2014; Staccini et al., 2000). Récupérer les différents points de vue des futurs utilisateurs est une démarche indispensable à la modélisation des processus métiers (Framinan et al., 2005) qui est systématiquement mise en œuvre lors du développement de SI similaires au nôtre (Ellingsen & Røed, 2010). Une telle attribution des rôles peut également se retrouver dans un cadre méthodologique de conception de simulations informatiques (Meurisse, 2004). On signalera que la nature

graphique de la représentation des processus métiers ainsi que la possibilité de les hiérarchiser contribuent fortement à l'efficacité de telles collaborations. En outre, afin d'être bien compris des cliniciens, il pourra s'avérer pratique d'utiliser le terme « scénario » pour désigner un workflow (Huser et al., 2011).

Une des perspectives de notre PAMMA est qu'à terme nos utilisateurs experts – les cliniciens et les chercheurs – puissent publier leurs *Type Processus* ou *Type Epreuve* via une plateforme en ligne (pour la signification de ces notations, voir la sous-section **L'élaboration du diagramme de classe métier**). En effet, chacun de ces processus peut être vu comme un véritable logiciel à visée médicale. Il conviendra donc de mettre en place un système de régulation, permettant de garantir qu'elles sont sûres d'usage et présentent un intérêt d'ordre médical, avant d'en autoriser la diffusion (Hatcliff et al., 2012). Naturellement, seuls les *Type Processus* n'étant pas fortement liés à la coordination de nombreuses personnes au sein d'une organisation spécifique, c'est-à-dire n'étant pas des processus de travail relevant d'une intégration horizontale complexe, seront susceptibles d'être proposés à la publication. Effectivement, seuls les processus dénotant d'une intégration verticale ou d'une intégration horizontale simple, et donc étant indépendants de l'organisation au sein d'une structure de soins particulière, peuvent véritablement être transposables.

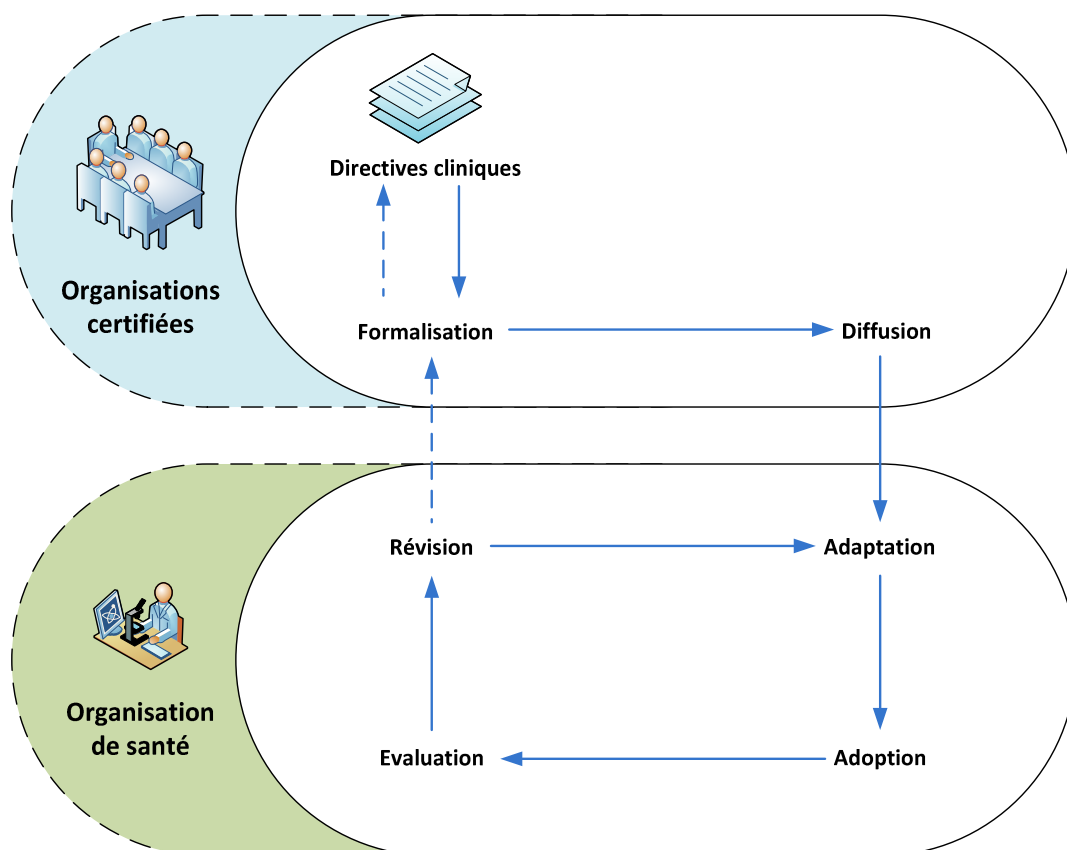


Figure 215 : Schématisation du processus d'informatisation des processus cliniques (d'après (Cicarese et al., 2005))

On peut noter que, conformément à notre aspiration de vouloir supporter le travail collaboratif (voir la section **Le principe général du système**), notre système doit pouvoir intégrer aussi bien les processus opérationnels que ceux organisationnels (Rahimi et al.,

2009). D'un point de vue technique, il n'y pas de réelle différence entre ces différents types de processus et la méthode mise en œuvre pour les modéliser pourront donc être relativement similaire. D'un point de vue fonctionnel, la standardisation des processus des soins, en particulier s'ils impliquant des équipes pluridisciplinaires, est dans l'intérêt premier du patient car elle favorise des soins de meilleure qualité et écarte les risques d'erreurs médicales (Powell & Hohenhaus, 2006). En particulier, les experts médicaux de ces processus doivent pouvoir garantir un très haut niveau de sûreté des workflows implémentés par le système (Fox, 2002). Pour mener à bien cette étape de modélisation, Framinan et al. (2005) proposent un découpage en plusieurs équipes de travail en charge chacune de certaines activités du processus de modélisation. Cette répartition des tâches semble avoir bien fonctionné pour eux, et peut constituer un point de départ pour la mise en place de notre propre processus de modélisation de processus métiers.

Comme pour la télémédecine, pour implémenter un système modélisant les processus métiers du milieu médical et les intégrer grâce à l'informatique, plusieurs recommandations sont nécessaires afin que le changement se passe dans de bonnes conditions. Effectivement, il faudra (Collins et al., 2010) :

- Rassurer les professionnels de santé quant aux bénéfices à utiliser ce système.
- Former les utilisateurs et leur fournir une assistance technique.
- Définir un planning réaliste pour l'implémentation et la formation des personnels de santé ainsi que ceux en charge des infrastructures informatiques.
- Partager les informations relatives aux performances de l'implémentation (positives ou négatives).

Pour cela un médecin « champion » doit être désigné pour porter ces changements et faire le lien entre l'équipe du milieu clinique et celle du milieu des nouvelles technologies (Levick & Osheroff, 2014).

Il faut également être conscient que l'intégration présente des risques. En effet, l'intégration est là pour structurer/systématiser les processus de travail et il y a un danger de rigidifier l'organisation ainsi intégrée, ce qui peut se traduire par des effets inverses à ceux escomptés. Par exemple, une intégration mal réalisée peut engendrer une perte d'efficacité, une augmentation des coûts, voire la banqueroute (Walston et al., 1996). En particulier, l'intégration verticale peut vraiment être désastreuse si elle conduit à étendre les responsabilités des gestionnaires à des domaines pour lesquels ils n'ont pas l'expertise (Hernandez, 2000). C'est d'ailleurs pour éviter cela que, dans notre modélisation, nous avons choisi de restreindre ce type d'intégration à des corps de métiers et/ou départements, constitués de professionnels partageant une même expertise, plutôt qu'à l'intégralité d'un établissement de santé par exemple (voir le chapitre **L'application de l'UP7: un modèle générique de prise en charge de la dyslexie**). De la sorte, avant l'exécution en « conditions réelles » des workflows, il pourra être pertinent de réaliser des analyses d'exactitude (Lu et al., 2006; van der Aalst, 2000) et/ou des simulations afin d'en vérifier les comportements attendus (comme, par exemple, ceci est effectué dans (Quaglini et al., 2000)). Pour réaliser des simulations, il faut tester l'efficacité des workflows sur d'anciennes données de patients et comparer les résultats de la procédure avec ceux obtenus précédemment. Il devient également possible d'analyser la charge de travail des différentes ressources du système (humaine et/ou matériel) et d'identifier des goulots d'étranglement ainsi que d'évaluer les coûts et les temps de traitements des activités effectuées.

Pour finir, nous rappelons que notre système informatique, d'une manière similaire à n'importe quel autre SI, est uniquement un outil/moyen pouvant permettre l'amélioration des processus métiers (Rahimi et al., 2009). L'amélioration réelle des pratiques dépend de l'usage qui en est fait et non de sa seule mise en œuvre. En outre, il est important d'avoir un grand nombre d'utilisateurs afin de provoquer un phénomène d'auto-renforcement du système, c'est-à-dire que l'adoption du système conduit à son amélioration en continu. A ce sujet, Hanseth et Lyytinen (2004) proposent un ensemble de principes que nous pourrions suivre pour implémenter et mettre en place des applications sur la PAMMA.

2) L'approche ludique et immersive

D'une manière générale, on remarque que les humains apprécient les jeux car ils aiment résoudre des problèmes (Schell, 2010). Les enfants et adolescents ne font pas exception à cette règle et sont attirés par les jeux vidéo pour de multiples raisons (Olson, 2010). Globalement, ils sont divertissants, presque par définition, et sont attrayants parce qu'ils répondent aux besoins psychologiques des jeunes et les aident à aborder leur développement (Durkin, 2010). On remarquera aussi que dans les jeux vidéo l'impact de l'échec est très différent de celui vécu dans le milieu scolaire. En effet, dans le contexte du jeu, le prix de l'échec est amoindri car il est possible de recommencer indéfiniment. Ainsi l'échec est souvent considéré comme un moyen d'apprendre les mécaniques du jeu et éventuellement de gagner. Il est donc pertinent de dire que l'échec dans les jeux vidéo stimule l'apprentissage (Gee, 2008). D'autre part, l'interactivité présente dans les jeux vidéo est importante dans le processus d'apprentissage. Effectivement, tout apprentissage profond implique que les apprenants ressentent un fort sentiment d'appartenance autant que la capacité de construire et de ne pas juste consommer passivement (Gee, 2008). Ainsi, le jeu vidéo fait appel à diverses capacités, y compris sensorielles, cognitives et sociales (Durkin, 2010). Les jeux éducatifs d'apprentissage comprennent généralement tous les éléments des jeux traditionnels tels que des règles, des défis, de l'interactivité et la créativité (Ohene-Djan & Begum, 2008). On remarquera aussi que tous les jeux disposent de contenu : des faits, des compétences et des procédures que les joueurs doivent maîtriser (Gee, 2008). En particulier, dans nos programmes ludiques de remédiation ce contenu sera spécifique aux processus cognitifs déficitaires que nous voulons faire travailler chez ces enfants. Les enfants à qui s'adressent nos programmes de remédiation étant pour la plupart déficitaires dans ces capacités (INSERM (dir.), 2007), une remédiation basée sur l'utilisation de jeux vidéos va donc pouvoir s'avérer pertinente. En plus de ceux que nous avons déjà évoqués (voir la sous-section **Les outils de diagnostic et de remédiation**), il a déjà été tenté de développer des jeux éducatifs pour les enfants souffrant de troubles spécifiques des apprentissages tels que, par exemple, un jeu d'entraînement de la mémoire de travail développé par Al-Wabil et al. (2010). Cette dernière expérience, où la reconnaissance d'images est mise à l'épreuve, nous apprend plusieurs choses à respecter lors de la conception d'un tel jeu :

- Éviter de présenter simultanément à l'enfant des images de même catégorie sémantique.
- Utiliser des objets familiers que l'enfant a l'habitude de voir dans son environnement.

D'autre part, on pourra aussi noter qu'en cas de présence, au sein du jeu, d'un narrateur ou personnage principal il faudra tenir compte du sexe de l'enfant pour lui proposer un personnage l'incitant à un engagement plus grand vis-à-vis de l'environnement proposé : on choisira de préférence un personnage de même sexe que celui de l'enfant ou un personnage de genre neutre (Al-Wabil et al., 2010). Effectivement, les jeux vidéo peuvent créer une empathie permettant au joueur de se projeter dans l'univers virtuel qui lui est proposé, ce qui est d'ailleurs particulièrement propice à la compréhension et l'appropriation de connaissances (Gee, 2008). Il est donc important de favoriser la mise en place de cette projection des enfants dans le jeu en choisissant de personnages auxquels ils s'identifieront facilement. Ces bonnes pratiques peuvent d'ailleurs être étendues à l'iconographie, la symbolique, les couleurs et les animations choisies pour les menus et interfaces du jeu. Ceci vise à les rendre ainsi facilement accessibles par l'enfant dès la prise en main de l'application. De la même façon, il faut éviter de proposer des interfaces trop complexes et minimiser au possible la présence de données textuelles afin d'éviter toute déstabilisation ou surcharge cognitive chez l'utilisateur.

Ces considérations, qui relèvent du « game design » (**Figure 216**), permettent d'augmenter l'engagement de l'enfant vis-à-vis des jeux, les rendant plus amusants et intéressants, ce qui contribue à l'efficacité de l'apprentissage (Ohene-Djan & Begum, 2008). Selon Schell (2010), « le game design est l'art de décider de ce qu'un jeu devrait être ». Le game design consiste ainsi en une série de prises de décision tout au long de la création d'un jeu. Le jeu en lui-même a pour but de faire vivre une expérience au joueur : c'est ce qui doit véritablement importer à son concepteur, le game designer. En particulier, afin d'assurer la satisfaction de nos utilisateurs, notre remédiation ludique va devoir simultanément ne pas frustrer et distraire (Allen et al., 2003). Il va donc falloir apporter un soin particulier au game design de nos futures applications afin de maximiser le critère de satisfaction de nos jeunes utilisateurs. Par ailleurs, nous savons que le game design va être un facteur important de l'efficacité de la remédiation. Assurément, il est reconnu qu'un bon game design accroît l'efficacité d'apprentissage des jeux éducatifs (Gee, 2008). Par exemple, les progrès de l'enfant peuvent lui être signifiés en quasi temps réel via des animations, contrairement au cas d'une méthode qualifiée de « papier-stylo ». Ce type de feedback sert à maintenir la motivation et l'engagement de l'enfant dans le processus d'apprentissage ainsi engagé (Ohene-Djan & Begum, 2008). Un des impératifs, pour le développement futur d'applications pour la PAMMA, sera de se pencher avec minutie sur les éléments de game design (**Figure 216**) pouvant être mises en œuvre dans le contexte de la remédiation d'enfants souffrant de dyslexie.

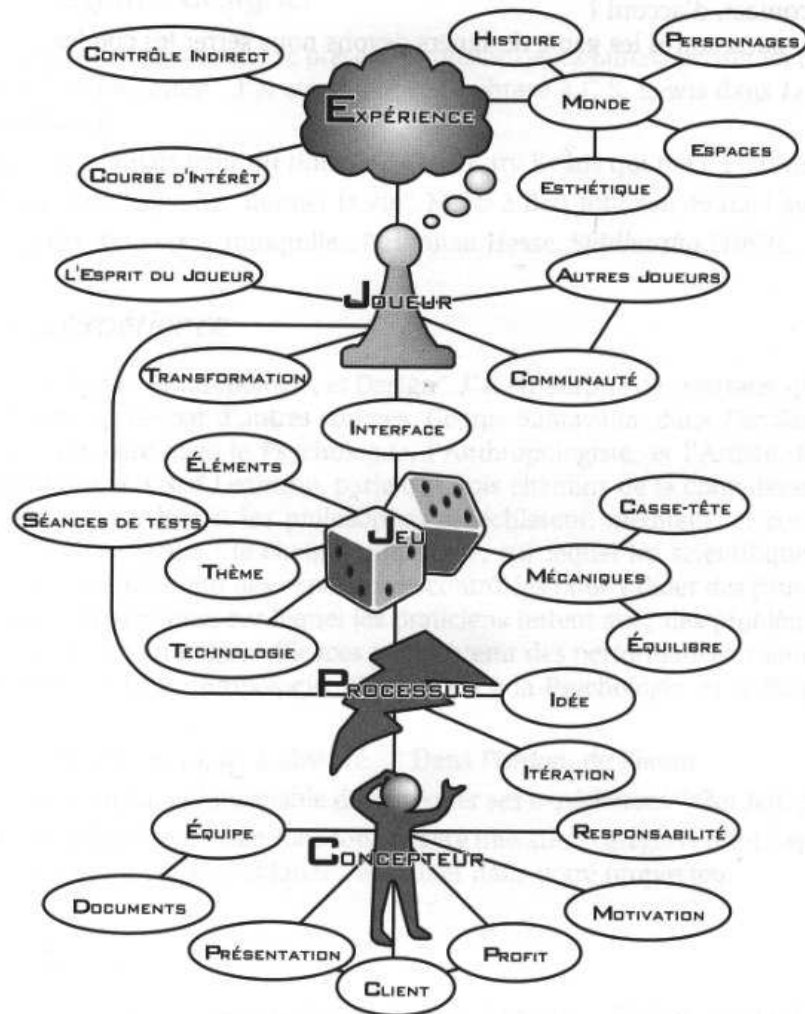


Figure 216 : « Carte du game design » de J. Schell (Schell, 2010)

Dans un « jeu sérieux », c'est-à-dire dans un jeu avec des perspectives pédagogiques et/ou informatives (pour une recherche approfondie sur ce type d'applications et le processus de game design qui leur est associé, consulter (Djaouti, 2011)), l'aspect social dans lequel le jeu se situe, qui pourrait paraître secondaire au premier abord, doit être aussi soigné que le game design (Gee, 2008). En effet, la mise en place de telles possibilités d'échange autour du jeu va faciliter l'apprentissage et donc, dans notre cas, va accentuer l'effet de la remédiation. On peut aussi remarquer que les émotions jouent un rôle étonnamment important dans l'attrait des enfants pour les jeux vidéo, en particulier chez les garçons. Effectivement, des recherches (Gee, 2008) ont montré le rôle des émotions dans l'apprentissage, et suggère que les jeux vidéo aident à motiver l'apprentissage en reliant les émotions à la résolution de problèmes. Dans un sondage, réalisé par Olson en 2007 auprès de 1254 enfants entre 12 et 14 ans, 62% des garçons et 44% des filles qui ont joué à des jeux électroniques étaient de peu à fortement d'accord sur le fait qu'ils utilisent parfois ces jeux pour s'aider à se détendre. Un nombre important également utilisent des jeux pour faire face à la colère, pour oublier leurs problèmes ou encore pour affronter la solitude (Olson, 2010). Ce constat nous conforte dans l'idée qu'une remédiation vidéo ludique va intrinsèquement induire une sensation de bien-être chez les jeunes patients. Ces derniers vont ainsi être dans un état

d'esprit propice à l'engagement dans la thérapie qui leur est proposée. Comme le dit Schell (2010), travailler et jouer sont, respectivement, équivalents à la servitude et à la liberté. La différence entre une activité ludique et une activité laborieuse n'est pas liée à l'activité en elle-même mais à la motivation de la personne qui l'exerce. Ainsi, moins on se sent obligé de réaliser un travail, plus il s'apparente à un jeu.

La multimodalité de ces médias peut également présenter un intérêt pédagogique majeur. Un enseignement multi-sensoriel/multimodal se compose d'une série de stratégies multi-sensorielles qui incluent des techniques reliant les yeux, les oreilles, la voix et les mouvements de main à l'apprentissage symbolique. L'approche adoptée consiste à essayer d'engager autant de récepteurs sensoriels dans le processus d'apprentissage que possible. En effet, il semblerait que les enfants ayant des difficultés d'apprentissage bénéficient particulièrement de cette approche dans la mesure où elle favorise leur mémorisation lors de l'apprentissage (Ohene-Djan & Begum, 2008). Ainsi, avec ces enfants, il va être judicieux d'utiliser les technologies d'assistance qui contribuent à l'apprentissage multi-sensoriel (Hecker & Engstrom, 2005). On a d'ailleurs pu observer la mise en place de tels outils en milieu scolaire et universitaire (Mejía Corredor & Fabregat Gesa, 2012). Les systèmes, au moins en partie tels que celui que nous voulons développer, sont connus comme des technologies d'assistance pour la rééducation cognitive ou comme prothèses cognitives. Il s'agit d'interventions technologiques destinées à améliorer les capacités fonctionnelles des personnes ayant une déficience cognitive (Sampath et al., 2010a).

Très certainement, la technologie informatique ultime permettant d'allier une approche ludique et une interaction multimodale riche est la réalité virtuelle. Effectivement, les systèmes utilisant cette technique sont de plus en plus reconnus comme un moyen utile pour l'étude, l'évaluation, et la remédiation des processus cognitifs et des capacités fonctionnelles (Klinger, 2006; Rizzo et al., 2000). De tels systèmes de représentation devraient permettre à l'observateur d'interagir « naturellement » avec des objets et d'autres personnes au sein d'un environnement simulé (ou « monde »), afin de lui faire vivre une expérience impossible à distinguer de la « réalité normale » (Loomis, Blascovich, & Beall, 1999). Bien qu'encore loin de cet idéal, de nombreux environnements virtuels ont déjà fait leur apparition. Ces derniers utilisent des transducteurs numériques pour stimuler les organes sensoriels humains afin de donner une illusion de « réalité normale » au cerveau de l'observateur. Un environnement virtuel immersif est, de la sorte, un environnement dans lequel l'utilisateur est entouré d'un point de vue perceptif par un environnement virtuel (Loomis et al., 1999). Le degré d'immersion peut, par ailleurs, varier selon les dispositifs utilisés pour sous-tendre les interactions avec l'environnement virtuel. Il existe deux types fréquents d'implémentation de tels dispositifs :

- Le CAVE (Cruz-Neira, Sandin, & DeFanti, 1993) où les images sont projetées sur les parois d'une pièce dans laquelle l'observateur évolue.
- Le HMD (Zhang, 2007) qui mesure les changements de position et l'orientation de la tête de l'utilisateur afin de produire une représentation 3D de l'environnement simulé devant les yeux de l'observateur.

De tels systèmes sont précieux pour les recherches sur la perception visuelle, la cognition spatiale et les interactions sociales (Blascovich et al., 2002; Loomis et al., 1999; Rizzo et al., 2000). Toutes les manipulations expérimentales sont possibles dans un environnement immersif (Blascovich et al., 2002). Cependant, bien que les comportements non verbaux

soient un domaine extrêmement important de la recherche sur la communication lors des soins, ces comportements peuvent être difficiles et fastidieux à coder et quantifier (Persky, 2011). D'autre part, on notera que le réalisme et la qualité de l'immersion sont conditionnés par le matériel assurant l'immersion dans l'environnement virtuel (**Figure 217**). En effet, ce dernier ne peut pas, à l'heure actuelle, offrir des rendus identiques à la « réalité » et peut même engendrer chez les utilisateurs des sensations d'inconfort. En outre, l'utilisation de l'électronique et de l'informatique engendre nécessairement des retards et erreurs de mesure pouvant compromettre la validité des observations réalisées (Loomis et al., 1999) ainsi qu'une sensation d'inconfort chez les utilisateurs.

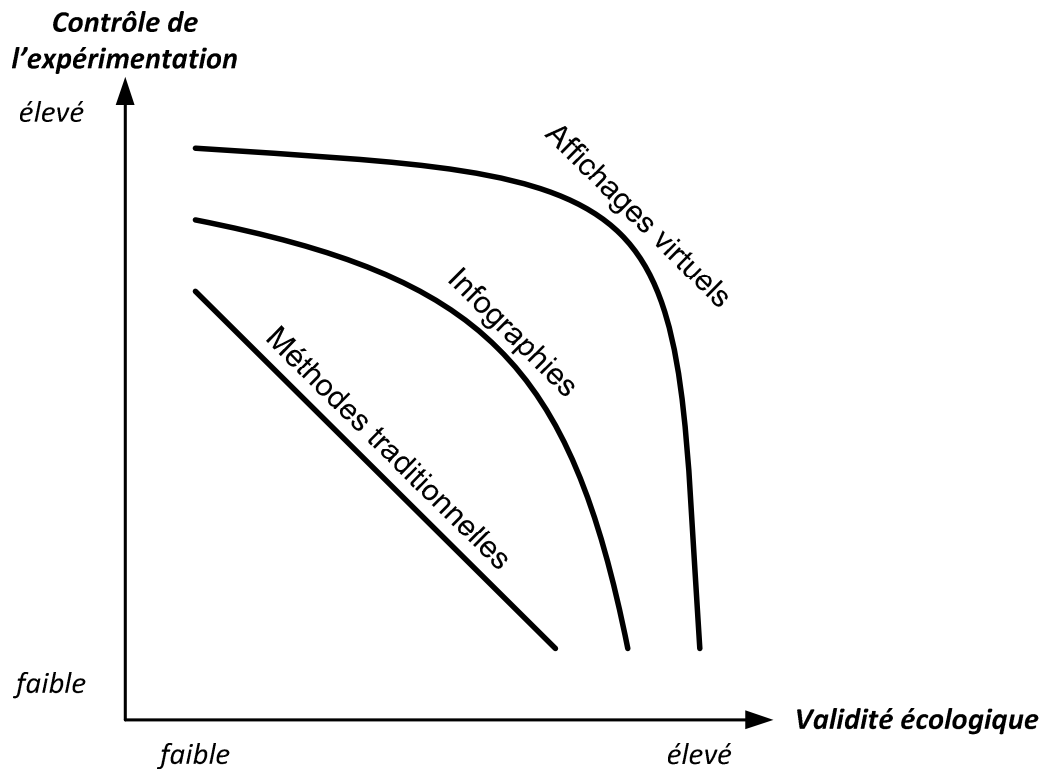


Figure 217 : Représentation du rapport entre contrôle de l'expérimentation en psychologie par rapport à sa validité écologique (d'après (Blascovich et al., 2002; Loomis et al., 1999))

Parmi les avantages méthodologiques notables conférés par les environnements virtuels immersifs, on notera leur capacité à maintenir simultanément un contrôle expérimental élevé et le réalisme (**Figure 217**), leur aptitude à en manipuler les variables de nouvelle manière et leur pouvoir de mesure du comportement unique. En effet, en général, plus une situation expérimentale se rapproche de la « vie réelle », moins les expérimentateurs ont de contrôle sur l'environnement de l'expérience. Les technologies d'environnement virtuel permettent alors simultanément de réduire l'impact des manipulations/instructions expérimentales sur les participants en conservant le réalisme et le caractère répliquable de la situation (Blascovich et al., 2002; Loomis et al., 1999; Persky, 2011). Il est ainsi possible de tirer, d'expériences conduites de la sorte, des conclusions qui ont de bien meilleures chances de rester valides dans des situations de la vie quotidienne (Klinger, 2006). Dans cette étude succincte du domaine, nous pouvons déduire l'importance potentielle des techniques de réalité virtuelle pour la réalisation de futures applications destinées à fonctionner sur la

PAMMA, avec un fort niveau supposé de validité clinique aussi bien pour le diagnostic que la remédiation. Ce domaine devra ainsi faire l'objet d'investigations plus approfondies ainsi que de réalisations de prototypes dans le cadre de nos développements à venir.

C) La réalisation d'un casque « tout-en-un »

Une des finalités de notre projet est de créer, à terme, un unique matériel permettant l'acquisition de l'ensemble des données produites par les patients, mais qui soit également utilisable comme dispositif d'IHM interactif. Ce matériel constitue la partie tangible d'une interface multimodale (**Figure 179**). Dans le contexte de notre application de prise en charge de la dyslexie, l'utilité de cette interface tangible se limite, pour le moment, à l'acquisition passive de données. C'est pourquoi, nous limiterons actuellement le besoin fondamental de ce dispositif à ce type d'usage (**Figure 218**).

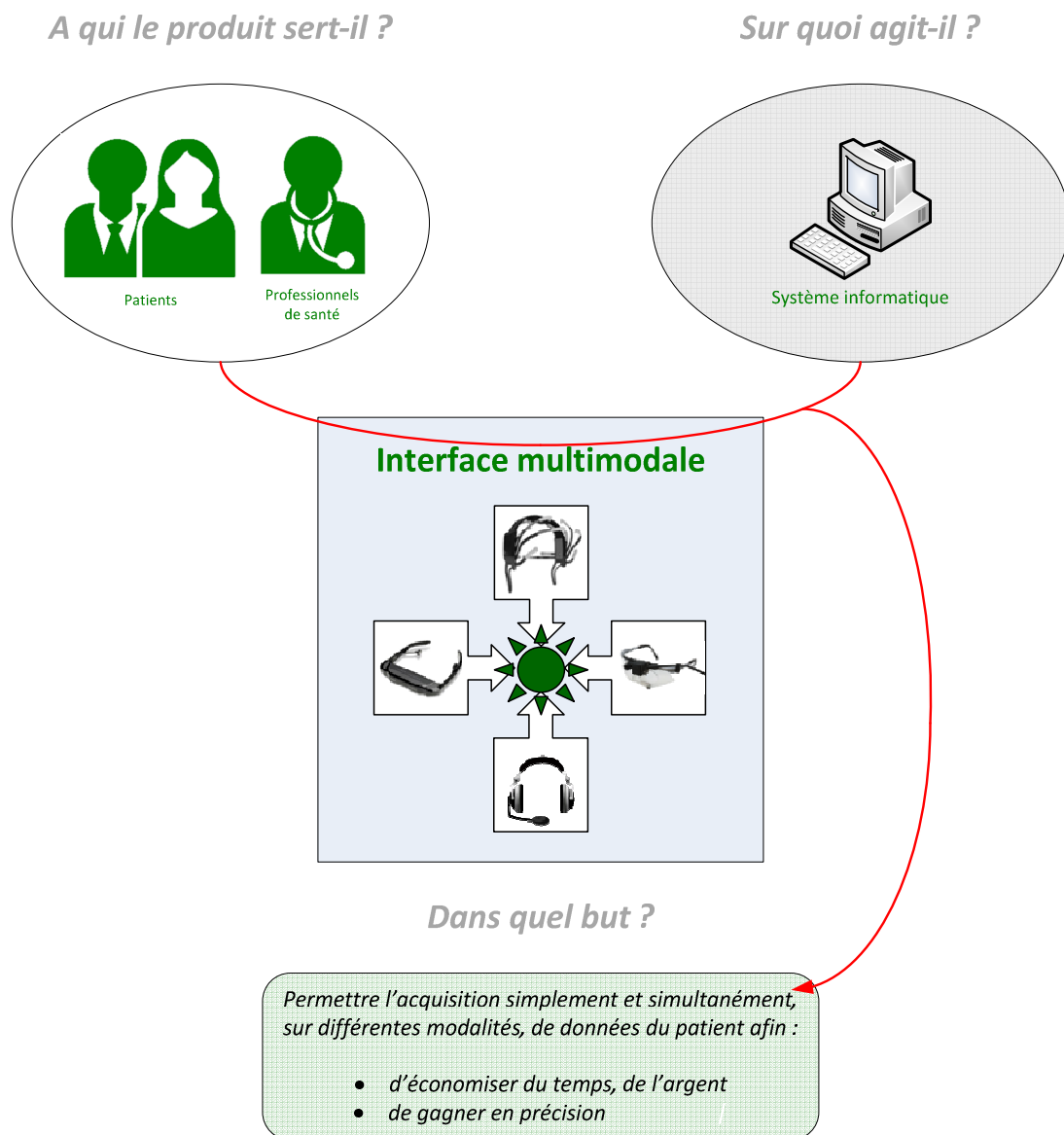


Figure 218 : Représentation fonctionnelle du besoin fondamental de l'interface multimodale

Pour répondre à ce besoin, notre interface multimodale sera composée d'un support de type « casque », de capteurs et de transducteurs ainsi que de l'électronique nécessaire à l'intégration des mesures et de leur acheminement jusqu'à un système informatique. Il nous est d'ailleurs possible de clairement définir et schématiser les acteurs et fonctions nécessaires à la satisfaction du besoin fondamental (**Figure 219**).

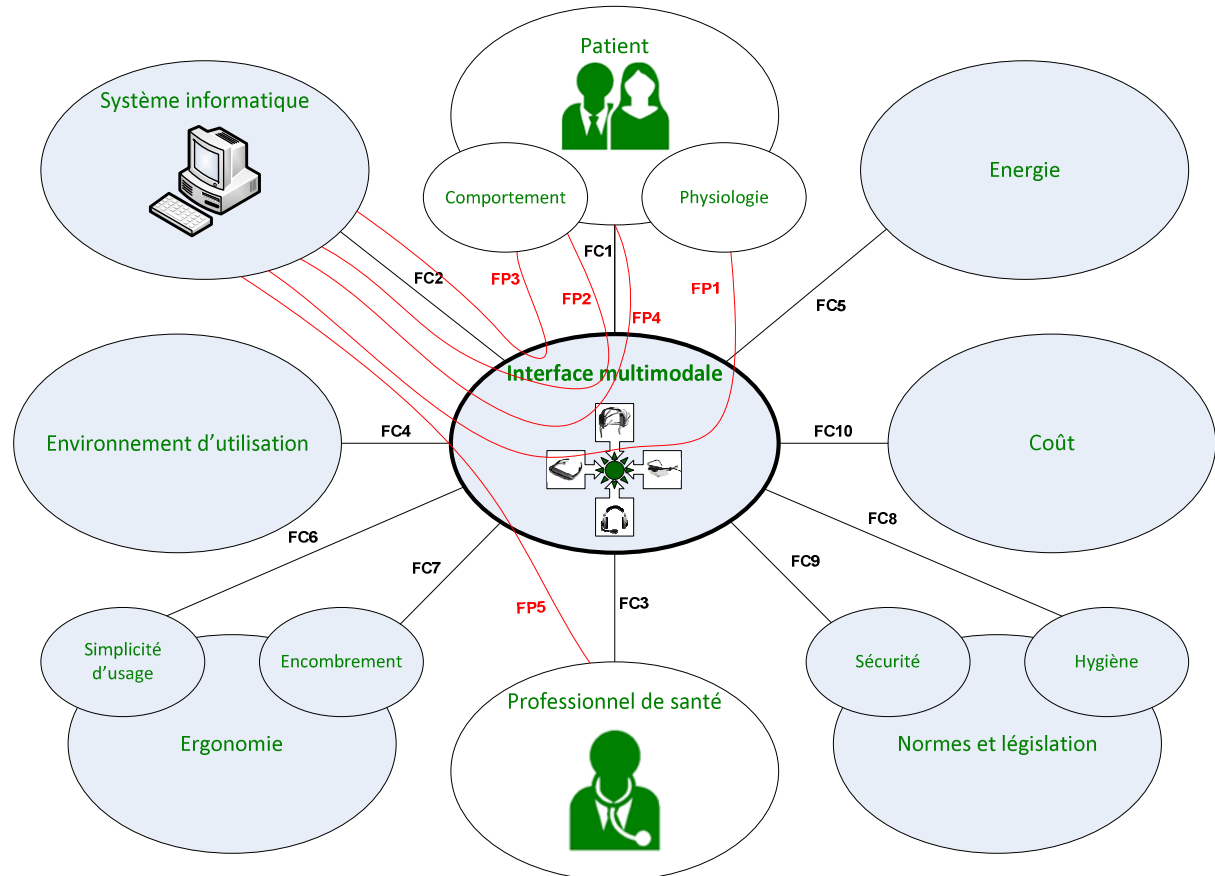


Figure 219 : Diagramme des interacteurs de l'interface multimodale

Ce diagramme met en évidence un certain nombre de **Fonctions Principales** (en rouge) et de **Fonctions Complémentaires** (en noir) qui sont requises dans le contexte d'usage pour lequel nous avons imaginé ce matériel. Le **Tableau 35** présente le détail des différentes fonctions représentées sur la **Figure 219**.

Au même titre que toute IHM permettant à des utilisateurs identifiés d'atteindre un objectif spécifié, ce matériel devra répondre aux critères d'utilisabilité ISO 9241-11 (Carroll, 2001) :

- La fiabilité, qui correspond à maximiser la qualité du résultat obtenu.
- L'efficacité, à savoir l'optimisation du processus d'obtention du résultat.
- La satisfaction, c'est-à-dire la génération d'une expérience utilisateur positive.

Ainsi, nous devons nous appuyer sur des métriques quantitatives et qualitatives centrées autour de l'individu et du contexte dans lequel il va utiliser notre technologie pour évaluer ces critères. Ne pas procéder de la sorte reviendrait à négliger une partie de la problématique et donc à potentiellement proposer une solution inappropriée, ne tenant pas compte de la spécificité de chaque utilisateur (Mele & Federici, 2012a).

En particulier, le critère de satisfaction prend pour nous une grande importance car le dispositif doit pouvoir être utilisé dans un contexte applicatif se voulant ludique, c'est-à-dire assimilable à celui de jeux vidéos (à visée médicale/paramédicale dans notre cas). En effet, nous voulons que notre système puisse accroître l'engagement de l'enfant dans son parcours de soins, et les jeux vidéo utilisent l'apprentissage, la résolution de problèmes et la maîtrise au service de l'engagement et du plaisir (Gee, 2008). De la sorte, le critère de satisfaction va devoir être approfondi/étendu afin de s'adapter à ce contexte récréatif (Allen et al., 2003). A ce titre, les interfaces mises en place, à la fois matérielles et logicielles, se devront d'être divertissantes, et, dans ce contexte, l'interface matérielle peut carrément être apparentée à un jouet et devra donc répondre aux impératifs d'un tel objet (Schell, 2010).

Fonctions principales	Fonctions complémentaires ou contraintes
FP1 : Assurer l'acquisition sur le patient de données électroencéphalographiques	FC1 : S'adapter à la physionomie et la physiologie de l'utilisateur
FP2 : Assurer l'acquisition sur le patient de données permettant son suivi oculaire	FC2 : Garantir une connectabilité avec le système informatique
FP3 : Assurer l'acquisition des productions verbales du patient	FC3 : Intégrer et couvrir les pratiques et processus de soins utilisés par le professionnel, en garantissant la qualité des résultats proposés
FP4 : Permettre la présentation au patient de stimuli auditifs et visuels	FC4 : Fonctionner dans un environnement remplissant des conditions adéquates d'usage
FP5 : Permettre le pilotage du dispositif par un professionnel de santé	FC5 : Assurer une alimentation en électricité
	FC6 : Etre utilisable très simplement
	FC7 : Entraver le moins possible le porteur
	FC8 : Respecter les exigences concernant les appareils utilisés pour les examens médicaux
	FC9 : Respecter la réglementation en vigueur pour ce genre d'appareil électronique
	FC10 : Etre bon marché autant que possible

Tableau 35 : Fonctions des interacteurs de l'interface multimodale

Une des perspectives de notre dispositif est qu'il soit accessible à un large public. Il doit donc être bon marché et simple à utiliser afin de trouver des usages dans un large éventail d'applications (Schneider et al., 2011), ce qui a été rendu possible aujourd'hui par la large diffusion de l'ordinateur personnel (Mele & Federici, 2012a). Cependant les critères de coût du dispositif et la complexité des traitements de signaux à effectuer sont antagonistes (De Santis & Iacoviello, 2009a). Il va donc nous falloir trouver un juste compromis entre les performances offertes par un matériel bon marché et les traitements algorithmiques qui seront effectués sur les données transmises par celui-ci. Dans cette même logique de large

application/utilisation de notre dispositif, ce dernier sera donc amené à avoir plus qu'un usage d'appareil de mesure/capteur et pourra donc être utilisé comme un véritable périphérique d'entrée (Miyoshi & Murata, 2001a). A ce titre, il faut penser à prendre en compte d'autres facteurs matériels qui peuvent être limitant, tel que le nombre de contrôleurs USB présents sur l'ordinateur, par exemple, ainsi que leur bande passante (Schneider et al., 2011). En effet, négliger ce genre d'aspects peut mener à un ralentissement de l'acquisition du matériel et réduire les possibilités qu'il offre.

Nous ne rentrons pas dans les détails des capteurs, des transducteurs et du support mécanique que cette interface multimodale requiert. En effet, la réalisation des nombreuses fonctions de ce casque mérite des études approfondies de chaque modalité prise isolément avant d'analyser les interactions et recouvrements, aussi bien fonctionnels que techniques, que leur association représente. A ce titre, nous avons déjà commencé ce travail en réalisant un cahier des charges fonctionnel spécifique à la modalité de suivi oculaire (voir l'annexe **Le prototypage de l'interface multimodale**).

Bien que nous n'en ayons pas réellement disserté jusqu'à maintenant, on signalera qu'une PAMMA doit permettre l'utilisation en toute sécurité d'appareils médicaux en en prévenant tout mode de fonctionnement dangereux pour les patients (Arney et al., 2014), qui peuvent, par exemple, relever d'une interaction entre plusieurs appareils (Hatcliff et al., 2012). L'interopérabilité entre les appareils doit d'ailleurs permettre de conserver le même niveau de sûreté et d'efficacité quel que soit le matériel utilisé pour une même tâche. A ce titre, il faudra apporter un soin particulier à chaque module et composant de l'interface multimodale afin de garantir la sécurité de ses utilisateurs ainsi que le respect de la législation en vigueur.

Conclusion

Avec cette recherche, notre but global était de proposer un outil informatique pour pallier les difficultés rencontrées lors de la prise en charge de la dyslexie. En effet, un certain nombre de barrières, inhérentes à la caractérisation clinique par exclusion de la dyslexie, à la multitude de praticiens impliqués ainsi qu'au manque d'objectivité de certaines méthodes existantes, entravent l'efficacité de la prise en charge de cette pathologie. Au premier abord, la chose qui était claire, à la vue de ces difficultés, était que rien ne l'était réellement.

Notre travail a donc débuté par l'établissement clair de la problématique de notre recherche. Pour cela, nous avons adopté une approche éclectique afin de ne négliger consciemment aucune piste. Il nous a alors été donné d'identifier les difficultés majeures posées par la prise en charge de la dyslexie ainsi que différents axes d'amélioration, rendus possibles par l'outil informatique, pour tenter d'y remédier. Nous avons alors commencé à entrevoir les pourtours d'un système informatique et nous sommes fixés des objectifs pour en prouver la réalisabilité.

Ensuite, nous avons cherché comment pragmatiquement mettre en place une méthode de développement visant à intégrer au sein d'un seul et unique système l'ensemble des besoins que nous avons fait apparaître. Cette étape de notre travail a ainsi été l'occasion d'étudier les modèles de développement existants et les modèles d'analyse utilisables pour représenter le fonctionnement de notre système. De ces efforts ont émergé plusieurs frameworks :

- Un framework méthodologique, qui définit un cadre de développement.
- Un framework architectural, pour l'implémentation d'applications médicales, comprenant :
 - o Un niveau de description fonctionnelle (le niveau 3 de l'éditeur).
 - o Deux niveaux de description technique au travers de SMA (le niveau 2 de l'éditeur), d'agents et de protocoles d'interaction (le niveau 1 de l'éditeur).

Nous avons ainsi pu prendre du recul vis-à-vis de la problématique et en avons produit une méta-analyse. En particulier, le framework architectural qui a été défini a permis d'entrevoir la conception d'une plate-forme logicielle propice au développement d'applications médicales conformes aux règles conceptuelles de ce framework. Cette plate-forme porte la dénomination de PAMMA. Par ailleurs, de par sa nature même, cette PAMMA permet potentiellement de supporter tous les champs d'applications des SMA tels que décrits dans la littérature : son usage s'étend donc théoriquement bien au-delà de la sphère applicative de notre recherche.

La suite de notre apport a consisté à modéliser un système générique de prise en charge de la dyslexie. Ce système se présente au final véritablement comme un chapeau sous lequel vont pouvoir être développées des applications spécifiques au diagnostic ou à la remédiation de la dyslexie. Il pourra d'ailleurs être implémenté sur la PAMMA et il permettra de réduire les efforts d'intégration de ces applications spécifiques. D'un point de vue fonctionnel, il servira à uniformiser l'utilisation des différentes applications de prise en charge de la dyslexie ainsi que la gestion des patients et de leurs DMP.

Nous avons également investigué du côté du matériel qui pourrait être utilisé par les applications de la PAMMA. A ce titre, il nous a paru judicieux de vouloir prendre en charge des appareils permettant l'établissement de mesures objectives. Ainsi, nous avons posé les jalons du développement d'un système de suivi oculaire bon marché et intégré un casque EEG grand public. Notre étude nous indique qu'il est donc entièrement possible de concevoir et produire des appareils bon marché, aux résultats cliniquement exploitables, qui pourront être utilisés pour améliorer la qualité de la prise en charge des individus souffrant de dyslexie.

En résumé, notre apport au domaine de la prise en charge informatisée de la dyslexie a été celui de concepteurs et cette thèse porte majoritairement sur le domaine de la méta-implémentation. Effectivement, nous nous sommes efforcés de concevoir un outil permettant ultérieurement l'implémentation, qui consiste à définir les concepts du domaine et les relations qui existent entre eux, puis l'exécution d'applications médicales. Dans le cas de la prise en charge des troubles neuropsychologiques, ces utilisations applicatives auront lieu en contexte médical/paramédical et seront effectuées à la demande de professionnels de santé. La **Figure 220** résume la relation entre ces différents niveaux d'abstraction conceptuelle.

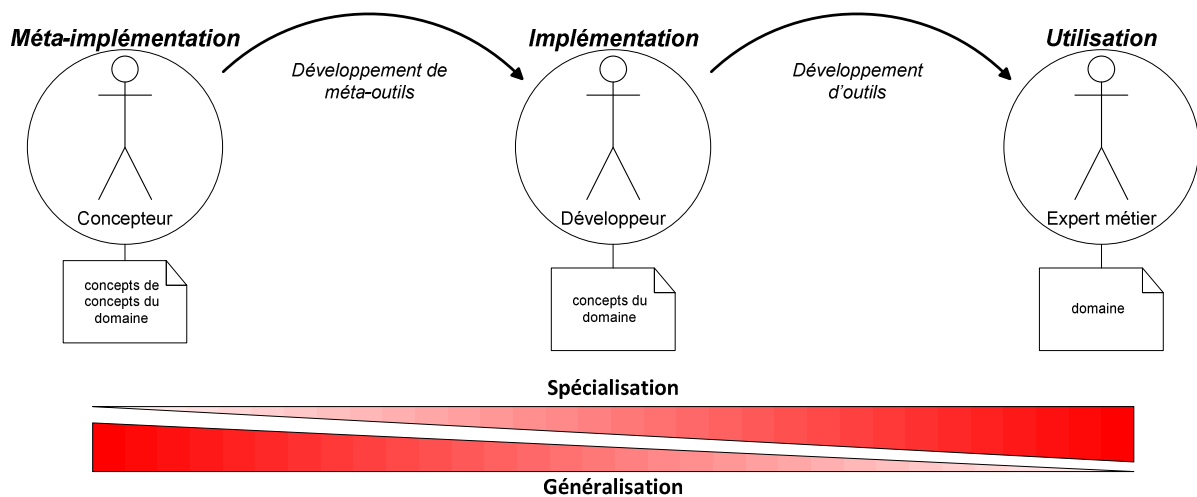


Figure 220 : Les différents niveaux d'abstraction traités dans cette thèse

Notre travail a permis de mettre en évidence une approche prometteuse. Néanmoins, les difficultés que nous avons pu rencontrer ont fait apparaître la nécessité de disposer de plus de moyens, tant financiers et qu'en nombre de personnes impliquées. Effectivement, ces moyens devront être largement supérieurs à ceux qui nous ont été alloués à l'occasion de cette recherche pour que ce projet puisse passer au stade de son industrialisation. Cette industrialisation pourrait alors débiter par l'implémentation de notre framework MA ainsi que du noyau de la PAMMA afin de permettre rapidement d'intégrer des processus de soins d'une manière immersive et ludique. Le développement d'un casque multimodal, relativement bon marché et tirant partie de l'architecture MA des applications de la PAMMA, serait également un atout qui pourrait permettre de moderniser des pratiques qui manquent bien souvent d'objectivité.

Bibliographie

- Abbas, S. Q., & Srivastava, N. (2012). Development of Framework for Automatic Speech Recognition. *International Journal on Computer Science and Engineering*, 4(05), 707–710.
- Abouabdellah, A., Aguezoul, A., & Abdelghani, C. (2010). Application des méthodes du génie logiciel à la modélisation du processus d'autonomisation d'un patient atteint d'un cancer. In *Conférence Francophone en Gestion et Ingénierie des Systèmes Hospitaliers (GISEH10)* (pp. 1–8). Clermont-Ferrand, France.
- Académie de Dijon. (2009). *Guide enseignants : La dyslexie*.
- ACM Inc. (2015). ACM Digital Library. Retrieved from <http://dl.acm.org/>
- Adam, E. (2000). *Modèle d'organisation multi-agent pour l'aide au travail coopératif dans les processus d'entreprise: application aux systèmes administratifs complexes*. Université de Valenciennes et du Hainaut-Cambrésis.
- Adam, E., & Mandiau, R. (2005). Roles and hierarchy in multi-agent organizations. In *Multi-Agent Systems and Applications IV* (pp. 539–542). Springer Berlin Heidelberg. doi:10.1007/11559221_55
- Ahn, S. J., & Rauh, W. (1999). Geometric least squares fitting of circle and ellipse. *International Journal of Pattern Recognition and Artificial Intelligence*, 13(7), 987–996.
- Aigner, W., Kaiser, K., & Miksch, S. (2008). Visualization methods to support guideline-based care management. In A. Ten Teije, S. Miksch, & P. Lucas (Eds.), *Computer-based Medical Guidelines and Protocols: A Primer and Current Trends* (pp. 140–159). IOS Press. doi:10.3233/978-1-58603-873-1-140
- Aigner, W., & Miksch, S. (2006). CareVis: integrated visualization of computerized protocols and temporal patient data. *Artificial Intelligence in Medicine*, 37(3), 203–218. doi:10.1016/j.artmed.2006.04.002
- Al-Edaily, A., Al-Wabil, A., & Al-Ohali, Y. (2013a). Dyslexia Explorer: A Screening System for Learning Difficulties in the Arabic Language Using Eye Tracking. In A. Holzinger, M. Ziefle, M. Hitz, & M. Debevc (Eds.), *Human Factors in Computing and Informatics* (pp. 831–834). Springer Berlin Heidelberg. doi:10.1007/978-3-642-39062-3_63
- Al-Edaily, A., Al-Wabil, A., & Al-Ohali, Y. (2013b). Interactive Screening for Learning Difficulties: Analyzing Visual Patterns of Reading Arabic Scripts with Eye Tracking. In *HCI International 2013* (pp. 3–7). Las Vegas, NV, USA: Springer Berlin Heidelberg. doi:10.1007/978-3-642-39476-8_1
- Alhir, S. S. (2002). Understanding the Unified Process (UP). *Methods & Tools*, 10(1), 2–17.
- Allen, A., & Hayes, J. (1994). Patient satisfaction with telemedicine in a rural clinic. *American Journal of Public Health*, 84(10), 1693. doi:10.2105/AJPH.84.10.1693

- Allen, M. J., Gough, N., Mehdi, Q., & Wink, B. (2003). A review of eye-tracking and usability in computer games. In *4th International Conference on Intelligent Games and Simulation GAMEON 2003 Proceedings (2003)* (pp. 26–30).
- Allison, R. S., Eizenman, M., & Cheung, B. S. K. (1996). Combined head and eye tracking system for dynamic testing of the vestibular system. *IEEE Transactions on Biomedical Engineering*, *43*(11), 1073–1082. doi:10.1109/10.541249
- Alm, J. (2004). *Dyslexia: Relevance of concepts, validity of measurements, and cognitive functions*. Uppsala University.
- Al-Rahayfeh, A., & Faezipour, M. (2013). Eye Tracking and Head Movement Detection: A State-of-Art Survey. *IEEE Journal of Translational Engineering in Health and Medicine*, *1*(August), 11–22. doi:10.1109/JTEHM.2013.2289879
- Alsinet, T., Béjar, R., Fernández, C., & Manyà, F. (2000). A Multi-agent system architecture for monitoring medical protocols. In C. Sierra, M. Gini, & J. S. Rosenschein (Eds.), *the fourth international conference on Autonomous agents (AGENTS'00)* (pp. 499–505). Barcelona, Spain: ACM Press. doi:10.1145/336595.337580
- Altschuler, E. L., Altschuler, B. M., Altschuler, D. L., Samber, D., & Ramachandran, V. S. (2007). “ ‘ ‘ ‘ Stroop Concordant ’ ’ ’ coloring of letters for remediation of dyslexia. *Medical Hypotheses*, *69*(2), 381–382. doi:10.1016/j.mehy.2006.12.032
- Al-Wabil, A., & Al-Sheaha, M. (2010). Towards an interactive screening program for developmental dyslexia: eye movement analysis in reading Arabic texts. In K. Miesenberger, J. Klaus, W. Zagler, & A. Karshmer (Eds.), *Computers Helping People with Special Needs* (pp. 25–32). Springer Berlin Heidelberg. doi:10.1007/978-3-642-14100-3_5
- Al-Wabil, A., Meldah, E., Al-Suwaidan, A., & AlZahrani, A. (2010). Designing Educational Games for Children with Specific Learning Difficulties: Insights from Involving Children and Practitioners. *2010 Fifth International Multi-Conference on Computing in the Global Information Technology*, 195–198. doi:10.1109/ICCGI.2010.43
- Ambler, S. W. (1998). *An introduction to process patterns*.
- American Clinical Neurophysiology Society. (2006). *Guideline 5: Guidelines for Standard Electrode Position Nomenclature*.
- American Clinical Neurophysiology Society. (2008). *Standard for Transferring Digital Neurophysiological Data Between Independent Computer Systems* (Vol. 1).
- American Clinical Neurophysiology Society. (2015). American Clinical Neurophysiology Society. Retrieved from <http://www.acns.org/>
- American Physical Therapy Association. (2014). Coding for Interventions (CPT-4). Retrieved from <http://www.apta.org/Payment/Coding/Interventions/>
- American Psychological Association. (2000). *Diagnostic and Statistical Manual of Mental Disorders, Fourth Edition, Text Revision (DSM-IV-TR)*. Text (Vol. 1). American Psychiatric Association. doi:10.1176/appi.books.9780890423349
- American Society for Testing and Materials. (2000). ASTM E1467-94(2000), Standard Specification for Transferring Digital Neurophysiological Data Between Independent

- Computer Systems (Withdrawn 2004). Retrieved from <http://www.astm.org/Standards/E1467.htm>
- American Society for Testing and Materials. (2013). ASTM F2761-09(2013), Medical Devices and Medical Systems - Essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE) - Part 1: General requirements and conceptual model. Retrieved from <http://www.astm.org/Standards/F2761.htm>
- Ametller, J., Robles, S., & Ortega-Ruiz, J. A. (2004). Self-protected mobile agents. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '04)* (pp. 362–367). New York, New York, USA: IEEE. doi:10.1109/AAMAS.2004.241
- Anderson, J., Fleek, F., Garrity, K., & Drake, F. (2001). Integrating usability techniques into software development. *IEEE Software*, 18(1), 46–53. doi:10.1109/52.903166
- Anderson, R. J. (1996). A security policy model for clinical information systems. In *IEEE Symposium on Security and Privacy* (pp. 30–43). Oakland, CA: IEEE. doi:10.1109/SECPRI.1996.502667
- Andersson, R., Nyström, M., & Holmquist, K. (2010). Sampling frequency and eye-tracking measures: how speed affects durations, latencies, and more. *Journal of Eye Movement Research*, 3(3), (6)1–12.
- Andonoff, E., Bouaziz, W., Hanachi, C., & Bouzguenda, L. (2009). An Agent-Based Model for Autonomic Coordination of Inter-Organizational Business Processes. *Informatica*, 20(3), 323–342.
- Angelini, E., Jin, Y., & Laine, A. (2005). State of the art of level set methods in segmentation and registration of medical imaging modalities. In *Handbook of Biomedical Image Analysis - Volume III: Registration Models* (pp. 47–101). Springer US. doi:10.1007/0-306-48608-3_2
- Ans, B., Carbonnel, S., & Valdois, S. (1998). A connectionist multiple-trace memory model for polysyllabic word reading. *Psychological Review*, 105(4), 678–723. doi:10.1037/0033-295X.105.4.678-723
- AOS Group. (2008). *JACK®: An Agent Infrastructure for Providing the Decision-Making Capability Required for Autonomous Systems*.
- AOS Group. (2014). JACK®. AOS Group. Retrieved from <http://aosgrp.com/products/jack/>
- Apelon Inc. (2014). Apelon's TermWorks. Apelon Inc. Retrieved from <http://apelon.com/Products/TermWorks/tabid/98/Default.aspx>
- Aridor, Y., & Lange, D. B. (1998). Agent design patterns: Elements of Agent Application Design. In *Second international conference on Autonomous agents - AGENTS '98* (pp. 108–115). New York, New York, USA: ACM Press. doi:10.1145/280765.280784
- Arnbak, E., & Elbro, C. (2000). The effects of morphological awareness training on the reading and spelling skills of young dyslexics. *Scandinavian Journal of Educational Research*, 44(3), 229–251. doi:10.1080/00313830050154485
- Arney, D., Fischmeister, S., Goldman, J. M., Lee, I., & Trausmuth, R. (2009). Plug-and-play for medical devices: Experiences from a case study. *Biomedical Instrumentation & Technology*, 43(4), 313–317. doi:10.2345/0899-8205-43.4.313

- Arney, D., Plourde, J., Schrenker, R., Mattegunta, P., Whitehead, S. F., & Goldman, J. M. (2014). Design Pillars for Medical Cyber-Physical System Middleware. In V. Turau, M. Kwiatkowska, R. Mangharam, & C. Weyer (Eds.), *5th Workshop on Medical Cyber-Physical Systems* (pp. 124–132). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. doi:10.4230/OASlcs.MCPS.2014.124
- Asikainen, M. (2005). *Diagnosing specific language impairment*. Tampere University Press.
- Assecondi, S., Bianchi, A. M., Hallez, H., Staelens, S., Casarotto, S., Lemahieu, I., & Chiarenza, G. A. (2009). Automated identification of ERP peaks through Dynamic Time Warping : An application to developmental dyslexia. *Clinical Neurophysiology*, 120(10), 1819–1827. doi:10.1016/j.clinph.2009.06.023
- Assemblée Nationale. LOI n°2005-102 du 11 février 2005 pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées. , Pub. L. No. SANX0300217L (2005). JOURNAL OFFICIEL DE LA RÉPUBLIQUE FRANÇAISE.
- Asteriadis, S., Tzouveli, P., Karpouzis, K., & Kollias, S. (2007). Non-Verbal Feedback on User Interest Based on Gaze Direction and Head Pose. In *Second International Workshop on Semantic Media Adaptation and Personalization (SMAP 2007)* (pp. 171–178). Ieee. doi:10.1109/SMAP.2007.50
- Aubry, C. (2013). *Scrum : Le guide pratique de la méthode agile la plus populaire* (3ème ed.). Dunod.
- Babcock, J. S., & Pelz, J. B. (2004). Building a lightweight eyetracking headgear. In A. T. Duchowski & R. Vertegaal (Eds.), *Symposium on Eye Tracking Research & Applications (ETRA'04)* (pp. 109–114). San Antonio, TX, USA: ACM Press. doi:10.1145/968363.968386
- Badcock, N. a, Mousikou, P., Mahajan, Y., de Lissa, P., Thie, J., & McArthur, G. (2013). Validation of the Emotiv EPOC® EEG gaming system for measuring research quality auditory ERPs. *PeerJ*, 1, e38. doi:10.7717/peerj.38
- Bakar, N. A., & Selamat, A. (2013). Assessing Agents Interaction Quality via Multi-agent Runtime Verification. In C. Bădică, N. T. Nguyen, & M. Brezovan (Eds.), *Computational Collective Intelligence. Technologies and Applications* (pp. 175–184). Craiova, Romania: Springer Berlin Heidelberg. doi:10.1007/978-3-642-40495-5_18
- Balzer, R., Cheatham, T. E., & Green, C. (1983). Software Technology in the 1990's: Using a New Paradigm. *Computer*, 16(11), 39–45. doi:10.1109/MC.1983.1654237
- Banerjee, S., & Bhattacharya, M. (2010). Segmentation of medical images using Selective Binary and Gaussian Filtering regularized level set (SBGFRLS) method. In *Biomedical Engineering and Informatics (BMEI 2010)* (pp. 541–545).
- Bång, M., & Eriksson, H. (1997). Generation of development environments for the Arden Syntax. In *AMIA Annual Symposium Proceedings* (pp. 313–317). Nashville, TN, USA: American Medical Informatics Association.
- Baptiste, J. L. (2012). *Merise - Guide pratique : modélisation des données et des traitements, manipulations avec le langage SQL* (2ème ed.). Editions ENI.

- Barbuceanu, M., & Fox, M. S. (1995). COOL: A Language for Describing Coordination in Multi Agent Systems. In V. Lesser (Ed.), *First International Conference on Multi-Agent Systems (ICMAS'95)* (pp. 17–24). San Francisco, California, USA: AAAI/MIT Press.
- Barbuceanu, M., & Lo, W.-K. (1999). Conversation oriented programming in COOL: current state and future directions. In *Agents' 99: Working Notes of the Workshop on Specifying and Implementing Conversation Policies*.
- Barrouillet, P., Billard, C., de Agostini, M., Démonet, J.-F., Fayol, M., Gombert, J.-E., ... Rondet-Grellier, C. (2007). *Dyslexie, dysorthographe, dyscalculie : bilan des données scientifiques*. INSERM.
- Bass, L., Clements, P., & Kazman, R. (2003). *Software architecture in practice* (2nd ed.). Addison-Wesley Professional.
- Bass, L., Faneuf, R., Little, R., Mayer, N., Pellegrino, B., Reed, S., ... Szczur, M. R. (1992). A metamodel for the runtime architecture of an interactive system. *SIGCHI Bulletin*, 24(1), 32–37. doi:10.1145/142394.142401
- Bates, D. W., Cohen, M., Leape, L. L., Overhage, J. M., Shabot, M. M., & Sheridan, T. (2001). Reducing the frequency of errors in medicine using information technology. *Journal of the American Medical Informatics Association*, 8(4), 299–309. doi:10.1136/jamia.2001.0080299
- Bates, D. W., & Gawande, A. A. (2003). Improving Safety with Information Technology. *The New England Journal of Medicine*, 348(25), 2526–2534. doi:10.1056/NEJMs020847
- Bauer, B. (2002). UML class diagrams revisited in the context of agent-based systems. In M. J. Wooldridge, G. Weiß, & P. Ciancarini (Eds.), *Agent-Oriented Software Engineering II* (pp. 101–118). Montreal, Canada: Springer Berlin Heidelberg. doi:10.1007/3-540-70657-7_7
- Bauer, B., Müller, J. P., & Odell, J. (2001). Agent UML: A Formalism for Specifying Multiagent Interaction. In P. Ciancarini & M. J. Wooldridge (Eds.), *Agent-Oriented Software Engineering* (pp. 91–103). Limerick, Ireland: Springer Berlin Heidelberg. doi:10.1007/3-540-44564-1_6
- Bauer, B., & Odell, J. (2005). UML 2.0 and agents: how to build agent-based systems with the new UML standard. *Engineering Applications of Artificial Intelligence*, 18(2), 141–157. doi:10.1016/j.engappai.2004.11.016
- Bäumer, D., Gryczan, G., Knoll, R., Lilienthal, C., Riehle, D., & Züllighoven, H. (1997). Framework development for large systems. *Communications of the ACM*, 40(10), 52–59. doi:10.1145/262793.262804
- Bayardo, R. J. J., Bohrer, W., Brice, R., Cichocki, A., Fowler, J., Helal, A., ... Woelk, D. (1997). InfoSleuth: agent-based semantic integration of information in open and dynamic environments. In J. M. Peckman, S. Ram, & M. Franklin (Eds.), *ACM SIGMOD international conference on Management of data (SIGMOD'97)* (pp. 195–206). Tucson, AZ, USA: ACM Press. doi:10.1145/253260.253294
- Beaudouin-Lafon, M. (2001). 40 ans d'interaction homme-machine : points de repère et perspectives. In *A la rencontre de l'interaction - Journées ASTI 2001* (pp. 1–9). Paris, France.

- Beaudouin-Lafon, M., & Mackay, W. (2007). Prototyping tools and techniques. In A. Sears & J. A. Jacko (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications* (2nd ed., pp. 1017–1039). CRC Press. doi:10.1201/9781410615862.ch52
- Becchetti-Bizot, C., Assouline, D., Bambuck, R., Brisset, C., Chateau, J.-Y., Claus, P., ... Simler, B. (2008). *La scolarisation des élèves handicapés*.
- Behrens, F., MacKeben, M., & Schröder-Preikschat, W. (2010). An improved algorithm for automatic detection of saccades in eye movement data and for calculating saccade parameters. *Behavior Research Methods*, 42(3), 701–708. doi:10.3758/BRM.42.3.701
- Bellifemine, F., Caire, G., Poggi, A., & Rimassa, G. (2003). JADE: A white paper. *EXP - in Search of Innovation*, 3(3), 6–19.
- Bellifemine, F., Caire, G., Poggi, A., & Rimassa, G. (2008). JADE: A software framework for developing multi-agent applications. Lessons learned. *Information and Software Technology*, 50(1-2), 10–21. doi:10.1016/j.infsof.2007.10.008
- Ben Gurion University. (2004). Uruz. Ben Gurion University. Retrieved from http://medinfo.ise.bgu.ac.il/medlab/ResearchProjects/RP_Uruz3.htm
- Benington, H. D. (1987). Production of large computer programs. In *9th International Conference on Software Engineering (ICSE '87)* (pp. 299–310). IEEE. doi:10.1109/MAHC.1983.10102
- Bergenti, F., & Poggi, A. (2000). Exploiting UML in the Design of Multi-agent Systems. In A. Omicini, R. Tolksdorf, & F. Zambonelli (Eds.), *Engineering Societies in the Agents World* (pp. 106–113). Berlin, Germany: Springer Berlin Heidelberg. doi:10.1007/3-540-44539-0_8
- Berlin, A., Sorani, M., & Sim, I. (2006). A taxonomic description of computer-based clinical decision support systems. *Journal of Biomedical Informatics*, 39(6), 656–667. doi:10.1016/j.jbi.2005.12.003
- Bernon, C., Gleizes, M.-P., & Picard, G. (2009). Méthodes orientées agent et multi-agent. In A. El FallahSeghrouchni & J.-P. Briot (Eds.), *Technologies des systèmes multi-agents et applications industrielles* (IC2 ed., pp. 45–76). Lavoisier.
- Bernon, C., Gleizes, M.-P., Picard, G., & Glize, P. (2002). The ADELFE Methodology For an Intranet System Design. In *Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002 at CAiSE'02)* (pp. 1–15). Toronto, Ontario, Canada: CEUR-WS.
- Bernsen, N. O. (1994). Foundations of multimodal representations: a taxonomy of representational modalities. *Interacting with Computers*, 6(4), 347–371. doi:10.1016/0953-5438(94)90008-6
- Bertrand, G. (1994). Simple points, topological numbers and geodesic neighborhoods in cubic grids. *Pattern Recognition Letters*, 15(10), 1003–1011. doi:10.1016/0167-8655(94)90032-9
- Bertrand, G., Everat, J.-C., & Couprie, M. (1997). Image segmentation through operators based on topology. *Journal of Electronic Imaging*, 6(4), 395–405. doi:10.1117/12.276856

- Bertrand, G., & Malandain, G. (1992). A new topological classification of points in 3d images. In *Computer Vision—ECCV'92* (pp. 710–714). Springer Berlin Heidelberg. doi:10.1007/3-540-55426-2_78
- Beymer, D., & Flickner, M. (2003). Eye gaze tracking using an active stereo head. In *Conference on Computer Vision and Pattern Recognition* (Vol. 2). IEEE. doi:10.1109/CVPR.2003.1211502
- Beymer, D., & Russell, D. M. (2005). WebGazeAnalyzer: A System for Capturing and Analyzing Web Reading Behavior Using Eye Gaze. In *CHI 2005* (pp. 1913–1916). Portland, Oregon, USA: ACM Press. doi:10.1145/1056808.1057055
- Bharadwaj, V., & Reddy, Y. V. R. (2003). A framework to support collaboration in heterogeneous environments. *ACM SIGGROUP Bulletin*, 24(3), 103–116. doi:10.1145/1052829.1052852
- Biedert, R., Buscher, G., & Dengel, A. (2009). The eyeBook – Using Eye Tracking to Enhance the Reading Experience. *Informatik-Spektrum*, 33(3), 272–281. doi:10.1007/s00287-009-0381-2
- Billington, J., Ginbayashi, J., Hillah, L., Kindler, E., Kordon, F., Petrucci, L., ... Trèves, N. (2014). Pnml.org - PNML reference site. Retrieved from <http://www.pnml.org/>
- Blascovich, J. J., Loomis, J. M., Beall, A. C., Swinth, K. R., Hoyt, C. L., & Bailenson, J. N. (2002). Immersive Virtual Environment Technology as a Methodological Tool for Social Psychology. *Psychological Inquiry: An International Journal for the Advancement of Psychological Theory*, 13(2), 103–124. doi:10.1207/S15327965PLI1302_01
- Blaszczynski, J., Farion, K., Michalowski, W., Wilk, S., Rubin, S., & Weiss, D. (2005). Mining clinical data: selecting decision support algorithm for the MET-AP system. In S. Miksch, J. Hunter, & E. T. Keravnou (Eds.), *Artificial Intelligence in Medicine* (pp. 429–433). Springer Berlin Heidelberg. doi:10.1007/11527770_58
- Blatt, M. N., Crouse, B., & Wilson, B. (2012). *Collaborative Workflows, Coordinated Care: Meeting the Challenges of 21st Century Healthcare*.
- Blonk, A. M., Van Den Bercken, J. H. L., & De Bruyn, E. E. J. (1996). Evaluation of DYSLEXPART: A comparison of a knowledge-based system with experienced clinicians in the diagnosis of dyslexia. *Computers in Human Behavior*, 12(4), 567–586. doi:10.1016/S0747-5632(96)00025-8
- Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5), 61–82. doi:10.1109/2.59
- Bogliotti, C., & Serniclaes, W. (2011). Développement d'un instrument multimédia de remédiation de la dyslexie. In *4e Journées de Phonétique Clinique* (Vol. 87, p. 1). Strasbourg, France.
- Boissier, O., Gitton, S., & Glize, P. (2004). Caractéristiques des systèmes et des applications multi-agents. In Observatoire français des techniques avancées (Ed.), *Systèmes Multi-Agents* (Arago (29)., pp. 25–54). OFTA.
- Bonitasoft Inc. (2014). Bonita BPM Studio. Bonitasoft Inc. Retrieved from <http://www.bonitasoft.com/how-we-do-it>

- Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Conallen, J., & Houston, K. A. (2007). *Object-Oriented Analysis and Design with Applications* (Vol. 33). Addison-Wesley Professional.
- Boon, H., Verhoef, M., O'Hara, D., & Findlay, B. (2004). From parallel practice to integrative health care: a conceptual framework. *BMC Health Services Research*, *4*(15), 1–5. doi:10.1186/1472-6963-4-15
- Bosse, M.-L., Tainturier, M. J., & Valdois, S. (2007). Developmental dyslexia: the visual attention span deficit hypothesis. *Cognition*, *104*(2), 198–230. doi:10.1016/j.cognition.2006.05.009
- Bosse, M.-L., & Valdois, S. (2009). Influence of the visual attention span on child reading performance: a cross-sectional study. *Journal of Research in Reading*, *32*(2), 230–253. doi:10.1111/j.1467-9817.2008.01387.x
- Bottrighi, A., Molino, G., Montani, S., Terenziani, P., & Torchio, M. (2013). Supporting a distributed execution of clinical guidelines. *Computer Methods and Programs in Biomedicine*, *112*(1), 200–210. doi:10.1016/j.cmpb.2013.04.003
- Bouchet, J. (2006). *Ingénierie de l'interaction multimodale en entrée : approche à composants ICARE*. Université Joseph Fourier - Grenoble 1.
- Bouzguenda, L., & Turki, M. (2012). Coupling clinical decision support system with computerized prescriber order entry and their dynamic plugging in the medical workflow system. In *International Conference on Information Technology and e-Services (ICITeS 2012)* (pp. 1–6). Sousse: IEEE. doi:10.1109/ICITeS.2012.6216631
- Bower, A., & McGlashan, B. (2000). Twisting the triad: The evolution of the Dolphin Smalltalk MVP application framework. In *Tutorial Paper for ESUG 2000* (pp. 1–7).
- Boxwala, A. A., Peleg, M., Tu, S. W., Ogunyemi, O., Zeng, Q. T., Wang, D., ... Shortliffe, E. H. (2004). GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines. *Journal of Biomedical Informatics*, *37*(3), 147–61. doi:10.1016/j.jbi.2004.04.002
- Boxwala, A. A., Tu, S. W., Peleg, M., Zeng, Q., Ogunyemi, O., Greenes, R. A., ... Patel, V. L. (2001). Toward a representation format for sharable clinical guidelines. *Journal of Biomedical Informatics*, *34*(3), 157–169. doi:10.1006/jbin.2001.1019
- Bradlow, A. R., Kraus, N., & Hayes, E. (2003). Speaking Clearly for Children With Learning Disabilities: Sentence Perception in Noise. *Journal of Speech, Language, and Hearing Research*, *46*, 80–97. doi:10.1044/1092-4388(2003/007)
- Bradski, G. R. (1998). Real time face and object tracking as a component of a perceptual user interface. In *Fourth IEEE Workshop on Applications of Computer Vision (WACV'98)* (pp. 214–219). IEEE Comput. Soc. doi:10.1109/ACV.1998.732882
- Bragge, M. (2013). *Model-View-Controller architectural pattern and its evolution in graphical user interface frameworks*. Lappeenranta University of Technology.
- Breen, G.-M., & Matusitz, J. (2010). An evolutionary examination of telemedicine: a health and computer-mediated communication perspective. *Social Work in Public Health*, *25*(1), 59–71. doi:10.1080/19371910902911206

- Bresac, M., & Chigros, M. (2012). *PRESCRIRE L'ORTHOPHONIE : POUR QUI ? POUR QUOI ? Création d'un DVD en vue de présenter aux médecins la profession d'orthophoniste, ses missions, ses champs d'intervention. Pour une interdisciplinarité de qualité*. Université Lille 2 Droit et Santé.
- Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3), 203–236. doi:10.1023/B:AGNT.0000018806.20944.ef
- Brewster, R., & dotPDN LLC. (2011). Paint.NET. Retrieved from <http://www.getpaint.net>
- Broca, P. (1865). Sur le siège de la faculté du langage articulé. *Bulletins de La Société D'anthropologie de Paris, 1^o série*(6), 377–393. doi:10.3406/bmsap.1865.9495
- Brumby, D. P., Cutrell, E., & Sarin, R. (2006). Eye-tracking in practice: The path to evaluating a new interface for personal desktop search. In *SIGCHI'06 Workshop*. doi:10.1.1.109.517
- Bucci, M. P., Nassibi, N., Gerard, C.-L., Bui-Quoc, E., & Seassau, M. (2012). Immaturity of the oculomotor saccade and vergence interaction in dyslexic children: evidence from a reading and visual search study. *PloS One*, 7(3), e33458. doi:10.1371/journal.pone.0033458
- Burbeck, S. (1992). *Applications programming in smalltalk-80®: How to use Model-View-Controller (MVC)*. Smalltalk-80 v2.5.
- Bury, J., Fox, J., & Sutton, D. R. (2001). The PROforma guideline specification language: progress and prospects. In B. Heller, M. Löffler, M. Musen, & M. Stefanelli (Eds.), *Computer-Based Support for Clinical Guidelines and Protocols* (pp. 13–29). IOS Press. doi:10.3233/978-1-60750-927-1-13
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern-Oriented Software Architecture (Volume 1): A System of Patterns* (Vol. Vol. 1). Wiley. doi:ISBN: 978-0-471-95869-7
- Büttgen, B., Oggier, T., Lehmann, M., Kaufmann, R., & Lustenberger, F. (2005). CCD/CMOS lock-in pixel for range imaging: Challenges, limitations and state-of-the-art. In *1st Range Imaging Research Day* (pp. 21–32).
- Caetano, A., Pereira, C., & Sousa, P. (2012). Generation of business process model views. *Procedia Technology*, 5, 378–387. doi:10.1016/j.protcy.2012.09.042
- Campbell, C. S., & Maglio, P. P. (2001). A robust algorithm for reading detection. In *2001 workshop on Perceptive user interfaces* (pp. 1–7). doi:10.1145/971478.971503
- Canonical Ltd. (2014). AnalyseSI - a Merise analysis tool. Canonical Ltd. Retrieved from <https://launchpad.net/analysesi>
- Cao, L.-B., & Dai, R.-W. (2003). Human-computer cooperated intelligent information system based on multi-agents. *Acta Automatica Sinica*, 29(1), 86–94.
- Cao, L.-B., Li, C.-S., Zhang, C.-Q., & Dai, R.-W. (2003). Open Giant Intelligent Information Systems and Its Multiagent-Oriented System Design. In *International Conference on Software Engineering Research and Practice (SERP 03)* (pp. 816–822). Las Vegas, Nevada, USA: CSREA Press.

- Carroll, J. M. (2001). *Human-Computer Interaction in the New Millennium*. (J. M. Carroll, Ed.) *Human Computer Interaction in the New Millennium* (Vol. 1st). ACM Press. doi:10.1109/MNET.2002.1081761
- Casalicchio, E., & Tucci, S. (2013). Public Administration Workflows Re-engineering: An Agent-Based M&S Approach. In M. Ganzha & L. C. Jain (Eds.), *Multiagent Systems and Applications* (pp. 257–278). Springer Berlin Heidelberg. doi:10.1007/978-3-642-33323-1_10
- Casarotto, S., Cerutti, S., Bianchi, A. M., & Chiarenza, G. A. (2003). Dynamic time warping in the study of ERPs in dyslexic children. In *The 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 2311–2314). Cancun, Mexico: IEEE. doi:10.1109/IEMBS.2003.1280378
- Caselles, V., Catté, F., Coll, T., & Dibos, F. (1993). A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1), 1–31. doi:10.1007/BF01385685
- Caselles, V., Kimmel, R., & Sapiro, G. (1997). Geodesic Active Contours. *International Journal of Computer Vision*, 22(1), 61–79. doi:10.1023/A:1007979827043
- Castel, C. (2008). *La Dyslexie Développementale: Origines et Remédiation*. Université d'Aix-Marseille I.
- Castel, C., Pech-Georgel, C., George, F., & Ziegler, J. C. (2008). Lien entre dénomination rapide et lecture chez les enfants dyslexiques. *L'Année Psychologique*, 108(3), 395–421. doi:10.4074/S0003503308003011
- Castles, A., & Coltheart, M. (1993). Varieties of developmental dyslexia. *Cognition*, 47(2), 149–180. doi:10.1016/0010-0277(93)90003-E
- Cavalieri, S., Garetti, M., Macchi, M., & Taisch, M. (2000). An experimental benchmarking of two multi-agent architectures for production scheduling and control. *Computers in Industry*, 43(2), 139–152. doi:10.1016/S0166-3615(00)00063-4
- Centers for Medicare & Medicaid Services. (2008). *Selecting a development approach*.
- Cerrolaza, J. J., Villanueva, A., & Cabeza, R. (2008). Taxonomic study of polynomial regressions applied to the calibration of video-oculographic systems. In K.-J. Rähä & A. T. Duchowski (Eds.), *Symposium on Eye Tracking Research & Applications (ETRA'08)* (pp. 259–266). Savannah, GA, USA: ACM Press. doi:10.1145/1344471.1344530
- Červenka, R., Trenčanský, I., Calisti, M., & Greenwood, D. (2005). AML: Agent Modeling Language Toward Industry-Grade Agent-Based Modeling. In J. Odell, P. Giorgini, & J. P. Müller (Eds.), *Agent-Oriented Software Engineering V* (pp. 31–46). New York, NY, USA: Springer Berlin Heidelberg. doi:10.1007/978-3-540-30578-1_3
- Cesta, A., & D'Aloisi, D. (1996). Building interfaces as personal agents: a case study. *ACM SIGCHI Bulletin*, 28(3), 108–113. doi:10.1145/231132.231154
- Cha, E., Wood, J., & Finkelstein, J. (2012). Using gaming platforms for telemedicine applications: A cross-platform comparison. In *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI 2012)* (pp. 918–921). Hong Kong: IEEE. doi:10.1109/BHI.2012.6211737
- Chaib-draa, B., & Dignum, F. (2002). Trends in agent communication language. *Computational Intelligence*, 18(2), 89–101. doi:10.1111/1467-8640.00184

- Chan, F. T. S., Zhang, J., & Li, P. (2003). Agent- and CORBA-Based Application Integration Platform for an Agile Manufacturing Environment. *The International Journal of Advanced Manufacturing Technology*, 21(6), 460–468. doi:10.1007/s001700300054
- Chan, T. F., & Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2), 266–277. doi:10.1109/83.902291
- Chen, P. (2002). Entity-relationship modeling: historical events, future trends, and lessons learned. In M. Broy & E. Denert (Eds.), *Software Pioneers* (pp. 296–310). Springer Berlin Heidelberg. doi:10.1007/978-3-642-59412-0_17
- Chen, Q., Hsu, M., Dayal, U., & Griss, M. (1999). *Multi-agent cooperation, dynamic workflow and XML for e-commerce automation*. Palo Alto, CA.
- Chen, Y. (2010). Documenting Transitional Information in EMR. In E. Mynatt, G. Fitzpatrick, S. Hudson, K. Edwards, & T. Rodden (Eds.), *SIGCHI Conference on Human Factors in Computing Systems (CHI'10)* (pp. 1787–1796). Atlanta, GA, USA: ACM Press. doi:10.1145/1753326.1753594
- Chen, Y.-W., & Kubo, K. (2007). A Robust Eye Detection and Tracking Technique Using Gabor Filters. In *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)* (pp. 109–112). Kaohsiung: IEEE. doi:10.1109/IIH-MSP.2007.58
- Cheung, T. (1996). Petri nets for protocol engineering. *Computer Communications*, 19(14), 1250–1257. doi:10.1016/S0140-3664(96)01158-9
- Chevrie-Müller, C., Simon, A. M., Le Normand, M. T., & Fournier, S. (1988). *Batterie d'évaluation psycholinguistique : I. BEPL A. II BEPL B*. Paris: Éditions du centre de psychologie appliquée.
- Chi, J., Zhang, P., Zheng, S., Zhang, C., & Huang, Y. (2009). Key Techniques of Eye Gaze Tracking Based on Pupil Corneal Reflection. In *2009 WRI Global Congress on Intelligent Systems* (pp. 133–138). IEEE. doi:10.1109/GCIS.2009.338
- Choe, J., & Yoo, S. K. (2008). Web-based secure access from multiple patient repositories. *International Journal of Medical Informatics*, 77(4), 242–248. doi:10.1016/j.ijmedinf.2007.06.001
- Christov, S., Chen, B., Avrunin, G. S., Clarke, L. A., Osterweil, L. J., Brown, D., ... Mertens, W. (2008). Formally defining medical processes. *Methods of Information in Medicine*, 47(5), 392–398. doi:10.3414/ME9120
- Ciccarese, P., Caffi, E., Boiocchi, L., Halevy, A., Quaglini, S., Kumar, A., & Stefanelli, M. (2003). The NewGuide Project: Guidelines, Information Sharing and Learning from Exceptions. In M. Dojat, E. T. Keravnou, & P. Barahona (Eds.), *Artificial Intelligence in Medicine* (pp. 163–167). Protaras, Cyprus: Springer Berlin Heidelberg. doi:10.1007/978-3-540-39907-0_23
- Ciccarese, P., Caffi, E., Quaglini, S., & Stefanelli, M. (2005). Architectures and tools for innovative Health Information Systems: The Guide Project. *International Journal of Medical Informatics*, 74(7-8), 553–562. doi:10.1016/j.ijmedinf.2005.02.001

- Ciccarese, P., Kumar, A., & Quaglini, S. (2002). NEW-GUIDE: a new approach to representing clinical practice guidelines. In *Advances in Clinical Knowledge Management* (pp. 15–18). London, UK.
- Clarity Informatics. (2014). Prodigy. Retrieved from <http://prodigy.clarity.co.uk/>
- Clay, A., Couture, N., & Nigay, L. (2009). eMotion: un outil pour personnaliser la reconnaissance d'émotions. In *Ergonomie et Informatique Avancée Conférence 2010 (Ergo'IA '10)* (pp. 59–66). ACM. doi:10.1145/1868650.1868660
- Clayton, P. D., Pryor, T. A., Wigertz, O. B., & Hripcsak, G. (1989). Issues and Structures for Sharing Medical Knowledge among Decision-Making Systems: The 1989 Arden Homestead Retreat. In L. C. Kingsland (Ed.), *Annual Symposium on Computer Application in Medical Care* (pp. 116–121). Washington, DC: IEEE.
- Clifton, C., Staub, A., & Rayner, K. (2007). Eye Movements in Reading Words and Sentences. In R. P. G. van Gompel, M. H. Fischer, W. S. Murray, & R. L. Hill (Eds.), *Eye movements: A window on mind and brain* (pp. 341–372). Elsevier Ltd. doi:10.1016/B978-008044980-7/50017-3
- Clinical Data Interchange Standards Consortium. (2011). Study/Trial Design Model | CDISC. Retrieved from <http://www.cdisc.org/study-trial-design>
- Clinical Data Interchange Standards Consortium. (2015). CDISC | Strength Through Collaboration. Retrieved from <http://www.cdisc.org/>
- C-Log International. (2014). Workey Designer. C-Log International. Retrieved from <http://www.c-log.com/modelisation/>
- Cohen, L. D. (1991). On Active Contour Models and Balloons. *CVGIP: Image Understanding*, 53(2), 211–218. doi:10.1016/1049-9660(91)90028-N
- Cohen, P. R., Cheyer, A., Wang, M., & Baeg, S. C. (1994). An open agent architecture. In O. Etzioni (Ed.), *AAAI 1994 Spring Symposium: Technical Report SS-94-03* (pp. 1–8). AAAI Press.
- Coiera, E. (2001). Mediated agent interaction. In S. Quaglini, P. Barahona, & S. Andreassen (Eds.), *Artificial Intelligence in Medicine* (pp. 1–15). Springer Berlin Heidelberg. doi:10.1007/3-540-48229-6_1
- Collins, C., Hewson, D. L., Munger, R., & Wade, T. (2010). *Evolving models of behavioral health integration in primary care*. New York.
- Colmerauer, A., & Roussel, P. (1996). The birth of Prolog. *ACM SIGPLAN Notices*, 28(3), 37–52. doi:10.1145/155360.155362
- Coltheart, M., Rastle, K., Perry, C., Langdon, R., & Ziegler, J. (2001, January). DRC: a dual route cascaded model of visual word recognition and reading aloud. *Psychological Review*. doi:10.1037/0033-295X.108.1.204
- Comité Européen de Normalisation. (2014). European Committee for Standardization. Retrieved from <http://www.cen.eu>
- Conn, N., & McTear, M. (2000). Speech technology: A solution for people with disabilities. In *Seminar on Speech and Language Processing for Disabled and Elderly People* (pp. 7/1–7/6). IEEE. doi:10.1049/ic:20000137

- Cooperative State University Karlsruhe. (2014). Workflow Petri Net Designer (WoPeD). Cooperative State University Karlsruhe. Retrieved from <http://woped.org/>
- Coriat, M. (1993). Stepwise development of correct agents: a behavioural approach based on colored Petri nets. In *Electrical and Computer Engineering* (pp. 489–492). Vancouver, BC: IEEE. doi:10.1109/CCECE.1993.332346
- Corkill, D. D. (2003). Collaborating software: Blackboard and multi-agent systems & the future. In *International Lisp Conference* (pp. 1–12). New York, New York, USA.
- Corriveau, K. H., & Goswami, U. (2009). Rhythmic motor entrainment in children with speech and language impairments: Tapping to the beat. *Cortex*, 45(1), 119–130. doi:10.1016/j.cortex.2007.09.008
- Cortés, U., Annicchiarico, R., & Urdiales, C. (2008). Agents and Healthcare: Usability and Acceptance. In R. Annicchiarico, U. Cortés, & C. Urdiales (Eds.), *Agent Technology and e-Health* (pp. 1–4). Birkhäuser Basel. doi:10.1007/978-3-7643-8547-7_1
- COSSAC. (2012). Tallis. Retrieved from <http://www.cossac.org/tallis>
- Cossentino, M. (2002). Different perspectives in designing multi-agent systems. In *Agent Technology and Software Engineering Workshop at (NODE'02)* (pp. 1–13). Erfurt, Germany.
- Cossentino, M., & Potts, C. (2002). A CASE tool supported methodology for the design of multi-agent systems. In *2002 International Conference on Software Engineering Research and Practice (SERP'02)* (pp. 295–306). Las Vegas, NV, USA.
- Costa, M., Zavaleta, J., Serra da Cruz, S. M., Manhães, M., Cerceau, R., Carvalho, L. A., & Mousinho, R. (2013). A Computational Approach for Screening Dyslexia. In *26th International Symposium on Computer-Based Medical Systems (CBMS'13)* (pp. 565–566). Porto: IEEE. doi:10.1109/CBMS.2013.6627879
- Coupric, M., & Bertrand, G. (2009). New characterizations of simple points in 2D, 3D, and 4D discrete spaces. *Pattern Analysis and Machine Intelligence*, 31(4), 637–648. doi:10.1109/TPAMI.2008.117
- Coupric, M., & Bertrand, G. (2012). Discrete Topological Transformations for Image Processing. In V. E. Brimkov & R. P. Barneva (Eds.), *Digital Geometry Algorithms* (Vol. 2, pp. 73–107). Springer Netherlands. doi:10.1007/978-94-007-4174-4_3
- Coutaz, J. (1987). PAC, an object oriented model for dialog design. In H.-J. Bullinger & B. Shackel. (Eds.), *Human-Computer Interaction (Interact'87)* (pp. 431–436). Stuttgart, Federal Republic of Germany: Elsevier B.V.
- Coutaz, J., & Balbo, S. (1991). Applications: A dimension space for user interface management systems. In S. P. Robertson, G. M. Olson, & J. S. Olson (Eds.), *SIGCHI Conference on Human Factors in Computing Systems (CHI'91)* (pp. 27–32). New Orleans, LA, USA: ACM Press. doi:10.1145/108844.108849
- Coutaz, J., Nigay, L., & Salber, D. (1996). Agent-based architecture modelling for interactive systems. In D. Benyon & P. Palanque (Eds.), *Critical Issues in User Interface Systems Engineering* (pp. 191–209). Springer London. doi:10.1007/978-1-4471-1001-9_11

- Coutinho, F. L., & Morimoto, C. H. (2006). Free head motion eye gaze tracking using a single camera and multiple light sources. In *2006 19th Brazilian Symposium on Computer Graphics and Image Processing* (pp. 171–178). IEEE. doi:10.1109/SIBGRAPI.2006.21
- Cragin, M. H., & Shankar, K. (2006). Scientific Data Collections and Distributed Collective Practice. *Computer Supported Cooperative Work*, 15(2-3), 185–204. doi:10.1007/s10606-006-9018-z
- Crane, H. D., & Steele, C. M. (1985). Generation-V dual-Purkinje-image eyetracker. *Applied Optics*, 24(4), 527–537. doi:10.1364/AO.24.000527
- Creative Technology Ltd. (2015). Creative Senz3D - Webcams - Creative Labs (France). Retrieved from <http://fr.creative.com/p/web-cameras/creative-senz3d>
- Cresswell, I., Monteith-Hodge, D., & Winfield, M. (1997). An approach to the diagnosis and remediation of developmental dyslexia using microcomputers. In *Computers in the Service of Mankind: Helping the Disabled* (pp. 12/1 – 12/5). IEEE. doi:10.1049/ic:19970641
- Cruz-Correia, R. J., Vieira-Marques, P. M., Costa, P., Ferreira, A., Oliveira-Palhares, E., Araújo, F., & Costa-Pereira, A. (2005). Integration of hospital data using agent technologies - a case study. *AI Communications*, 18(3), 191–200.
- Cruz-Neira, C., Sandin, D. J., & DeFanti, T. A. (1993). Surround screen projection-based virtual reality: The design and implementation of the CAVE. In *The 20th annual conference on Computer graphics and interactive techniques (SIGGRAPH'93)* (pp. 135–142). ACM Press. doi:10.1145/166117.166134
- Cuba Gyllensten, O. (2014). *Evaluation of classification algorithms for smooth pursuit eye movements: Evaluating current algorithms for smooth pursuit detection on Tobii Eye Trackers*. KTH Royal Institute of Technology (Stockholm).
- Cubizolle, S. (2012). *Analyse des saccades et de la poursuite oculaire dans les syndromes parkinsoniens : Mise en place d'un outil diagnostique et de recherche au CHU de Bordeaux*. Université de Bordeaux II.
- Czibula, G., Czibula, I. G., Cojocar, G. S., & Guran, A. M. (2008). IMASC - An Intelligent MultiAgent System for Clinical Decision Support. In *2008 First International Conference on Complexity and Intelligence of the Artificial and Natural Complex Systems, Medical Applications of the Complex Systems, Biomedical Computing* (pp. 185–190). Targu Mures, Mures: IEEE. doi:10.1109/CANS.2008.28
- Da Silva, J. L. T., & Demazeau, Y. (2002). Vowels co-ordination model. *Les Cahiers Du Laboratoire Leibniz*, 52, 1–21.
- Da Silva, V. T., Choren, R., & de Lucena, C. J. P. (2004). A UML based approach for modeling and implementing multi-agent systems. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)* (pp. 914–921). New York, New York, USA: IEEE. doi:10.1109/AAMAS.2004.36
- Da Silva, V. T., & de Lucena, C. J. P. (2004). From a conceptual framework for agents and objects to a multi-agent system modeling language. *Autonomous Agents and Multi-Agent Systems*, 9(1-2), 145–189. doi:10.1023/B:AGNT.0000019691.42633.07

- Dadam, P., Reichert, M., & Kuhn, K. (2000). Clinical workflows — the killer application for process-oriented information systems? In W. Abramowicz & M. E. Orłowska (Eds.), *4th International Conference on Business Information Systems (BIS 2000)* (pp. 36–59). Poznań, Poland: Springer-Verlag London. doi:10.1007/978-1-4471-0761-3_3
- Danon-Boileau, L., & Barbier, D. (2006). Play-On. *Ortho Magazine*, 12(66), 37.
- Dart, T., Xu, Y., Chatellier, G., & Degoulet, P. (2001). Computerization of Guidelines: Towards a “Guideline Markup Language.” In V. L. Patel, R. Rogers, & R. Haux (Eds.), *MEDINFO 2001: Proceedings of the 10th World Congress on Medical Informatics* (pp. 186–190). IOS Press. doi:10.3233/978-1-60750-928-8-186
- Das, S. K., Fox, J., Elsdon, D., & Hammond, P. (1997). A flexible architecture for autonomous agents. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(4), 407–440. doi:10.1080/095281397146979
- Dastani, M., Hindriks, K. V., & Meyer, J.-J. C. (2010). *Specification and Verification of Multi-agent Systems*. (M. Dastani, K. V. Hindriks, & J.-J. C. Meyer, Eds.). Springer US. doi:10.1007/978-1-4419-6984-2
- Davis, R. (2008). *ARIS Design Platform: Advanced Process Modelling and Administration*. (R. Davis, Ed.). Springer London. doi:10.1007/978-1-84800-111-4
- Davis, R., & Brabänder, E. (2007). *ARIS Design Platform: Getting Started with BPM*. (R. Davis & E. Brabänder, Eds.). Springer London. doi:10.1007/978-1-84628-613-1
- Dayal, U., Hsu, M., & Ladin, R. (2001). Business Process Coordination: State of the Art, Trends, and Open Issues. In P. M. G. Apers, P. Atzeni, S. Ceri, S. Paraboschi, K. Ramamohanarao, & R. T. Snodgrass (Eds.), *27th International Conference on Very Large Data Bases (VLDB'01)* (pp. 3–13). Roma, Italy: Morgan Kaufmann Publishers Inc. doi:10.1.1.21.7413&rep=rep1&type=pdf
- De Clercq, P. A., Blom, J. A., Korsten, H. H. M., & Hasman, A. (2004). Approaches for creating computer-interpretable guidelines that facilitate decision support. *Artificial Intelligence in Medicine*, 31(1), 1–27. doi:10.1016/j.artmed.2004.02.003
- De Clercq, P. A., & Hasman, A. (2004). Experiences with the development, implementation and evaluation of automated decision support systems. In M. Fieschi, E. Coiera, & Y.-C. J. Li (Eds.), *MEDINFO 2004: Proceedings of the 11th World Congress on Medical Informatics* (pp. 1033–1037). IOS Press. doi:10.3233/978-1-60750-949-3-1033
- De Clercq, P. A., Hasman, A., Blom, J. A., & Korsten, H. H. M. (2001). Design and implementation of a framework to support the development of clinical guidelines. *International Journal of Medical Informatics*, 64(2-3), 285–318. doi:10.1016/S1386-5056(01)00189-7
- De Loor, P., Le Bodic, L., Calvet, G., & Tisseau, J. (2006). Un simulateur d’usage pour l’évaluation des systèmes interactifs multimodaux. *Revue d’Interaction Homme-Machine*, 7(1), 59–91.
- De Montaigne, M. (1588). Chapitre XIII : De l’expérience. In *Essais (Volume III)*.
- De Paor, D. G. (1988). Strain determination from three known stretches—an exact solution. *Journal of Structural Geology*, 10(6), 639–642. doi:10.1016/0191-8141(88)90029-6

- De Santis, A., & Iacoviello, D. (2006). Optimal segmentation of pupillometric images for estimating pupil shape parameters. *Computer Methods and Programs in Biomedicine*, 84(2-3), 174–87. doi:10.1016/j.cmpb.2006.07.005
- De Santis, A., & Iacoviello, D. (2007). A discrete level set approach to image segmentation. *Signal, Image and Video Processing*, 1(4), 303–320. doi:10.1007/s11760-007-0032-5
- De Santis, A., & Iacoviello, D. (2009a). A Robust Eye Tracking Procedure for Medical and Industrial Applications. In J. M. R. S. Tavares & R. M. N. Jorge (Eds.), *Advances in Computational Vision and Medical Image Processing* (Vol. 13, pp. 173–185). Springer Netherlands.
- De Santis, A., & Iacoviello, D. (2009b). Robust real time eye tracking for computer interface for disabled people. *Computer Methods and Programs in Biomedicine*, 96(1), 1–11. doi:10.1016/j.cmpb.2009.03.010
- Defense Advanced Research Projects Agency. (1993). Knowledge Sharing Effort. Retrieved from www.cs.umbc.edu/kse
- Dejerine, J. (1891). Sur un cas de cécité verbale avec agraphie suivi d'autopsie. *Mémoires de La Société de Biologie*, 3, 197–201.
- Dejonckère, L. (2013). *Les adaptations pédagogiques dans l'apprentissage de la lecture pour l'élève dyslexique, en CM2, en mettant l'accent sur la prise en charge des difficultés de compréhension*. Université d'Artois.
- Delamare, C. (2012). *Dyslexie-dysorthographe et handicap scolaire: les outils informatiques de compensation du langage écrit et leur intégration dans la prise en charge en orthophonie. Réalisation d'une plaquette d'information et d'un site Internet à l'usage des orthophonis*. Université de Poitiers.
- DeLoach, S. A., Wood, M. F., & Sparkman, C. H. (2001). Multiagent systems engineering. *International Journal of Software Engineering and Knowledge Engineering*, 11(3), 231–258. doi:10.1142/S0218194001000542
- Delpierre, C., Cuzin, L., Fillaux, J., Alvarez, M., Massip, P., & Lang, T. (2004). A systematic review of computer-based patient record systems and quality of care: more randomized clinical trials or a broader approach? *International Journal for Quality in Health Care*, 16(5), 407–416. doi:10.1093/intqhc/mzh064
- Demazeau, Y. (1995). From Interactions to Collective Behaviour in Agent-Based Systems. In *1st. European Conference on Cognitive Science* (pp. 117–132). Saint-Malo, France.
- Demeyer, S., Meijler, T. D., Nierstrasz, O., & Steyaert, P. (1997). Design guidelines for “tailorable” frameworks. *Communications of the ACM*, 40(10), 60–64. doi:10.1145/262793.262805
- Démonet, J.-F., Taylor, M. J., & Chaix, Y. (2004). Developmental dyslexia. *The Lancet*, 363(9419), 1451–1460. doi:DOI: 10.1016/S0140-6736(04)16106-0
- Dernoncourt, F. (2012). Replacing the computer mouse. In *MIT CSAIL Student Workshop* (pp. 1–9). Boston, MA, USA.
- DFKI GmbH. (1998). CoMMA-MAPs. Retrieved from http://www.lt-world.org/kb/players-and-teams/projects/obj_61457

- Dieste, O., Juristo, N., & Shull, F. (2008). Understanding the customer: what do we know about requirements elicitation? *IEEE Software*, 25(2), 11–13. doi:10.1109/MS.2008.53
- Dillon, D. S., Dillon, T. S., & Chang, E. (2008). Using UML 2.1 to Model Multi-agent Systems. In U. Brinkschulte, T. Givargis, & S. Russo (Eds.), *Software Technologies for Embedded and Ubiquitous Systems* (pp. 1–8). Anacardi, Capri Island, Italy: Springer Berlin Heidelberg. doi:10.1007/978-3-540-87785-1_1
- Djaouti, D. (2011). *Serious Game Design : Considérations théoriques et techniques sur la création de jeux vidéo à vocation utilitaire*. Université Toulouse III Paul Sabatier.
- Doherty, W. J. (1995). The why's and levels of collaborative family health care. *Family Systems Medicine*, 13(3-4), 275–281. doi:10.1037/h0089174
- Dorador, J. M., & Young, R. I. . M. (2000). Application of IDEF0, IDEF3 and UML methodologies in the creation of information models. *International Journal of Computer Integrated Manufacturing*, 13(5), 430–445. doi:10.1080/09511920050117928
- DotNetNuke Corporation. (2012). DotNetNuke. Retrieved from <http://www.dnnsoftware.com/>
- Dropbox Inc. (2014). DropBox. Retrieved from <https://www.dropbox.com/>
- Dube, K., Mansour, E., & Wu, B. (2005). Supporting collaboration and information sharing in computer-based clinical guideline management. In *18th IEEE Symposium on Computer-Based Medical Systems* (pp. 232–237). Dublin, Ireland: IEEE. doi:10.1109/CBMS.2005.95
- Dube, K., & Wu, B. (2009). A generic approach to computer-based Clinical Practice Guideline management using the ECA Rule paradigm and active databases. *International Journal of Technology Management*, 47(1/2/3), 75–95. doi:10.1504/IJTM.2009.024115
- Dubey, A. K., & Chueh, H. (1998). Using the extensible markup language (XML) in automated clinical practice guidelines. In *AMIA Annual Symposium Proceedings* (pp. 735–739). American Medical Informatics Association.
- Duchowski, A. T. (2002). A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments, & Computers : A Journal of the Psychonomic Society, Inc*, 34(4), 455–70.
- Duchowski, A. T. (2007). *Eye tracking methodology: Theory and Practice*. (A. T. Duchowski, Ed.) (2nd ed.). Springer London. doi:10.1007/978-1-84628-609-4
- Duffy, F. D., Gordon, G. H., Whelan, G., Cole-Kelly, K., Frankel, R., Buffone, N., ... Langdon, L. (2004). Assessing competence in communication and interpersonal skills: the Kalamazoo II report. *Academic Medicine : Journal of the Association of American Medical Colleges*, 79(6), 495–507.
- Dumas, P., & Charbonnel, G. (1990). *La méthode OSSAD : pour maîtriser les technologies de l'information (Tome 1 : Principes)*. Les Éditions d'Organisation.
- Durkin, K. (2010). Videogames and young people with developmental disorders. *Review of General Psychology*, 14(2), 122–140. doi:10.1037/a0019438
- E=mc dys. (2013). Site web de E=mc dys, l'association pour la promotion d'un réseau de santé Dys/10 concernant les Troubles Spécifiques d'Apprentissage. Retrieved from <http://www.emcdys.fr/>

- EBSCO Industries Inc. (2015). EBSCOhost Online Research Databases. Retrieved from <http://www.ebscohost.com/>
- Ecma International. (1966). *Standard ECMA-4: Flow Charts*.
- Eden, G. F., Jones, K. M., Cappell, K., Gareau, L., Wood, F. B., Zeffiro, T. A., ... Flowers, D. L. (2004). Neural changes following remediation in adult developmental dyslexia. *Neuron*, 44(3), 411–422. doi:10.1016/j.neuron.2004.10.019
- EdrawSoft. (2014). IDEF0 Diagram Software - Create IDEF0 diagrams rapidly with examples and templates. Retrieved from <http://www.edrawsoft.com/IDEF0-flowcharts.php>
- Ehrler, L., Fleurke, M., Purvis, M., & Savarimuthu, B. T. R. (2005). Agent-based Workflow Management Systems (WfMSs): JBees – a distributed and adaptive WfMS with monitoring and controlling capabilities. *Journal of Information Systems and E-Business Management*, 4(1), 5–23. doi:10.1007/s10257-005-0010-9
- Eichmann, M., & Luessi, M. (2005). *Efficient multilevel image thresholding*. Hochschule für Technik Rapperswil, Switzerland.
- Elkin, P. L., Brown, S. H., & Chute, C. G. (2001). Guideline for Health Informatics: Controlled Health Vocabularies - Vocabulary Structure and High-level Indicators. In V. L. Patel, R. Rogers, & R. Haux (Eds.), *MEDINFO 2001: Proceedings of the 10th World Congress on Medical Informatics* (pp. 191–195). IOS Press. doi:10.3233/978-1-60750-928-8-191
- Ellingsen, G., & Røed, K. (2010). The Role of Integration in Health-Based Information Infrastructures. *Computer Supported Cooperative Work (CSCW)*, 19(6), 557–584. doi:10.1007/s10606-010-9122-y
- Elsevier B.V. (2015). ScienceDirect.com. Retrieved from <http://www.sciencedirect.com/>
- Emotiv Inc. (2013). Emotic SDK. Emotiv Inc.
- Emotiv Inc. (2015). Emotiv's Web site. Retrieved from <http://emotiv.com/>
- Encyclopædia Universalis. (2015). Encyclopædia Universalis France. Retrieved from www.universalis.fr
- Esteva, M., Rosell, B., Rodriguez-Aguilar, J. A., & Arcos, J. L. (2004). AMELI: an agent-based middleware for electronic institutions. In *The Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)* (pp. 236–243). New York, New York, USA: IEEE. doi:10.1109/AAMAS.2004.56
- Evolus. (2013). Pencil. Evolus. Retrieved from <http://pencil.evolus.vn/>
- Facoetti, A., Trussardi, A. N., Ruffino, M., Lorusso, M. L., Cattaneo, C., Galli, R., ... Zorzi, M. (2010). Multisensory spatial attention deficits are predictive of phonological decoding skills in developmental dyslexia. *Journal of Cognitive Neuroscience*, 22(5), 1011–25. doi:10.1162/jocn.2009.21232
- Faulkner, S., Kolp, M., Nguyen, T., Coyette, A., & Do, T. T. (2004). Information Integration Architecture Development: A Multi-Agent Approach. In F. Maurer & G. Ruhe (Eds.), *16th International Conference on Software Engineering and Knowledge Engineering (SEKE'04)* (pp. 192–198). Banff, Canada.

- Faure-Brac, C., Gombert, A., & Roussey, J.-Y. (2012). Les enseignants du secondaire et les élèves porteurs de troubles spécifiques du langage écrit. *Le Français Aujourd'hui*, 177(2), 65–78. doi:10.3917/lfa.177.0065
- Fayad, M. E., & Schmidt, D. C. (1997). Object-oriented application frameworks. *Communications of the ACM*, 40(10), 32–38. doi:10.1145/262793.262798
- Ferber, J. (1995). *Les systèmes multi-agents : vers une intelligence collective* (Informatiq.). InterEditions.
- Ferber, J., & Gutknecht, O. (1998). A meta-model for the analysis and design of organizations in multi-agent systems. In *International Conference on Multi Agent Systems* (pp. 128–135). Paris, France: IEEE. doi:10.1109/ICMAS.1998.699041
- Ferber, J., Gutknecht, O., & Michel, F. (2004). From Agents to Organizations: An Organizational View of Multi-agent Systems. In P. Giorgini, J. P. Müller, & J. Odell (Eds.), *Agent-Oriented Software Engineering IV* (pp. 214–230). Melbourne, Australia: Springer Berlin Heidelberg. doi:10.1007/978-3-540-24620-6_15
- Ferrara, F. M. (1996). The middleware-based architectural approach for opening and evolving healthcare information systems. In J. Brender, J. P. Christensen, J.-R. Scherrer, & P. McNair (Eds.), *Medical Informatics Europe '96* (pp. 264–270). IOS Press. doi:10.3233/978-1-60750-878-6-264
- Ferré, X., Juristo, N., Windl, H., & Constantine, L. (2001). Usability basics for software developers. *IEEE Software*, 18(1), 22–29. doi:10.1109/52.903160
- Ferreira, A., Barreto, L., Brandão, P., Correia, R., Sargento, S., & Antunes, L. (2009). Accessing an existing virtual electronic patient record with a secure wireless architecture. In P. Olla & J. Tan (Eds.), *Mobile Health Solutions for Biomedical Applications* (pp. 24–44). IGI Global. doi:10.4018/978-1-60566-332-6.ch002
- Figueira, M. V., de Azevedo, D. F. G., Russomano, T., & Lilienthal, R. F. (2008). Evaluation tests for eye tracking systems. In *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 5765–5768). Vancouver, British Columbia, Canada: IEEE. doi:10.1109/IEMBS.2008.4650524
- Finin, T., Fritzon, R., McKay, D., & McEntire, R. (1994). KQML as an agent communication language. In *Third international conference on Information and knowledge management* (pp. 456–463). New York, NY, USA: ACM Press. doi:10.1145/191246.191322
- Finin, T., Weber, J., Widerhold, G., Genesereth, M., Fritzon, R., McKay, D., ... Beck, C. (1993). *DRAFT specification of the KQML agent-communication language* (Vol. 5). Palo Alto, CA.
- FIPA. (2002a). *FIPA Abstract Architecture Specification (SC00001L)*. Geneva, Switzerland.
- FIPA. (2002b). *FIPA ACL Message Structure Specification (SC00061G)*. Geneva, Switzerland.
- FIPA. (2002c). *FIPA Brokering Interaction Protocol Specification (SC00033H)*. Geneva, Switzerland.
- FIPA. (2002d). *FIPA SL Content Language Specification (SC00008I)*. Geneva, Switzerland.
- FIPA. (2004). *FIPA Agent Management Specification (SC00023K)*. Geneva, Switzerland.

- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395. doi:10.1145/358669.358692
- Fitzgibbon, A. W., Pilu, M., & Fisher, R. B. (1996). Direct Least Squares Fitting of Ellipses. In *The 13th International Conference on Pattern Recognition* (pp. 253–257). Vienna: IEEE. doi:10.1109/ICPR.1996.546029
- Fitzgibbon, A. W., Pilu, M., & Fisher, R. B. (1999). Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5), 476–480. doi:10.1109/34.765658
- Fonseca i Casas, P. (2014). Using Specification and Description Language to Formalize Multiagent Systems. *Applied Artificial Intelligence*, 28(5), 504–531. doi:10.1080/08839514.2014.905820
- Forslund, D. W., Phillips, R. L., Kilman, D. G., & Cook, J. L. (1996). Experiences with a distributed virtual patient record system. In *Conference of the American Medical Informatics Association / AMIA Annual Fall Symposium* (pp. 483–487). American Medical Informatics Association.
- Fortune, J., & Valerdi, R. (2013). A framework for reusing systems engineering products. *Systems Engineering*, 16(3), 304–312. doi:10.1002/sys.21232
- Fox, J. (2002). Quality and safety of clinical decision support technologies: a discussion of the role of formal methods. In H. Ehrig, B. J. Kramer, & A. Ertz (Eds.), *World Conference on Integrated Design and Process Technology (IDPT-2002)* (pp. 1–8). Society for Design and Process Science.
- Fox, J., Beveridge, M., & Glasspool, D. (2003). Understanding intelligent agents: analysis and synthesis. *AI Communications*, 16(3), 139–152.
- Fox, J., Johns, N., Lyons, C., Rahmanzadeh, A., Thomson, R., & Wilson, P. (1997). PROforma: a general technology for clinical decision support systems. *Computer Methods and Programs in Biomedicine*, 54(1-2), 59–67. doi:10.1016/S0169-2607(97)00034-5
- Fox, J., Johns, N., & Rahmanzadeh, A. (1998). Disseminating medical knowledge: the PROforma approach. *Artificial Intelligence in Medicine*, 14(1-2), 157–181. doi:10.1016/S0933-3657(98)00021-9
- Fox, J., Johns, N., Rahmanzadeh, A., & Thomson, R. (1996). PROforma: A method and language for specifying clinical guidelines and protocols. In *Medical Informatics Europe '96* (Studies in., pp. 516–520). IOS Press. doi:10.3233/978-1-60750-878-6-516
- Framinan, J. M., Parra, C. L., Montes, M., & Pérez, P. (2005). Collaborative Healthcare Process Modelling: A Case Study. In L. M. Camarinha-Matos, H. Afsarmanesh, & A. Ortiz (Eds.), *Collaborative Networks and Their Breeding Environments* (pp. 395–402). Springer US. doi:10.1007/0-387-29360-4_42
- Frank, A. (2001). Agent protocols for integration and cooperation in a design application framework. In Z. Maamar, W. Shen, & H. H. Ghenniwa (Eds.), *The 2nd International Workshop on Software Agents and Workflow for Systems Interoperability* (pp. 31–38). London, Canada: NRC Research Council Canada.

- Frankel, R., Altschuler, A., George, S., Kinsman, J., Jimison, H., Robertson, N. R., & Hsu, J. (2005). Effects of exam-room computing on clinician-patient communication: a longitudinal qualitative study. *Journal of General Internal Medicine*, 20(8), 677–682. doi:10.1111/j.1525-1497.2005.0163.x
- Franklin, S., & Graesser, A. (1997). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In S. Franklin & A. Graesser (Eds.), *Intelligent Agents III Agent Theories, Architectures, and Languages* (pp. 21–35). Budapest, Hungary: Springer Berlin Heidelberg. doi:10.1007/BFb0013570
- Fridsma, D. B., Gennari, J. H., & Musen, M. A. (1996). Making generic guidelines site-specific. In *AMIA Annual Symposium Proceedings* (pp. 597–601). American Medical Informatics Association.
- Funk, M., & Ivbijaro, G. (2008). *Integrating mental health into primary care: A global perspective*.
- Gabay, J., & Gabay, D. (2008). *UML 2 : Analyse et conception*. Dunod.
- Galaburda, A. M. (1985). Norman Geschwind: 1926-1984. *Neuropsychologia*, 23(3), 297–304. doi:10.1016/0028-3932(85)90016-8
- Galaburda, A. M. (1993). Neurology of developmental dyslexia. *Current Opinion in Neurobiology*, 3(2), 237–242. doi:10.1016/0959-4388(93)90216-L
- Galaburda, A. M., Sherman, G. F., Rosen, G. D., Aboitiz, F., & Geschwind, N. (1985). Developmental dyslexia: four consecutive patients with cortical anomalies. *Annals of Neurology*, 18(2), 222–33. doi:10.1002/ana.410180210
- Galland, S., Gaud, N., Rodriguez, S., Balbo, F., Picard, G., & Boissier, O. (2014). Contextualiser l'interaction entre agents en combinant dimensions sociale et physique au sein de l'environnement. In R. Courdier & J.-P. Jamont (Eds.), *22èmes Journées Francophones sur les Systèmes Multi-Agents (JFSMA 2014)* (pp. 65–74). Lorient-sur-Drôme, France: Cépaduès Éditions.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design patterns: elements of reusable object-oriented software*. Addison-Wesley. doi:10.1093/carcin/bgs084
- Gangat, Y., Payet, D., & Courdier, R. (2012). Methodology for a new agent architecture based on the MVC pattern. In *Artificial Intelligence: Methodology, Systems, and Applications* (Vol. 7557, pp. 230–239). Springer Berlin Heidelberg. doi:10.1007/978-3-642-33185-5_26
- Garcia, G., & Barra, V. (2015). Evaluating the simple point criterion in 2D using a parsimonious algorithm. *Pattern Recognition Letters*.
- Garcia-Ojeda, J. C., DeLoach, S. A., & Robby. (2009a). agentTool III: from process definition to code generation. In *8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'09)* (pp. 1393–1394). Budapest, Hungary: International Foundation for Autonomous Agents and Multiagent Systems.
- Garcia-Ojeda, J. C., DeLoach, S. A., & Robby. (2009b). agentTool process editor: supporting the design of tailored agent-based processes. In S. Y. Shin & S. Ossowski (Eds.), *The 2009 ACM symposium on Applied Computing* (pp. 707–714). Honolulu, HI, USA: ACM Press. doi:10.1145/1529282.1529430

- Garcia-Ojeda, J. C., DeLoach, S. A., Robby, Oyenon, W. H., & Valenzuela, J. (2008). O-MaSE: a customizable approach to developing multiagent development processes. In M. Luck & L. Padgham (Eds.), *Agent-Oriented Software Engineering VIII* (pp. 1–15). Honolulu, HI, USA: Springer Berlin Heidelberg. doi:10.1007/978-3-540-79488-2_1
- Garg, A. X., Adhikari, N. K. J., McDonald, H., Rosas-Arellano, M. P., Devereaux, P. J., Beyene, J., ... Haynes, R. B. (2005). Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review. *Journal of the American Medical Association*, 293(10), 1223–1238. doi:10.1001/jama.293.10.1223
- Garlan, D., Allen, R., & Ockerbloom, J. M. (2009). Architectural mismatch: Why reuse is still so hard. *IEEE Software*, 26(4), 66–69. doi:10.1109/MS.2009.86
- Gavrilova, M. L., & Monwar, M. (2011). Current Trends in Multimodal Biometric System—Rank Level Fusion. In P. S. P. Wang (Ed.), *Pattern Recognition, Machine Intelligence and Biometrics* (pp. 657–673). Springer Berlin Heidelberg. doi:10.1007/978-3-642-22407-2_25
- Gee, J. P. (2008). Learning and Games. *The Ecology Of Games: Connecting Youth, Games, and Learning*, 33(10), 21–40. doi:10.1162/dmal.9780262693646.021
- Genesereth, M. R. (1997). An agent-based framework for interoperability. In J. M. Bradshaw (Ed.), *Software agents* (pp. 317–345). MIT Press.
- Genesereth, M. R., & Fikes, R. E. (1992). *Knowledge Interchange Format Version 3.0: Reference Manual*. Stanford, California.
- Genesereth, M. R., & Ketchpel, S. P. (1994). Software agents. *Communications of the ACM*, 37(7), 48–53. doi:10.1145/176789.176794
- Gerber, M. S., Brown, D. E., & Harrison, J. H. (2012). *Automatic Formalization of Clinical Practice Guidelines*.
- Geschwind, N. (2010). Disconnexion Syndromes in Animals and Man: Part I. *Neuropsychology Review*, 20(2), 128–157. doi:10.1007/s11065-010-9131-0
- Geschwind, N., & Levitsky, W. (1968). Human brain: left-right asymmetries in temporal speech region. *Science*, 161(3837), 186–187. doi:10.1126/science.161.3837.186
- Giaglis, G. M. (2001). A taxonomy of business process modeling and information systems modeling techniques. *International Journal of Flexible Manufacturing Systems*, 13(2), 209–228. doi:10.1023/A:1011139719773
- Gillois, P., Chatellier, G., Jaulent, M. C., Colombet, I., Fieschi, M., & Degoulet, P. (2001). From Paper-Based to Electronic Guidelines: Application to French Guidelines. In V. L. Patel, R. Rogers, & R. Haux (Eds.), *MEDINFO 2001: Proceedings of the 10th World Congress on Medical Informatics* (Vol. 84, pp. 196–200). IOS Press. doi:10.3233/978-1-60750-928-8-196
- Giorgini, P., & Henderson-Sellers, B. (2005). Agent-Oriented Methodologies: an introduction. In B. Henderson-Sellers & P. Giorgini (Eds.), *Agent-Oriented Methodologies* (pp. 1–19). IGP. doi:10.4018/978-1-59140-581-8.ch001
- Giraud, K., Trébuchon-DaFonseca, A., Démonet, J. F., Habib, M., & Liégeois-Chauvel, C. (2008). Asymmetry of voice onset time-processing in adult developmental dyslexics.

Clinical Neurophysiology: Official Journal of the International Federation of Clinical Neurophysiology, 119(7), 1652–1663. doi:10.1016/j.clinph.2008.02.017

- Giret, A., & Botti, V. (2004). Holons and agents. *Journal of Intelligent Manufacturing*, 15(5), 645–659. doi:10.1023/B:JIMS.0000037714.56201.a3
- Glasspool, D. W., Fox, J., Oettinger, A., & Smith-spark, J. (2006). Argumentation in Decision Support for Medical Care Planning for Patients and Clinicians. In *AAAI Spring Symposium: Argumentation for Consumers of Healthcare* (pp. 58–63). AAAI Press.
- Gleizes, M., & Glize, P. (2002). ABROSE: Multi Agent Systems for Adaptive Brokerage. In P. Giorgini, Y. Lespérance, G. Wagner, & E. Yu (Eds.), *Fourth International Bi-Conference Workshop on AGENT-ORIENTED INFORMATION SYSTEMS (AOIS-2002 at CAiSE*02)* (pp. 1–4). Toronto (Ontario, Canada): CEUR-WS.
- Glick, T. H., & Moore, G. T. (2001). Time to learn: the outlook for renewal of patient-centred education in the digital age. *Medical Education*, 35(5), 505–509. doi:10.1046/j.1365-2923.2001.00935.x
- Goldblatt, R. (1992). Prepositional Dynamic Logic. In R. Goldblatt (Ed.), *Logics of Time and Computation* (2nd ed., pp. 109–139). CSLI.
- Goldstone, W. (2010). *Développez des jeux 3D avec Unity*. Pearson.
- Gomathi, P. M., & Nasira, G. M. (2015). Detection Of Dyslexia From Eye Movements Using Anfis & Bbwp Feature Extraction Methods. *Research Inventy: International Journal Of Engineering And Science*, 05(01), 20–29.
- Gombert, A., Feuilladiou, S., Gilles, P.-Y., & Roussey, J.-Y. (2008). La scolarisation d'élèves dyslexiques sévères en classe ordinaire de collège: lien entre adaptations pédagogiques, point de vues des enseignants et vécu de l'expérience scolaire des élèves. *Revue Française de Pédagogie*, 164, 123–138.
- Gonçalves, F. M. (2013). Bossa. Retrieved from <http://sourceforge.net/projects/bossa/>
- Gonzalez, A. (2014). OpenDyslexic: Free and Open Source Dyslexia Typeface. Retrieved from <http://opendyslexic.org/>
- Gonzalez-Sanchez, J., Chavez-Echeagaray, M. E., Atkinson, R., & Burleson, W. (2011). ABE: An Agent-Based Software Architecture for a Multimodal Emotion Recognition Framework. In *2011 Ninth Working IEEE/IFIP Conference on Software Architecture* (pp. 187–193). IEEE. doi:10.1109/WICSA.2011.32
- González-Vélez, H., Mier, M., Julià-Sapé, M., Arvanitis, T. N., García-Gómez, J. M., Robles, M., ... Lluch-Ariet, M. (2009). HealthAgents: Distributed Multi-Agent Brain Tumor Diagnosis and Prognosis. *Applied Intelligence*, 30(3), 191–202. doi:10.1007/s10489-007-0085-8
- Gooch, P., & Roudsari, A. (2011). Computerization of workflows, guidelines, and care pathways: a review of implementation challenges for process-oriented health information systems. *Journal of the American Medical Informatics Association*, 18(6), 738–748. doi:10.1136/amiajnl-2010-000033
- Google Inc. (2015a). Google. Retrieved from <https://www.google.fr>
- Google Inc. (2015b). Google Drive. Retrieved from <https://drive.google.com>
- Google Inc. (2015c). Google Scholar. Retrieved from <http://scholar.google.fr/>

- Gordon, C., Herbert, I., & Johnson, P. D. (1996). Knowledge Representation and Clinical Practice Guidelines: the DILEMMA and PRESTIGE projects. In J. Brender, J. P. Christensen, J.-R. Scherrer, & P. McNair (Eds.), *Medical Informatics Europe '96* (pp. 511–515). IOS Press. doi:10.3233/978-1-60750-878-6-511
- Goswami, U. (2003). Why theories about developmental dyslexia require developmental designs. *Trends in Cognitive Sciences*, 7(12), 534–540. doi:10.1016/j.tics.2003.10.003
- Goswami, U., Wang, H., & Cruz, A. (2010). Language-universal sensory deficits in developmental dyslexia: English, Spanish, and Chinese. *Journal of Cognitive Neuroscience*, 23(2), 325–337. doi:10.1162/jocn.2010.21453
- Green, S., Hurst, L., Nangle, B., Cunningham, P., Somers, F., & Evans, R. (1997). *Software agents: A review*. Dublin.
- Greenes, R. A. (2014). Features of computer-based clinical decision support. In R. A. Greenes (Ed.), *Clinical decision support: the road ahead* (2nd ed., pp. 111–144). Elsevier. doi:10.1016/B978-0-12-398476-0.00003-8
- Greenes, R. A., Boxwala, A., Sloan, W. N., Ohno-Machado, L., & Deibel, S. R. A. (1999). A framework and tools for authoring, editing, documenting, sharing, searching, navigating, and executing computer-based clinical guidelines. In *AMIA Annual Symposium Proceedings* (pp. 261–265). American Medical Informatics Association.
- Greggio, N., Bernardino, A., Laschi, C., Dario, P., & Santos-Victor, J. (2010). An algorithm for the least square-fitting of ellipses. In *22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 351–353). Arras: IEEE. doi:10.1109/ICTAI.2010.124
- Griss, M. L., Chen, Q., Bolcer, G. A., Kessler, R. R., & Osterweil, L. J. (1999). Agents and Workflow -- An Intimate Connection, or Just Friends? In *31st International Conference on Technology of Object-Oriented Language and Systems (TOOLS 31)* (p. 558). Nanjing, China: IEEE. doi:10.1109/TOOLS.1999.10038
- Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199–220. doi:10.1006/knac.1993.1008
- Grudin, J. (2012). A moving target: The evolution of human-computer interaction. In J. A. Jacko (Ed.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications* (3rd ed., pp. 1–25). CRC Press. doi:10.1201/b11963-1
- Guedes, G. T. A., & Vicari, R. M. (2009). Applying AUML and UML 2 in the Multi-agent Systems Project. In C. A. Heuser & G. Pernul (Eds.), *Advances in Conceptual Modeling - Challenging Perspectives* (pp. 106–115). Springer Berlin Heidelberg. doi:10.1007/978-3-642-04947-7_14
- Guérin, B.-A. (2012). *Conduite de projets informatiques : Développement, analyse et pilotage (2nd édition)*. Editions ENI.
- Guidelines International Network. (2014). Welcome to G-I-N. Retrieved from <http://www.g-i-n.net/>
- Guo, L. (2007). *Enacting a Decentralised Workflow Management System on a Multi-agent Platform*. University of Edinburgh.

- Gupta, P., & Govil, M. C. (2010). MVC Design Pattern for the multi framework distributed applications using XML, spring and struts framework. *International Journal on Computer Science and Engineering*, 02(04), 1047–1051.
- Gurban, M., & Thiran, J.-P. (2010). Modality Integration Methods. In J.-P. Thiran, F. Marqués, & H. Bourlard (Eds.), *Multimodal Signal Processing: Theory and Applications for Human–Computer Interaction* (1st ed., pp. 171–184). Elsevier Ltd. doi:10.1016/b978-0-12-374825-6.00007-1
- Gutknecht, O., Ferber, J., & Michel, F. (2000). *MadKit : une architecture de plate-forme multi-agent générique*. Montpellier, France.
- Gutknecht, O., Ferber, J., & Michel, F. (2001). Integrating tools and infrastructures for generic multi-agent systems. In E. André, S. Sen, C. Frasson, & J. P. Müller (Eds.), *Fifth International Conference on Autonomous Agents (AGENTS '01)* (pp. 441–448). Montréal, Quebec, Canada: ACM Press. doi:10.1145/375735.376410
- Habib, M. (2000). The neurological basis of developmental dyslexia: an overview and working hypothesis. *Brain: A Journal of Neurology*, 123 Pt 12, 2373–99. doi:10.1093/brain/123.12.2373
- Habib, M., Joannette, Y., & Lecours, A. R. (2000). Le cerveau humain et les origines du langage. *Médecine/Sciences*, 16(2), 171–180. doi:10.4267/10608/1618
- Habib, M., & Joly-Pottuz, B. (2008). Dyslexie, du diagnostic à la thérapeutique: un état des lieux. *Revue de Neuropsychologie*, 18(4), 247–325.
- Habib, M., & Robichon, F. (1996). Les mécanismes cérébraux de la lecture : un modèle en neurologie cognitive. *Médecine/Sciences*, 12(6-7), 707–714. doi:10.4267/10608/811
- Hadorn, D. C. (1995). Use of algorithms in clinical guideline development. In Agency for Health Care Policy and Research (Ed.), *Clinical Practice Guideline Development: Methodology Perspectives* (pp. 93–104). AHCPH Pub.
- Hall, R., Jones, N., & Purves, I. N. (2002). Acquiring, encoding, and re-using clinical knowledge in PRODIGY. *Informatics in Primary Care*, 10(4), 211–216.
- Hammermüller, K., & Miksch, S. (1999). A Workflow Model for the Asgaard Project: A Time-oriented, Skeletal Planning Workbench in Medicine. In *1st Workshop on Computers in Anaesthesia and Intensive Care: Knowledge-Based Information Management in conjunction with the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM'99)*. Aalborg, Denmark.
- Han, B.-M., Song, S.-J., Lee, K. M., Jang, K.-S., & Shin, D.-R. (2006). Multi-agent system based efficient healthcare service. In *Advanced Communication Technology (ICACT 2006)* (pp. 47–51). Phoenix Park: IEEE. doi:10.1109/ICACT.2006.205916
- Han, X., Xu, C., & Prince, J. L. (2001). A topology preserving deformable model using level sets. In *Computer Vision and Pattern Recognition. CVPR 2001* (pp. 765–770).
- Han, X., Xu, C., & Prince, J. L. (2003). A topology preserving level set method for geometric deformable models. *Pattern Analysis and Machine Intelligence*, 25(6), 755–768.
- Hanák, P., Kiss, N., Kováčsházy, T., Pataki, B., Salamon, M., Seres, C., ... Varga, J. (2011). System Architecture for Home Health and Patient Activity Monitoring. In Á. Jobbágy (Ed.), *5th European Conference of the International Federation for Medical and*

- Biological Engineering* (pp. 945–948). Budapest, Hungary: Springer Berlin Heidelberg. doi:10.1007/978-3-642-23508-5_245
- Hansen, D. W., & Ji, Q. (2010). In the eye of the beholder: a survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3), 478–500. doi:10.1109/TPAMI.2009.30
- Hansen, D. W., & Pece, A. E. C. (2005). Eye tracking in the wild. *Computer Vision and Image Understanding*, 98(1), 155–181. doi:10.1016/j.cviu.2004.07.013
- Hanseth, O., & Lundberg, N. (2001). Designing work oriented infrastructures. *Computer Supported Cooperative Work (CSCW)*, 10(3-4), 347–372. doi:10.1023/A:1012727708439
- Hanseth, O., & Lyytinen, K. (2004). Theorizing about the design of Information Infrastructures: design kernel theories and principles. *Sprouts: Working Papers on Information Systems*, 4(12), 207–241.
- Hashmi, Z. I., Abidi, S. S. R., & Cheah, Y.-N. (2002). An intelligent agent-based knowledge broker for enterprise-wide healthcare knowledge procurement. In *15 th IEEE Symposium on Computer-Based Medical Systems (CBMS'02)* (pp. 0–5). IEEE. doi:10.1109/CBMS.2002.1011373
- Hatcliff, J., King, A., Lee, I., MacDonald, A., Fernando, A., Robkin, M., ... Goldman, J. M. (2012). Rationale and architecture principles for medical application platforms. In *Cyber-Physical Systems (ICCPS), 2012* (pp. 3–12). IEEE/ACM. doi:10.1109/ICCPS.2012.9
- Haute Autorité de Santé. (2013a). *Grille de pilotage et de sécurité d'un projet de télémédecine*.
- Haute Autorité de Santé. (2013b). Protocoles nationaux de diagnostic et de soins (PNDS). Retrieved from http://www.has-sante.fr/portail/jcms/c_1340879/fr/protocoles-nationaux-de-diagnostic-et-de-soins-pnds
- Haute Autorité de Santé. (2014). Tableau des recommandations de bonne pratique. Retrieved from http://www.has-sante.fr/portail/jcms/c_1101438/fr/tableau-des-recommandations-de-bonne-pratique
- Haute Autorité de Santé. (2015). Haute Autorité de Santé - Page d'accueil. Retrieved from <http://www.has-sante.fr>
- Hawryszkiewicz, I., & Debenham, J. (1998). A workflow system based on agents. In G. Quirchmayr, E. Schweighofer, & T. J. M. Bench-Capon (Eds.), *9th International Conference on Database and Expert Systems Applications (DEXA'98)* (pp. 135–144). Vienna, Austria: Springer Berlin Heidelberg. doi:10.1007/BFb0054475
- He, L., Peng, Z., Everding, B., Wang, X., Han, C. Y., Weiss, K. L., & Wee, W. G. (2008). A comparative study of deformable contour methods on medical image segmentation. *Image and Vision Computing*, 26(2), 141–163. doi:10.1016/j.imavis.2007.07.010
- Health Level Seven International. (2014). Health Level Seven International: Tools & Resources. Retrieved from <http://www.hl7.org/participate/toolsandresources.cfm>
- Heath, B., Romero, P. W., & Reynolds, K. (2013). *A standard framework for levels of integrated healthcare*. Washington, DC.

- Heath, C., & Luff, P. (1996). Documents and professional practice: “bad” organisational reasons for “good” clinical records. In M. S. Ackerman (Ed.), *The 1996 ACM conference on Computer supported cooperative work (CSCW '96)* (pp. 354–363). Boston, MA, USA: ACM Press. doi:10.1145/240080.240342
- Heckel, R., & Engels, G. (2002). Towards a Formal Framework for Inter-Enterprise Application Integration. *Electronic Notes in Theoretical Computer Science*, 51, 139–151. doi:10.1016/S1571-0661(04)80198-1
- Hecker, L., & Engstrom, E. U. (2005). Assistive Technology and Individuals with Dyslexia. In *Multisensory Teaching of Basic Language Skills (2nd Edition)* (Judith R. .). Paul H. Brookes Publishing Company.
- Hegadi, R., Kop, A., & Hangarge, M. (2010). A Survey on Deformable Model and its Applications to Medical Imaging. *IJCA, Special Issue on RTIPPR, 2008(2)*, 64–75.
- Herbert, S. I., Gordon, C. J., Jackson-Smale, A., & Salis, J.-L. R. (1995). Protocols for clinical care. *Computer Methods and Programs in Biomedicine*, 48(1-2), 21–26. doi:10.1016/0169-2607(95)01655-D
- Hernandez, S. R. (2000). Horizontal and vertical healthcare integration: lessons learned from the United States. *HealthcarePapers*, 1(2), 59–66. doi:10.12927/hcpap.2000.17219
- Hersh, W. (2004). Health Care Information Technology: Progress and Barriers. *Journal of the American Medical Association*, 292(18), 2273–2274. doi:10.1001/jama.292.18.2273
- Hewitt, C. (1977). Viewing control structures as patterns of passing messages. *Artificial Intelligence*, 8(3), 323–364. doi:10.1016/0004-3702(77)90033-9
- Hijazi, H., Khmour, T., & Alarabeyyat, A. (2012). A Review of Risk Management in Different Software Development Methodologies. *International Journal of Computer Applications*, 45(7), 8–12. doi:10.5120/6790-9113
- Hilitch, C. J. (1969). Linear Skeletons From Square Cupboards. *Machine Intelligence 4*, 403–420.
- Hollingsworth, D. (1995). *Workflow management coalition: the workflow reference model (TC00-1003 v1.1)*. Workflow Management Coalition. Hampshire, UK.
- Hollingsworth, D. (2004). The workflow reference model: 10 years on. In L. Fischer (Ed.), *The Workflow Handbook* (pp. 295–312). Future Strategies Inc.
- Hollingsworth, J. E., & Weide, B. W. (1994). *Micro-Architecture vs. Macro-Architecture (OSU-CISRC-11/94-TR57)*.
- Hripcsak, G. (1994). Writing Arden Syntax medical logic modules. *Computers in Biology and Medicine*, 24(5), 331–363. doi:10.1016/0010-4825(94)90002-7
- Hripcsak, G., Ludemann, P., Pryor, T. A., Wigertz, O. B., & Clayton, P. D. (1994). Rationale for the Arden syntax. *Computers and Biomedical Research*, 27(4), 291–324. doi:10.1006/cbmr.1994.1023
- Hsieh, F.-S., & Lin, J.-B. (2014). A multiagent approach for managing collaborative workflows in supply chains. In *IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (pp. 71–76). Hsinchu, Taiwan: IEEE. doi:10.1109/CSCWD.2014.6846819

- Hübner, J. F., Sichman, J. S., & Boissier, O. (2006). S-Moise+: A Middleware for Developing Organised Multi-agent Systems. In *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems* (pp. 64–77). Springer Berlin Heidelberg. doi:10.1007/11775331_5
- Huget, M.-P. (2001). *Une ingénierie des protocoles d'interaction pour les systèmes multi-agents*. Université Paris IX - Dauphine.
- Huget, M.-P. (2004a). Agent UML Notation for Multiagent System Design. *Internet Computing*, 8(4), 63–71. doi:10.1109/MIC.2004.6
- Huget, M.-P. (2004b). Representing goals in multiagent systems. In J. Mueller & P. Petta (Eds.), *4th International Symposium From Agent Theory to Agent Implementation (AT2AI-4)* (pp. 588–593). Vienna, Austria: Austrian Society for Cybernetic Studies.
- Huget, M.-P., & Koning, J.-L. (2003). Interaction Protocol Engineering. In M.-P. Huget (Ed.), *Communication in Multiagent Systems: Agent Communication Languages and Conversation Policies* (pp. 179–193). Springer Berlin Heidelberg. doi:10.1007/978-3-540-44972-0_9
- Huget, M.-P., & Odell, J. (2005). Representing agent interaction protocols with agent UML. In J. Odell, P. Giorgini, & J. P. Müller (Eds.), *Agent-Oriented Software Engineering V* (pp. 16–30). Springer Berlin Heidelberg. doi:10.1007/978-3-540-30578-1_2
- Huhns, M. N., & Singh, M. P. (1998). Managing heterogeneous transaction workflows with co-operating agents. In N. R. Jennings & M. J. Wooldridge (Eds.), *Agent Technology* (pp. 219–239). Springer Berlin Heidelberg. doi:10.1007/978-3-662-03678-5_12
- Hunt, D. L., Haynes, R. B., Hanna, S. E., & Smith, K. (1998). Effects of Computer-Based Clinical Decision Support Systems on Physician Performance and Patient Outcomes: A Systematic Review. *Journal of the American Medical Association*, 280(15), 1339–1346. doi:10.1001/jama.280.15.1339
- Hunt, J. (2003). An Introduction to the UML and the Unified Process. In J. Hunt (Ed.), *Guide to the Unified Process featuring UML, Java and Design Patterns* (Springer P., pp. 21–37). Springer London. doi:10.1007/1-85233-856-3_3
- Huser, V., Rasmussen, L. V., Oberg, R., & Starren, J. B. (2011). Implementation of workflow engine technology to deliver basic clinical decision support functionality. *BMC Medical Research Methodology*, 11(43), 1–19. doi:10.1186/1471-2288-11-43
- Huser, V., Rasmussen, L. V., & Starren, J. B. (2009). Representing clinical processes in XML process definition language (XPDL). In *AMIA Spring Symposium*. Orlando, Florida.
- Huser, V., Rocha, R. A., & James, B. C. (2006). Use of workflow technology tools to analyze medical data. In *19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)* (pp. 455–460). Salt Lake City, Utah: IEEE. doi:10.1109/CBMS.2006.162
- Hutzler, G., Kludel, H., & Wang, D. H. (2006). Systèmes multi-agents adaptatifs avec contraintes temps-réel : de la spécification formelle à la vérification et à la génération de code. *Technique et Science Informatiques*, 25(3), 317–342. doi:10.3166/tsi.25.317-342

- Hyrskykari, A. (2006). Utilizing eye movements: Overcoming inaccuracy while tracking the focus of attention during reading. *Computers in Human Behavior*, 22(4), 657–671. doi:10.1016/j.chb.2005.12.013
- IBM. (1970). *Flowcharting techniques (CG20-8152-1)*. New York, NY, USA.
- IBM. (2014). Rational Rose family. Retrieved from <http://www.ibm.com/software/awdtools/developer/rosexde/>
- IEEE Computer Society. (1998). *1320.1-1998 - IEEE Standard for Functional Modeling Language - Syntax and Semantics for IDEF0* (2004th ed., Vol. 1998). IEEE. doi:10.1109/IEEESTD.1998.89198
- IEEE Computer Society. (2014). The Foundation for Intelligent Physical Agents (FIPA). Retrieved from <http://www.fipa.org/>
- IEEE Computer Society. (2015). IEEE Xplore Digital Library. Retrieved from <http://ieeexplore.ieee.org>
- Iglesias, C. A., Garijo, M., González, J. C., & Velasco, J. R. (1998). Analysis and design of multiagent systems using MAS-CommonKADS. In M. P. Singh, A. Rao, & M. J. Wooldridge (Eds.), *Intelligent Agents IV Agent Theories, Architectures, and Languages* (pp. 313–327). Providence, Rhode Island, USA: Springer Berlin Heidelberg. doi:10.1007/BFb0026768
- IHE International. (2013). Integrating the Healthcare Environment. Retrieved from <http://www.ihe.net>
- Ilie, V., Courtney, J. F., & Van Slyke, C. (2007). Paper versus Electronic: Challenges Associated with Physicians' Usage of Electronic Medical Records. In *40th Annual Hawaii International Conference on System Sciences (HICSS'07)* (pp. 142–152). Waikoloa, HI: IEEE. doi:10.1109/HICSS.2007.424
- InferMed Limited. (2014). InferMed | Arezzo Clinical Decision Support | MACRO Electronic Data Capture. Retrieved from <http://www.infermed.com/>
- Ingvar, M., af Trampe, P., Greitz, T., Eriksson, L., Stone-Elander, S., & von Euler, C. (2002). Residual differences in language processing in compensated dyslexics revealed in simple word reading tasks. *Brain and Language*, 83(2), 249–267. doi:10.1016/S0093-934X(02)00055-X
- Inria. (2015). OpenViBE. Inria. Retrieved from <http://openvibe.inria.fr/>
- INSERM (dir.). (2007). *Dyslexie, dysorthographe, dyscalculie: bilan des données scientifiques*. Paris: INSERM.
- Intel. (2013). Intel® Perceptual Computing SDK. Retrieved from <http://software.intel.com/en-us/vcsource/tools/perceptual-computing-sdk>
- Intel. (2015). Intel® Developer Zone. Retrieved from <https://software.intel.com>
- Intel, & Willow Garage. (2013). OpenCV. Retrieved from <http://opencv.org/>
- International Health Terminology Standards Development Organisation. (2014). SNOMED CT: The Global Language of Healthcare. Retrieved from <http://www.ihtsdo.org/snomed-ct>

- International Health Terminology Standards Development Organisation. (2015). Welcome to IHTSDO. Retrieved from <http://www.ihtsdo.org/>
- International Organization for Standardization. (2011). Structured Query Language. Retrieved from <https://www.iso.org/obp/ui/#iso:std:iso-iec:9075:-1:ed-4:v1:en>
- International Organization for Standardization. (2013). ISO/TR 24971:2013. Retrieved from http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=59587
- International Organization for Standardization. (2015a). IEC 60601-1-11:2015. Retrieved from http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=65529
- International Organization for Standardization. (2015b). ISO/IEEE 11073 search on www.ISO.org. Retrieved from <http://www.iso.org/iso/home/search.htm?qt=%22ISO%2FIEEE+11073%22&sort=rel&type=simple&published=on>
- IOS Press. (2015). IOS Press | Impacting the World of Science. Retrieved from <http://www.iospress.nl/>
- Irwin, D. E. (2004). Fixation Location and Fixation Duration as Indices of Cognitive Processing. In J. Henderson & F. Ferreira (Eds.), *The interface of language, vision, and action: Eye movements and the visual world* (pp. 105–133). Psychology Press.
- Isern, D. (2008). *Agent-based management of clinical guidelines*. Universitat Politècnica de Catalunya.
- Isern, D., & Moreno, A. (2008). Computer-based execution of clinical guidelines: A review. *International Journal of Medical Informatics*, 77(12), 787–808. doi:10.1016/j.ijmedinf.2008.05.010
- Isern, D., Sánchez, D., & Moreno, A. (2010). Agents applied in health care: A review. *International Journal of Medical Informatics*, 79(3), 145–166. doi:10.1016/j.ijmedinf.2010.01.003
- Isern González, J., Cabrera Gámez, J., Hernández Sosa, J. D., & Domínguez Brito, A. C. (2007). Comparing self-calibration methods for static cameras. In *EUROCAST'07* (pp. 660–667).
- Isern González, J., Hernández Sosa, J. D., Domínguez Brito, A. C., & Naranjo Cabrera, A. M. (2005). Comparative analysis of calibration methods for a static camera. In *EUROCAST'05*.
- Iwashita, Y., Kurazume, R., Tsuji, T., Hasegawa, T., & Hara, K. (2004). Fast implementation of level set method and its real-time applications. In *Systems, Man and Cybernetics* (Vol. 7, pp. 6302–6307). IEEE. doi:10.1109/ICSMC.2004.1401389
- Jacob, R. J. K. (1991). The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems*, 9(2), 152–169. doi:10.1145/123078.128728
- Jacob, R. J. K., & Karn, K. S. (2003). Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In J. Hyönä, R. Radach, & H. Deubel (Eds.), *The*

- Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research* (pp. 573–605). Elsevier B.V. doi:10.1016/B978-044451020-4/50031-1
- Jacobs, A. K., Quinn, T. A., & Nelson, S. J. (2006). Mapping SNOMED-CT concepts to MeSH concepts. In *AMIA Annual Symposium Proceedings* (p. 965). American Medical Informatics Association.
- Jacobson, I. (2008a). *The Steps to an Architecture*.
- Jacobson, I. (2008b). *The Unified Process Works through Iterations*.
- Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The Unified Software Development Process*. Addison-Wesley.
- Jacquier-Roux, M., Lequette, C., Pouget, G., Valdois, S., & Zorman, M. (2010). *B.A.L.E. (Batterie Analytique du Langage Ecrit)*. (Cogni-Sciences, Ed.). Grenoble.
- Jain, K., Manghirmalani, P., Dongardive, J., & Abraham, S. (2009). Computational diagnosis of learning disability. *International Journal of Recent Trends in Engineering*, 2(3), 64–66.
- Jalender, B., Govardhan, A., & Premchand, P. (2012). Designing code level reusable software components. *International Journal of Software Engineering & Applications (IJSEA)*, 3(1), 219–229. doi:10.5121/ijsea.2012.3116
- Jennings, N. R. (1996). Using intelligent agents to manage business processes. In *IEE Colloquium on Intelligent Agents and Their Applications* (pp. 5/1–5/3). IET. doi:10.1049/ic:19960651
- Jennings, N. R. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4), 35–41. doi:10.1145/367211.367250
- Ji, Q., & Zhu, Z. (2002). Eye and gaze tracking for interactive graphic display. In *2nd International Symposium on Smart Graphics (SMARTGRAPH'02)* (pp. 79–85). Hawthorne, NY, USA: ACM Press. doi:10.1007/s00138-004-0139-4
- Jimenez, J., Gutierrez, D., Latorre, P., & Zaragoza, U. De. (2008). Gaze-based Interaction for Virtual Environments. *Journal of Universal Computer Science*, 14(19), 3085–3098.
- Johannesson, P., & Perjons, E. (2000). Design principles for application integration. In B. Wangler & L. Bergman (Eds.), *Advanced Information Systems Engineering* (Vol. 1789, pp. 212–231). Springer Berlin Heidelberg. doi:10.1007/3-540-45140-4_15
- Johnson, P. D., & Musen, M. A. (1997). *Development of a guideline authoring tool with PROTEGE-II based on the DILEMMA generic protocol and guideline model*. Pala Alto, CA, USA.
- Johnson, P. D., Tu, S., & Jones, N. (2001). Achieving reuse of computable guideline systems. In V. L. Patel, R. Rogers, & R. Haux (Eds.), *MEDINFO 2001: Proceedings of the 10th World Congress on Medical Informatics* (Vol. 84, pp. 99–103). IOS Press. doi:10.3233/978-1-60750-928-8-99
- Johnson, P. D., Tu, S. W., Booth, N., Sugden, B., & Purves, I. N. (2000). Using scenarios in chronic disease management guidelines for primary care. In *AMIA Annual Symposium Proceedings* (pp. 389–393). American Medical Informatics Association.

- Johnson, P. D., Tu, S. W., Musen, M. A., & Purves, I. (2001). A virtual medical record for guideline-based decision support. In *AMIA Annual Symposium Proceedings* (pp. 294–298).
- Johnson, R. E. (1997). Components, frameworks, patterns. *ACM SIGSOFT Software Engineering Notes*, 22(3), 10–18. doi:10.1145/258368.258378
- Jørgensen, A. H. (2008). Taking stock of user interface history. In *5th Nordic conference on Human-computer interaction building bridges - NordiCHI '08* (pp. 479–482). ACM Press. doi:10.1145/1463160.1463224
- Jørgensen, D., Hallenborg, K., & Demazeau, Y. (2014). Assessment of agent architectures for Telehealth. In J. M. Corchado, J. Bajo, J. Kozlak, P. Pawlewski, J. M. Molina, B. Gaudou, ... P. G. Teodoro (Eds.), *PAAMS 2014 International Workshops* (pp. 79–88). Salamanca, Spain: Springer International Publishing. doi:10.1007/978-3-319-07767-3_8
- Jovanov, E., Starčević, D., Samardžić, A., Marsh, A., & Obrenović, Z. (1999). EEG analysis in a telemedical virtual world. *Future Generation Computer Systems*, 15(2), 255–263. doi:10.1016/S0167-739X(98)00068-5
- Jucla, M. (2009). *Aspects phonologiques, orthographiques et lexico-sémantiques du traitement du langage: production orale versus écrite chez l'adulte et protocole d'intervention dans les dyslexies du développement - études comportementales et par la méthode des potentiels*. université de Toulouse.
- Jucla, M., Nenert, R., Chaix, Y., & Demonet, J.-F. (2010). Remediation effects on N170 and P300 in children with developmental dyslexia. *Behavioural Neurology*, 22(3-4), 121–129. doi:10.3233/ben-2009-0257
- Jurcak, V., Tsuzuki, D., & Dan, I. (2007). 10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems. *NeuroImage*, 34(4), 1600–1611. doi:10.1016/j.neuroimage.2006.09.024
- Juristo, N., Moreno, A. M., & Sanchez-Segura, M.-I. (2006). *Usability Elicitation Patterns (USEPs)*.
- Juristo, N., Moreno, A. M., & Sanchez-Segura, M.-I. (2007). Guidelines for eliciting usability functionalities. *IEEE Transactions on Software Engineering*, 33(11), 744–758. doi:10.1109/TSE.2007.70741
- Juristo, N., Windl, H., & Constantine, L. (2001). Introducing Usability. *IEEE Software*, 18(1), 20–21. doi:10.1109/MS.2001.903155
- Jürjens, J. (2002). UMLsec: Extending UML for secure systems development. In J.-M. Jézéquel, H. Hussmann, & S. Cook (Eds.), *« UML » 2002 — The Unified Modeling Language* (pp. 412–425). Dresden, Germany: Springer Berlin Heidelberg. doi:10.1007/3-540-45800-X_32
- Juziuk, J., Weyns, D., & Holvoet, T. (2014). Design Patterns for Multi-agent Systems: A Systematic Literature Review. In *Agent-Oriented Software Engineering: Reflections on Architectures, Methodologies, Languages, and Frameworks* (pp. 79–99). Springer Berlin Heidelberg. doi:10.1007/978-3-642-54432-3_5
- KAISHA-Tec Co. Ltd. Japan. (2014). ActiveModeler Advantage BPM Foundation. Retrieved from <http://www.activemodeler.com/>

- Kalra, D. (2006). Electronic health record standards. In *IMIA Yearbook 2006: Assessing Information - Technologies for Health* (pp. 136–144). Stuttgart: Schattauer Publishers.
- Kamel, N. (2006). *Un cadre formel générique pour la modélisation d'IHM multi-modales : cas de la multi-modalité en entrée*. Université de Poitiers.
- Kanatani, K., & Sugaya, Y. (2007). Performance evaluation of iterative geometric fitting algorithms. *Computational Statistics & Data Analysis*, 52(2), 1208–1222. doi:10.1016/j.csda.2007.05.013
- Kaplan, B. J., Wilson, B. N., Dewey, D., & Crawford, S. G. (1998). DCD may not be a discrete disorder. *Human Movement Science*, 17, 471–490.
- Kapur, J. N., Sahoo, P. K., & Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29(1), 273–285. doi:10.1016/S0734-189X(85)90156-2
- Karapetsas, A. V., & Zygouris, N. C. (2011). Event Related Potentials (ERPs) in prognosis, diagnosis and rehabilitation of children with dyslexia. *Encephalos: Archives of Neurology and Psychiatry*, 48(3), 118–127.
- Karjoth, G., Asokan, N., & Gülcü, C. (1998). Protecting the computation results of free-roaming agents. *Personal Technologies*, 2(2), 92–99. doi:10.1007/BF01324939
- Karmiloff-Smith, A. (1998). Development itself is the key to understanding developmental disorders. *Trends in Cognitive Sciences*, 2(10), 389–98.
- Kaschek, R., Matthews, C., Schewe, K.-D., & Wallace, C. (2006). Information systems design: through adaptivity to ubiquity. *Information Systems and E-Business Management*, 4(2), 137–158. doi:10.1007/s10257-005-0028-z
- Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 321–331.
- Kassner, M. P., & Patera, W. R. (2012). *PUPIL: constructing the space of visual attention*. MASSACHUSETTS INSTITUTE OF TECHNOLOGY.
- Kassner, M. P., Patera, W. R., & Bulling, A. (2014). Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction. In A. J. Brush, A. Friday, J. Kientz, J. Scott, & J. Song (Eds.), *The 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp'14 Adjunct)* (pp. 1151–1160). Seattle, Washington, USA: ACM Press. doi:10.1145/2638728.2641695
- Kavi, K., Kung, D. C., Bhambhani, H., Pancholi, G., & Kanikarla, M. (2003). Extending UML for Modeling and Design of Multi-Agent Systems. In *Second International Workshop on Software Engineering for Large Muti-Agent Systems (SELMAS'03)* (pp. 1–8). Portland, Oregon.
- Kawamoto, K., & Greenes, R. A. (2014). The Role of Standards: What We Can Expect and When. In R. A. Greenes (Ed.), *Clinical decision support: the road ahead* (2nd ed., pp. 599–615). Elsevier. doi:10.1016/B978-0-12-398476-0.00021-X
- Khamis, A., & Abdelmonem, A. (2002). The unified software development process and framework. *Dogus University Journal*, 3(1), 109–122.

- Kilman, D. G., & Forslund, D. W. (1997). An international collaboratory based on virtual patient records. *Communications of the ACM*, 40(8), 110–117. doi:10.1145/257874.257898
- Kim, S., Lombardino, L. J., Cowles, W., & Altmann, L. J. (2014). Investigating graph comprehension in students with dyslexia: an eye tracking study. *Research in Developmental Disabilities*, 35(7), 1609–1622. doi:10.1016/j.ridd.2014.03.043
- Kirn, S., & Gasser, L. (1998). *Organizational Approaches to Coordination in Multi-Agent Systems*.
- Kirn, S., Heine, C., Herrler, R., & Krempels, K.-H. (2003). Agent.Hospital - a Framework for Clinical Applications in Agentcities. In A. Moreno & J. L. Nealon (Eds.), *Applications of Software Agent Technology in the Health Care Domain* (pp. 67–85). Birkhäuser Basel. doi:10.1007/978-3-0348-7976-7_6
- Kishore, R., Zhang, H., & Ramesh, R. (2006). Enterprise integration using the agent paradigm: foundations of multi-agent-based integrative business information systems. *Decision Support Systems*, 42(1), 48–78. doi:10.1016/j.dss.2004.09.011
- Kittler, J., & Illingworth, J. (1985). On threshold selection using clustering criteria. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(5), 652–655. doi:10.1109/TSMC.1985.6313443
- Kittler, J., & Illingworth, J. (1986). Minimum error thresholding. *Pattern Recognition*, 19(1), 41–47. doi:10.1016/0031-3203(86)90030-0
- Klem, G. H., Lüders, H. O., Jasper, H. H., & Elger, C. (1999). The ten-twenty electrode system of the International Federation. *Electroencephalography And Clinical Neurophysiology (Supplement)*, 52, 3–6.
- Klette, G. (2003). Simple points in 2D and 3D binary images. *Computer Analysis of Images and Patterns*, 2756, 57–64. doi:10.1007/978-3-540-45179-2_8
- Klinger, E. (2006). *Apports de la réalité virtuelle à la prise en charge de troubles cognitifs et comportementaux*. Ecole Nationale Supérieure des Télécommunications.
- Klinker, G., Bhola, C., Dallemagne, G., Marques, D., & McDermott, J. (1991). Usable and reusable programming constructs. *Knowledge Acquisition*, 3(2), 117–135. doi:10.1016/1042-8143(91)90001-4
- Knowledge Based Systems Inc. (2014). AIØ WIN®: Activity Modeling & ABC. Retrieved from <http://www.kbsi.com/products/aio-win>
- Ko, R. K. L. (2009). A computer scientist's introductory guide to business process management (BPM). *Crossroads*, 15(4), 11–18. doi:10.1145/1558897.1558901
- Kohlbecher, S., Bardins, S., Bartl, K., Schneider, E., Poitschke, T., & Ablassmeier, M. (2008). Calibration-free eye tracking by reconstruction of the pupil ellipse in 3D space. In *The 2008 symposium on Eye tracking research & applications* (Vol. 1, pp. 135–138). IEEE. doi:10.1145/1344471.1344506
- Kolb, M. (1995). A Cooperation Language. In V. Lesser (Ed.), *First International Conference on Multi-Agent Systems (ICMAS'95)* (pp. 233–238). San Francisco, California, USA: AAAI/MIT Press.

- Komogortsev, O. V., & Karpov, A. (2013). Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades. *Behavior Research Methods*, 45(1), 203–215. doi:10.3758/s13428-012-0234-9
- Kong, T. Y. (1997). Topology-preserving deletion of 1's from 2-, 3-and 4-dimensional binary images. In *Discrete Geometry for Computer Imagery (DGCI'97)* (pp. 1–18). Montpellier (France): Springer Berlin Heidelberg. doi:10.1007/BFb0024826
- Kong, T. Y., & Rosenfeld, A. (1989). Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48(3), 357–393. doi:10.1016/0734-189X(89)90127-8
- Koning, J.-L., & Huget, M.-P. (2001). A semi-formal specification language dedicated to interaction protocols. In H. Jaakkola, H. Kangassalo, & E. Kawaguchi (Eds.), *Information Modelling and Knowledge Bases XII* (pp. 375–392). IOS Press.
- Koning, J.-L., Huget, M.-P., Wei, J., & Wang, X. (2002). Extended Modeling Languages for Interaction Protocol Design. In M. J. Wooldridge, G. Weiß, & P. Ciancarini (Eds.), *Agent-Oriented Software Engineering II* (pp. 68–83). Montreal, Canada: Springer Berlin Heidelberg. doi:10.1007/3-540-70657-7
- Koning, J.-L., & Pesty, S. (2001). Modèles de communication. In J.-P. Briot & Y. Demazeau (Eds.), *Principes et architecture des systèmes multi-agents* (Traité IC2., pp. 109–138). Lavoisier.
- Kook, Y.-G., Lee, J., Park, M.-W., Kim, J.-S., & Choi, K.-S. (2011). An approach to access the distributed data based on the multi-agent system for interoperability. In *Future Generation Information Technology* (pp. 215–222). Springer Berlin Heidelberg. doi:10.1007/978-3-642-27142-7_25
- Koons, D. B., & Thorisson, K. R. (1993). Estimating direction of gaze in multi-modal context. In *The Third International Conference on Cyberspace (3CYBERCONF)* (pp. 1–8). Austin, TX, USA.
- Kosara, R., & Miksch, S. (2001). Metaphors of movement: a visualization and user interface for time-oriented, skeletal plans. *Artificial Intelligence in Medicine*, 22(2), 111–131. doi:10.1016/S0933-3657(00)00103-2
- Kossiakoff, A., Sweet, W. N., Seymour, S., & Biemer, S. M. (2011). *Systems engineering principles and practices*. Skarpodd Arnold, Maryland, USA (2nd ed.). Wiley.
- Kostkova, P., Mani-Saada, J., Madle, G., & Weinberg, J. (2003). Agent-Based Up-to-date Data Management in National Electronic Library for Communicable Disease. In A. Moreno & J. L. Nealon (Eds.), *Applications of Software Agent Technology in the Health Care Domain* (pp. 105–124). Birkhäuser Basel. doi:10.1007/978-3-0348-7976-7_8
- Kowalski, R. A. (1988). The early years of logic programming. *Communications of the ACM*, 31(1), 38–43. doi:10.1145/35043.35046
- Krasner, G. E., & Pope, S. T. (1988). A Cookbook for Using the Model- View-Controller User Interface Paradigm in Smalltalk-80. *Journal of Object-Oriented Programming*, 1(3), 26–49.
- Kruchten, P. B. (1995). The 4+1 view model of architecture. *IEEE Software*, 12(6), 42–50. doi:10.1109/52.469759

- Krüger, M., Schäfer, A., Tewes, A., & Würtz, R. P. (2004). Communicating Agents Architecture with Applications in Multimodal Human Computer Interaction. In *Informatik verbindet* (pp. 641–645). Gesellschaft für Informatik.
- Kuhn, R. A., & Reider, R. S. (1994). A C++ framework for developing Medical Logic Modules and an Arden Syntax compiler. *Computers in Biology and Medicine*, 24(5), 365–370. doi:10.1016/0010-4825(94)90003-5
- Kuwabara, K., Ishida, T., & Osato, N. (1995a). AgenTalk: Coordination Protocol Description for Multiagent Systems. In V. Lesser (Ed.), *First International Conference on Multi-Agent Systems (ICMAS'95)* (p. 455). San Francisco, California, USA: AAAI/MIT Press.
- Kuwabara, K., Ishida, T., & Osato, N. (1995b). AgenTalk: Describing multiagent coordination protocols with inheritance. In *7th International Conference on Tools with Artificial Intelligence (ICTAI'95)* (Vol. 465, pp. 460–465). Herndon, VA: IEEE. doi:10.1109/TAI.1995.479841
- Labrou, Y. K., Finin, T., & Peng, Y. (1999). Agent communication languages: The current landscape. *IEEE Intelligent Systems*, 14(2), 45–52. doi:10.1109/5254.757631
- Lachmann, T., Berti, S., Kujala, T., & Schröger, E. (2005). Diagnostic subgroups of developmental dyslexia have different deficits in neural processing of tones and phonemes. *International Journal of Psychophysiology: Official Journal of the International Organization of Psychophysiology*, 56(2), 105–120. doi:10.1016/j.ijpsycho.2004.11.005
- Laganière, R. (2011). *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing.
- Lai, R., & Jirachiefpattana, A. (1998). SDL. In R. Lai & A. Jirachiefpattana (Eds.), *Communication Protocol Specification and Verification* (pp. 111–138). Springer US. doi:10.1007/978-1-4615-5549-0_5
- Lakbala, P., & Dindarloo, K. (2014). Physicians' perception and attitude toward electronic medical record. *SpringerPlus*, 3(63), 1–8. doi:10.1186/2193-1801-3-63
- Lalanne, D., Nigay, L., Palanque, P., Robinson, P., Vanderdonckt, J., & Ladry, J.-F. (2009). Fusion engines for multimodal input: a survey. In *International Conference on Multimodal Interfaces (ICMI-MLMI '09)* (pp. 153–160). doi:10.1145/1647314.1647343
- Laleci, G. B., Dogac, A., Olduz, M., Tasyurt, I., Yuksel, M., & Okcan, A. (2008). SAPHIRE: a multi-agent system for remote healthcare monitoring through computerized clinical guidelines. In R. Annicchiarico, U. Cortés, & C. Urdiales (Eds.), *Agent Technology and e-Health* (pp. 25–44). Birkhäuser Basel. doi:10.1007/978-3-7643-8547-7_3
- Lam, K.-M., & Yan, H. (1996). Locating and extracting the eye in human face images. *Pattern Recognition*, 29(5), 771–779.
- Lanzola, G., Gatti, L., Falasconi, S., & Stefanelli, M. (1999). A framework for building cooperative software agents in medical applications. *Artificial Intelligence in Medicine*, 16(3), 223–249. doi:10.1016/S0933-3657(99)00008-1
- Larousse. (2015). Dictionnaire de Français Larousse. Retrieved from <http://www.larousse.fr/dictionnaires/francais>

- Larsen, J. P., Høien, T., Lundberg, I., & Odegaard, H. (1990). MRI evaluation of the size and symmetry of the planum temporale in adolescents with developmental dyslexia. *Brain and Language*, 39(2), 289–301. doi:10.1016/0093-934X(90)90015-9
- Larson, B., Hatcliff, J., Procter, S., & Chalin, P. (2012). Requirements specification for apps in medical application platforms. In *Software Engineering in Health Care (SEHC), 2012* (pp. 26–32). IEEE. doi:10.1109/SEHC.2012.6227013
- Larsson, L., Nyström, M., Andersson, R., & Stridh, M. (2015). Detection of fixations and smooth pursuit movements in high-speed eye-tracking data. *Biomedical Signal Processing and Control*, 18, 145–152. doi:10.1016/j.bspc.2014.12.008
- LASER Process Working Group. (2006). *Little-JIL 1.5 Language Report*. Amherst.
- LASER Process Working Group. (2011). Little-JIL Tutorial. Retrieved from https://collab.cs.umass.edu/wiki/pages/K4S5s5/Little-JIL_Tutorial.html
- Lassus-Sangosse, D., N'guyen-Morel, M.-A., & Valdois, S. (2008). Sequential or simultaneous visual processing deficit in developmental dyslexia? *Vision Research*, 48(8), 979–88. doi:10.1016/j.visres.2008.01.025
- Latecki, L., & Ma, C. M. (1996). An Algorithm for a 3D Simplicity Test. *Computer Vision and Image Understanding*, 63(2), 388–393. doi:10.1006/cviu.1996.0028
- LaValle, S. M. (2006). Planning Algorithms, 1007. doi:10.1017/CBO9780511546877
- Le Bodic, L., De Loor, P., Calvet, G., & Kahn, J. (2004). SIHMM : Simulateur de l'Interaction Homme Machine Multimodale. In M. Noirhomme-Fraiture (Ed.), *16ème Conférence Francophone sur l'Interaction Homme-Machine (IHM'04), Annexes des actes* (pp. 31–34). Namur, Belgium: ACM Press.
- Le Bodic, L., De Loor, P., & Kahn, J. (2003). Multimodal interfaces evaluation with virtual reality simulation. In J. A. Jacko & C. Stephanidis (Eds.), *Human Computer Interaction, Theory and Practice (Part 1)* (pp. 661–665). CRC Press.
- Le Jan, G., Le Bouquin-Jeannès, R., Costet, N., Trolès, N., Scalart, P., Pichancourt, D., ... Gombert, J.-E. (2011). Multivariate predictive model for dyslexia diagnosis. *Annals of Dyslexia*, 61(1), 1–20. doi:10.1007/s11881-010-0038-5
- Le Jan, G., Trolès, N., Le Bouquin-Jeannès, R., Faucon, G., Gombert, J.-E., Scalart, P., & Pichancourt, D. (2007). Développement d'une plate-forme logicielle en vue de l'élaboration d'un outil d'aide au diagnostic de la dyslexie. *A.N.A.E.*, 91, 18–22.
- Lefavrais, P. (2005). *Alouette-R : Test d'analyse de la lecture et de la dyslexie* (ECPA.). Pearson.
- Leigh, R. J., & Kennard, C. (2004). Using saccades as a research tool in the clinical neurosciences. *Brain*, 127(3), 460–477. doi:10.1093/brain/awh035
- Lenz, R., Peleg, M., & Reichert, M. (2012). Healthcare process support: achievements, challenges, current research. *International Journal of Knowledge-Based Organizations*, 2(4), i–xv.
- Leonard, C. M., Voeller, K. K., Lombardino, L. J., Morris, M. K., Hynd, G. W., Alexander, A. W., ... Staab, E. V. (1993). Anomalous Cerebral Structure in Dyslexia Revealed With

- Magnetic Resonance Imaging. *Archives of Neurology*, 50(5), 461–9. doi:10.1001/archneur.1993.00540050013008
- Leonardi, G., Panzarasa, S., Quaglini, S., Stefanelli, M., & van der Aalst, W. M. P. (2007). Interacting agents through a web-based health serviceflow management system. *Journal of Biomedical Informatics*, 40(5), 486–499. doi:10.1016/j.jbi.2006.12.002
- Lerner, B. S., Osterweil, L. J., Sutton, S. M. J., & Wise, A. (1998). Programming process coordination in Little-JIL. In V. Gruhn (Ed.), *6th European Workshop on Software Process Technology (EWSPT'98)* (pp. 127–131). Weybridge, UK. doi:10.1007/3-540-64956-5_13
- Lesser, V., & Corkill, D. (2014). Challenges for multi-agent coordination theory based on empirical observations. In A. Lomuscio, P. Scerri, A. Bazzan, & M. Huhns (Eds.), *13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'14)* (pp. 1157–1160). Paris, France: International Foundation for Autonomous Agents and Multiagent Systems.
- Levick, D., & Osheroff, J. (2014). A Clinical Decision Support Implementation Guide: Practical Considerations. In R. A. Greenes (Ed.), *Clinical decision support: the road ahead* (2nd ed., pp. 689–709). Elsevier. doi:10.1016/B978-0-12-398476-0.00025-7
- Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7(1), 57–78. doi:10.1080/10447319509526110
- Leymann, F., & Roller, D. (1997). Workflow-based applications. *IBM Systems Journal*, 36(1), 102–123. doi:10.1147/sj.361.0102
- Li, D., Babcock, J. S., & Parkhurst, D. J. (2006). openEyes: a low-cost head-mounted eye-tracking solution. In K.-J. Rähä & A. T. Duchowski (Eds.), *Symposium on Eye Tracking Research & Applications (ETRA'06)* (pp. 95–100). San Diego, CA, USA: ACM Press. doi:10.1145/1117309.1117350
- Li, D., Winfield, D., & Parkhurst, D. J. (2005). Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (pp. 1–8). San Diego, CA, USA: IEEE. doi:10.1109/CVPR.2005.531
- Li, J., Wang, T., & Zhang, Y. (2011). Face Detection using SURF Cascade. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (pp. 2183–2190). Barcelona, Spain: IEEE. doi:10.1109/ICCVW.2011.6130518
- Li, J., & Zhang, Y. (2013). Learning SURF cascade for fast and accurate object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CPRV)* (pp. 3468–3475). Portland, Oregon: IEEE. doi:10.1109/CVPR.2013.445
- Liang, T.-P., Lai, H., Chen, N.-S., Wei, H., & Chen, M. C. (1994). When client/server isn't enough: coordinating multiple distributed tasks. *Computer*, 27(5), 73–79. doi:10.1109/2.291288
- Liao, P.-S., Chen, T.-S., & Chung, P.-C. (2001). A fast algorithm for multilevel thresholding. *Journal of Information Science and Engineering*, 17, 713–727.
- Lieberman, H., & Selker, T. (2003). Agents for the user interface. In J. Bradshaw (Ed.), *Handbook of Agent Technology* (pp. 1–21). AAAI/MIT Press.

- Lind, J. (2002). Specifying agent interaction protocols with standard UML. In M. J. Wooldridge, G. Weiß, & P. Ciancarini (Eds.), *Agent-Oriented Software Engineering II* (pp. 136–147). Montreal, Canada. doi:10.1007/3-540-70657-7_9
- Lind, J. (2003). Patterns in agent-oriented software engineering. In *Agent-Oriented Software Engineering III* (pp. 47–58). Springer Berlin Heidelberg. doi:10.1007/3-540-36540-0_4
- LIRMM. (2014). MaDKit: The Multiagent Development Kit. Montpellier, France: Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier. Retrieved from <http://www.madkit.org/>
- Lisle, R. J., & Robinson, J. M. (1995). The Mohr circle for curvature and its application to fold description. *Journal of Structural Geology*, 17(5), 739–750. doi:10.1016/0191-8141(94)00089-1
- Livingstone, M. S., Rosen, G. D., Drislane, F. W., & Galaburda, A. M. (1991). Physiological and anatomical evidence for a magnocellular defect in developmental dyslexia. *Proceedings of the National Academy of Sciences of the United States of America*, 88(18), 7943–7947. doi:10.1073/pnas.88.18.7943
- Lobach, D. F., & Hammond, W. E. (1997). Computerized decision support based on a clinical practice guideline improves compliance with care standards. *The American Journal of Medicine*, 102(1), 89–98. doi:10.1016/S0002-9343(96)00382-8
- Lobach, D. F., & Kerner, N. (2000). A systematic process for converting text-based guidelines into a linear algorithm for electronic implementation. In *AMIA Annual Symposium Proceedings* (pp. 507–511). Los Angeles, CA: American Medical Informatics Association.
- Lodderstedt, T., Basin, D., & Doser, J. (2002). SecureUML: A UML-based modeling language for model-driven security. In J.-M. Jézéquel, H. Hussmann, & S. Cook (Eds.), *UML 2002 — The Unified Modeling Language* (pp. 426–441). Dresden, Germany: Springer Berlin Heidelberg. doi:10.1007/3-540-45800-X_33
- Lohou, C., & Bertrand, G. (2002). A New 3D 6-Subiteration Thinning Algorithm Based on P - Simple Points. In *10th International Conference, Discrete Geometry for Computer Imagery (DGCI 2002)* (pp. 102–113). Bordeaux: Springer Berlin Heidelberg. doi:10.1007/3-540-45986-3_9
- Loomis, J. M., Blascovich, J. J., & Beall, A. C. (1999). Immersive virtual environment technology as a basic research tool in psychology. *Behavior Research Methods, Instruments, & Computers*, 31(4), 557–564.
- Lu, S., Bernstein, A., & Lewis, P. (2006). Automatic workflow verification and generation. *Theoretical Computer Science*, 353(1-3), 71–92. doi:10.1016/j.tcs.2005.10.035
- Luck, S. J. (2005). An Introduction to Event-Related Potentials and Their Neural Origins. In S. J. Luck (Ed.), *An Introduction to the Event-Related Potential Technique* (pp. 1–50). MIT Press.
- Luessi, M., Eichmann, M., Schuster, G. M., & Katsaggelos, A. K. (2009). Framework for efficient optimal multilevel image thresholding. *Journal of Electronic Imaging*, 18(1), 013004.
- M.D.P.H. de Guadeloupe. (2007). *La scolarisation des élèves en situation de handicap*.

- Ma, S. De. (1993). Conics-based stereo, motion estimation, and pose determination. *International Journal of Computer Vision*, 10(1), 7–25. doi:10.1007/BF01440844
- MacArthur, C. A., & Cavalier, A. R. (2004). Dictation and Speech Recognition Technology as Test Accommodations. *Exceptional Children*, 71(1), 43–58.
- Macaš, M. (2005). *Dyslexia Detection Using Artificial Neural Networks*. Czech Technical University in Prague Faculty of Electrical Engineering.
- Macdonald, S. J. (2007). *Hidden biographies: dyslexia, disability and social class*. University of Newcastle.
- Macdonald, S. J. (2010). Towards a social reality of dyslexia. *British Journal of Learning Disabilities*, 38(4), 271–279. doi:10.1111/j.1468-3156.2009.00601.x
- Madni, A. M., & Sievers, M. (2014). System of Systems Integration: Key Considerations and Challenges. *Systems Engineering*, 17(3), 330–347. doi:10.1002/sys.21272
- Magma Team. (2011). Magma | Research Team on Autonomous Agents and Multiagent Systems. Retrieved from <http://magma.imag.fr/>
- Magnan, A., & Ecalle, J. (2006). Audio-visual training in children with reading disabilities. *Computers & Education*, 46(4), 407–425. doi:10.1016/j.compedu.2004.08.008
- Magrabi, F., Lovell, N. H., Huynh, K., & Cell, B. G. (2001). Home telecare: system architecture to support chronic disease management. In *23rd Annual EMBS International Conference* (pp. 3559–3562). Istanbul, Turkey: IEEE. doi:10.1109/IEMBS.2001.1019601
- Mahmoud, Q. H., & Maamar, Z. (2006). Applying the MVC Design Pattern to Multi-Agent Systems. In *Canadian Conference on Electrical and Computer Engineering, CCECE'06* (pp. 2420–2423). IEEE. doi:10.1109/CCECE.2006.277427
- Mahmoud, Q. H., & Yu, L. (2005). An Architecture and Business Model for Making Software Agents Commercially Viable. In *The 38th Annual Hawaii International Conference on System Sciences* (pp. 1–9). IEEE. doi:10.1109/HICSS.2005.79
- Mahmoud, Q. H., & Yu, L. (2006). Making software agents user-friendly. *Computer*, 39(7), 93–95. doi:10.1109/MC.2006.239
- Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering*, 1(4), 267–284. doi:10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D
- Maini, E. S. (2006). Enhanced direct least square fitting of ellipses. *International Journal of Pattern Recognition and Artificial Intelligence*, 20(6), 939–953. doi:10.1142/S021800140600506X
- Mäkinen, E. (2007). *Face Analysis Techniques for Human-Computer Interaction*. University of Tampere, Finland.
- Makoul, G., Curry, R. H., & Tang, P. C. (2001). The use of electronic medical records: communication patterns in outpatient encounters. *Journal of the American Medical Informatics Association*, 8(6), 610–616. doi:10.1136/jamia.2001.0080610
- Malladi, R., Sethian, J. A., & Vemuri, B. C. (1995). Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2), 158–175. doi:10.1109/34.368173

- Maña, A., Montenegro, J. A., Sánchez, F., Ray, D., & Yagüe, M. (2003). Integrating & Automating Security Engineering In UML. In M. H. Hamza (Ed.), *IASTED International Conference on Communication, Network and Information Security (CNIS'03)* (pp. 1–6). New York, USA: International Association of Science and Technology for Development.
- Mandellos, G. J., Koutelakis, G. V., Panagiotakopoulos, T. C., Koukias, M. N., & Lymberopoulos, D. K. (2009). Requirements and solutions for advanced Telemedicine applications. In C. A. B. Barros de Mello (Ed.), *Biomedical Engineering* (pp. 1–15). InTech. doi:10.5772/7845
- Mansoux, B., Nigay, L., & Troccaz, J. (2006). Output multimodal interaction: the case of augmented surgery. In N. Bryan-Kinns, A. Blanford, P. Curzon, & L. Nigay (Eds.), *Human-Computer Interface HCI'06* (pp. 177–192). Londres, United Kingdom: Springer London. doi:10.1007/978-1-84628-664-3_14
- Martin, C. (2005). *Effets "top-down" lexicaux et attentionnels sur la reconnaissance de mots isolés en modalité visuelle*. Université Lumière Lyon II.
- Martinet, A. (1969). Fonctions du langage et linguistique appliquée. *Communication et Langages*, 1, 9–18. doi:10.3406/colan.1969.3705
- Martinez-Conde, S., Macknik, S. L., & Hubel, D. H. (2004). The role of fixational eye movements in visual perception. *Nature Reviews Neuroscience*, 5(3), 229–240. doi:10.1038/nrn1348
- Martins-Baltar, L., Laurillau, Y., & Calvary, G. (2013). Débridons l'interaction homme-machine pour une meilleure qualité des soins : Requis centrés utilisateurs et interfaces perlées pour les systèmes d'information de santé. *Ingénierie Des Systèmes d'Information*, 18(6), 113–139. doi:10.3166/isi.18.6.113-139
- Mateo, R. M. A., Cervantes, L. F., Yang, H.-K., & Lee, J. (2007). Mobile agents using data mining for diagnosis support in ubiquitous healthcare. In N. T. Nguyen, A. Grzech, R. J. Howlett, & L. C. Jain (Eds.), *Agent and Multi-Agent Systems: Technologies and Applications* (pp. 795–804). Wroclaw, Poland: Springer Berlin Heidelberg. doi:10.1007/978-3-540-72830-6_83
- Mathur, S., & Malik, S. (2010). Advancements in the V-Model. *International Journal of Computer Applications*, 1(12), 29–34. doi:10.5120/266-425
- Matusitz, J., & Breen, G. (2007). E-health: A new kind of telemedicine. *Social Work in Public Health*, 23(1), 95–114. doi:10.1300/J523v23n01
- Maurer, U., Bucher, K., Brem, S., Benz, R., Kranz, F., Schulz, E., ... Brandeis, D. (2009). Neurophysiology in Preschool Improves Behavioral Prediction of Reading Ability Throughout Primary School. *Biological Psychiatry*, 66(4), 341–348. doi:10.1016/j.biopsych.2009.02.031
- Mayer, R. J., Menzel, C. P., Painter, M. K., DeWitte, P. S., Blinn, T., & Perakath, B. (1995). *Information integration for concurrent engineering (IICE) IDEF3 process description capture method report*.
- McArthur, G. M., Hogben, J. H., Edwards, V. T., Heath, S. M., & Mengler, E. D. (2000). On the "specifics" of specific reading disability and specific language impairment. *Journal of Child Psychology and Psychiatry*, 41(7), 869–874. doi:10.1111/1469-7610.00674

- McCracken, D. D., & Jackson, M. A. (1982). Life cycle concept considered harmful. *ACM SIGSOFT Software Engineering Notes*, 7(2), 29–32. doi:10.1145/1005937.1005943
- McDonald, C., Huff, S., Deckard, J., Holck, K., & Vreeman, D. J. (2014). *Logical Observation Identifiers Names and Codes (LOINC®): Users' guide*.
- Mechanic, D. (2008). Rethinking Medical Professionalism: The Role of Information Technology and Practice Innovations. *The Milbank Quarterly*, 86(2), 327–358. doi:10.1111/j.1468-0009.2008.00523.x
- Mejía, C., Bull, S., Vatrappu, R., Florian, B., & Fabregat, R. (2012). PADA: a Dashboard of Learning Analytics for University Students with Dyslexia. In *Last ScandLE Seminar* (pp. 3–5). Copenhagen.
- Mejía, C., Clara, J., & Fabregat, R. (2011). DetectLD: Detecting university students with learning disabilities in reading and writing in the Spanish language. In T. Bastiaens & M. Ebner (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2011* (pp. 1122–1131). Chesapeake, VA: AACE.
- Mejía, C., Diaz, A., Jiménez, J. E., & Fabregat, R. (2012). BEDA: a computerized assessment battery for dyslexia in adults. *Procedia - Social and Behavioral Sciences*, 46, 1795–1800. doi:10.1016/j.sbspro.2012.05.381
- Mejía, C., Díaz, A., Jiménez, J. E., & Fabregat, R. (2011). Considering cognitive traits of university students with dyslexia in the context of a learning management system. In D. D. Schmorrow & C. M. Fidopiastis (Eds.), *Foundations of Augmented Cognition. Directing the Future of Adaptive Systems* (Lecture No., pp. 432–441). Springer Berlin Heidelberg. doi:10.1007/978-3-642-21852-1_50
- Mejía Corredor, C., & Fabregat Gesa, R. (2012). Framework for Intervention and Assistance in University Students with Dyslexia. In *2012 IEEE 12th International Conference on Advanced Learning Technologies* (pp. 342–343). Ieee. doi:10.1109/ICALT.2012.170
- Mele, M. L., & Federici, S. (2012a). A psychotechnological review on eye-tracking systems: towards user experience. *Disability and Rehabilitation. Assistive Technology*, 7(4), 261–81. doi:10.3109/17483107.2011.635326
- Mele, M. L., & Federici, S. (2012b). Gaze and eye-tracking solutions for psychological research. *Cognitive Processing*, 13 Suppl 1, 261–5. doi:10.1007/s10339-012-0499-z
- Mendeley Ltd. (2014). Mendeley. Retrieved from <http://www.mendeley.com>
- Meurisse, T. (2004). *Simulation multi-agent : du modèle à l'opérationnalisation*. Université de Paris 6.
- Microsoft. (2007a). Office Excel. Retrieved from <http://office.microsoft.com/excel>
- Microsoft. (2007b). Office PowerPoint. Retrieved from <http://office.microsoft.com/powerpoint>
- Microsoft. (2007c). Office Visio. Retrieved from <http://visio.microsoft.com>
- Microsoft. (2007d). Office Word. Retrieved from <http://office.microsoft.com/word/>
- Microsoft. (2008a). Microsoft Foundation Class Library. Microsoft. Retrieved from [http://msdn.microsoft.com/en-us/library/d06h2x6e\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/d06h2x6e(v=vs.90).aspx)

- Microsoft. (2008b). Visual Studio 2008 Professional Edition. Retrieved from <http://www.microsoft.com/visualstudio>
- Microsoft. (2010). Windows API. Microsoft. Retrieved from <http://msdn.microsoft.com/library/cc433218.aspx>
- Microsoft. (2015). Bing. Retrieved from <http://www.bing.com>
- Miksch, S., Kosara, R., Shahar, Y., & Johnson, P. (1998). AsbruView: Visualization of Time-Oriented, Skeletal Plans. In *Fourth International Conference on Artificial Intelligence Planning Systems 1998 (AIPS-98)* (Vol. 1998, pp. 11–18). Pittsburgh, Pennsylvania, USA: AAAI Press.
- Miksch, S., Shahar, Y., & Johnson, P. D. (1997). Asbru: a task-specific, intention-based, and time-oriented language for representing skeletal plans. In E. Motta, F. van Harmelen, C. von Pierret-Golbreich, I. Filby, & N. Wijngaards (Eds.), *7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97)* (pp. 1–25). Milton Keynes, UK.
- Ministère des affaires sociales et de la santé. (2014). Dossier Médical Personnel : Votre carnet de santé informatisé. Retrieved from <http://www.dmp.gouv.fr/>
- Mioduser, D., Tur-Kaspa, H., & Leitner, I. (2000). The learning value of computer-based instruction of early reading skills. *Journal of Computer Assisted Learning*, 16(1), 54–63. doi:10.1046/j.1365-2729.2000.00115.x
- Misès, R. (2012). *Classification française des troubles mentaux de l'enfant et de l'adolescent - R-2012* (5ème ed.). Ecole des Hautes Etudes en Santé Publique.
- Mitrović, N., & Mena, E. (2003). Improving user interface usability using mobile agents. In J. A. Jorge, N. J. Nunes, & J. F. e Cunha (Eds.), *Interactive Systems. Design, Specification, and Verification* (pp. 273–287). Funchal, Madeira Island, Portugal: Springer Berlin Heidelberg. doi:10.1007/978-3-540-39929-2_19
- Miyoshi, T., & Murata, A. (2001a). Input device using eye tracker in human-computer interaction. In *Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication* (pp. 580–585). Ieee. doi:10.1109/ROMAN.2001.981967
- Miyoshi, T., & Murata, A. (2001b). Usability of input device using eye tracker on button size, distance between targets and direction of movement. In *button size, distance between targets and direction of movement. IEEE International Conference on Systems, Man, and Cybernetics: ICSMC'01* (pp. 227–232). IEEE.
- Moita, F., Oliveira, R., Santos, V., & Silva, M. (2012). Development of interfaces for impaired users. *Przegląd Elektrotechniczny*, (1a), 127–131.
- Montagnat, J., Delingette, H., & Ayache, N. (2001). A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing*, 19, 1023–1040.
- Morchio, B., & Ott, M. (1989). Using computers in the diagnosis and remediation of developmental dyslexia. In C. Bell, J. Davies, & R. Winders (Eds.), *Promoting Learning (Aspects of Educational and Training Technology)* (pp. 231–236). Nichols Pub Co.
- Morchio, B., Ott, M., Pesenti, E., & Tavella, M. (1990). Developmental dyslexia: a computer based diagnostic test. In *International Conference on the role and impact of information technology in the assessment & rehabilitation of individuals with communication disorders* (pp. 1–17). Thessaloniki, Greece.

- Moreno, A., Sánchez, D., & Isern, D. (2003). Security measures in a medical multi-agent system. In J. L. Valverde, I. Aguiló, & M. T. Escrig (Eds.), *Artificial Intelligence Research and Development* (pp. 244–255). IOS Press.
- Morimoto, C. H., & Mimica, M. R. M. (2005). Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98(1), 4–24. doi:10.1016/j.cviu.2004.07.010
- Morley, C., Bia-Figueiredo, M., & Gillette, Y. (2011). *Processus métiers et S.I. : Gouvernance, management, modélisation* (3ème ed.). Dunod.
- Moser, W., Heller, U., Gries, F. A., Engelbrecht, R., & DIADOQ. (2001). Enabling Guidelines for Computer-based Decision Support - Process and Tools. In V. L. Patel, R. Rogers, & R. Haux (Eds.), *MEDINFO 2001: Proceedings of the 10th World Congress on Medical Informatics* (Vol. 84, pp. 231–235). IOS Press. doi:10.3233/978-1-60750-928-8-231
- Mounin, G. (2004). *Dictionnaire de la linguistique*. (Presses Universitaires de France, Ed.) *Lingua* (4ème ed., Vol. 33).
- Mousty, P., & Leybaert, J. (1999). Évaluation des habiletés de lecture et d'orthographe au moyen de BELEC : données longitudinales auprès d'enfants francophones testés en 2ème et 4ème années. *Revue Européenne de Psychologie Appliquée*, 49(5), 325–342.
- Müller, H., Maurer, H., Reihls, R., Sauer, S., & Zatloukal, K. (2011). Adaptive visual symbols for personal health records. In *15th International Conference on Information Visualisation (IV 2011)* (pp. 220–225). London, UK: IEEE. doi:10.1109/IV.2011.87
- Müller, H., Sauer, S., Zatloukal, K., & Bauernhofer, T. (2010). Interactive patient records. In *14th International Conference Information Visualisation (IV 2010)* (pp. 252–257). London, UK: IEEE. doi:10.1109/IV.2010.45
- Munassar, N. M. A., & Govardhan, A. (2010). A Comparison Between Five Models Of Software Engineering. *IJCSI*, 7(5), 94–101.
- Murawski, K. (2010). Method for determining the position of the pupil for eyetracking applications. In *Methods and Models in Automation and Robotics* (pp. 356–362). IEEE. doi:10.1109/MMAR.2010.5587207
- Murawski, K., & Rózanowski, K. (2013). Pattern Recognition Algorithm for Eye Tracker Sensor Video Data Analysis. *Acta Physica Polonica A*, 124(3), 509–512. doi:10.12693/APhysPolA.124.509
- Musen, M. A. (1998). Domain ontologies in software engineering: use of Protege with the EON architecture. *Methods Archive*, 37(4-5), 540–550.
- Musen, M. A., Tu, S. W., Das, A. K., & Shahar, Y. (1996). EON: A component-based approach to automation of protocol-directed therapy. *Journal of the American Medical Informatics Association*, 3(6), 367–388.
- Myers, B. A. (1998). A Brief History of Interaction Technology. *Interactions*, 5(2), 44–54. doi:10.1145/274430.274436
- Myers, B. A., Hudson, S. E., & Pausch, R. (2000). Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction*, 7(1), 3–28. doi:10.1145/344949.344959

- Mylopoulos, J., Kolp, M., & Castro, J. (2001). UML for agent-oriented software development: The Tropos proposal. In M. Gogolla & C. Kobryn (Eds.), *«UML» 2001 — The Unified Modeling Language. Modeling Languages, Concepts, and Tools* (pp. 422–441). Toronto, Canada: Springer Berlin Heidelberg. doi:10.1007/3-540-45441-1_31
- Nagamatsu, T., Kamahara, J., & Tanaka, N. (2008). 3D gaze tracking with easy calibration using stereo cameras for robot and human communication. In *The 17th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2008)* (pp. 59–64). Munich, Germany: IEEE. doi:10.1109/ROMAN.2008.4600643
- Nagamatsu, T., Kamahara, J., & Tanaka, N. (2009). Calibration-free gaze tracking using a binocular 3D eye model. In *The 27th International Conference on Human Factors in Computing Systems - Extended Abstracts (CHI EA'09)* (pp. 3613–3618). Boston, MA, USA: ACM Press. doi:10.1145/1520340.1520543
- Nagaratnam, N., & Lea, D. (1998). Role-based Protection and Delegation for Mobile Object Environments. In S. Demeyer & J. Bosch (Eds.), *ECOOP' 98 Workshops, Demos, and Posters* (pp. 1–12). Brussels, Belgium: Springer Berlin Heidelberg.
- Nahm, Y.-E., & Ishikawa, H. (2005). A hybrid multi-agent system architecture for enterprise integration using computer networks. *Robotics and Computer-Integrated Manufacturing*, 21(3), 217–234. doi:10.1016/j.rcim.2004.07.016
- Nanci, D., & Espinasse, B. (2001). *Ingenierie des systemes d'information : Merise, deuxieme generation* (4ème ed.). Vuibert.
- National Electrical Manufacturers Association. (2014). DICOM Homepage. Retrieved from <http://medical.nema.org/standard.html>
- National Electrical Manufacturers Association. (2015). NEMA - The Association of Electrical Equipment and Medical Imaging Manufacturers. Retrieved from <http://www.nema.org>
- National Institute of Standards and Technology. (2014). Role Based Access Control and Role Based Security. Retrieved from <http://csrc.nist.gov/groups/SNS/rbac/>
- National Transitions of Care Coalition. (2010). *Improving Transitions of Care with Health Information Technology*. Washington DC.
- Naylor, K. T., Kudlow, P., Li, F., & Yuen, K. (2011). Improving healthcare with information technology. *The University of Western Ontario Medical Journal*, 80(1), 17–19.
- Nealon, J. L., & Moreno, A. (2002). The application of agent technology to health care. In *Workshop AgentCities: Research in Large-scale Open Agent Environments in 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)* (pp. 169–173). Bologna, Italy.
- Nealon, J. L., & Moreno, A. (2003). Agent-Based Applications in Health Care. In A. Moreno & J. L. Nealon (Eds.), *Applications of Software Agent Technology in the Health Care Domain* (pp. 3–18). Birkhäuser Basel. doi:10.1007/978-3-0348-7976-7_2
- Neill, C. J., & Laplante, P. A. (2003). Requirements Engineering: The State of the Practice. *IEEE Software*, 20(6), 40–45. doi:10.1109/MS.2003.1241365
- Nenert, R., Leveque, C., Lenormand, M. T., Evrard, P., & De Schonen, S. (2009). Global effects of three remediation programs on children with developmental dyslexia: an fMRI study.

- In *Organization for Human Brain Mapping 15th Annual Meeting* (Vol. 20). San Francisco, CA, USA. doi:10.1016/S1053-8119(09)70779-7
- Ng, T. H., Cheung, S. C., Chan, W. K., & Yu, Y. T. (2006). Work experience versus refactoring to design patterns: a controlled experiment. In *14th ACM SIGSOFT international symposium on Foundations of Software Engineering* (pp. 12–22). ACM Press. doi:10.1145/1181775.1181778
- Nguyen, H. (2003). Vers une architecture générique de système de dialogue oral homme-machine. In *RÉCITAL 2003* (pp. 539–545). Bataz-sur-Mer.
- Nguyen, J. H., Shahar, Y., Tu, S. W., Das, A. K., & Musen, M. A. (1997). A temporal database mediator for protocol-based decision support. In *AMIA Annual Symposium Proceedings* (pp. 298–302).
- Nguyen, J. H., Shahar, Y., Tu, S. W., Das, A. K., & Musen, M. A. (1999). Integration of temporal reasoning and temporal-data maintenance into a reusable database mediator to answer abstract, time-oriented queries: The Tzolkín system. *Journal of Intelligent Information Systems*, 13(1-2), 121–145. doi:10.1023/A:1008711006591
- Nicolson, R. I., & Fawcett, A. J. (2007). Procedural learning difficulties: reuniting the developmental disorders? *Trends in Neurosciences*, 30(4), 135–41. doi:10.1016/j.tins.2007.02.003
- Niessen, W. J., ter Haar Romeny, B. M., & Viergever, M. A. (1998). Geodesic deformable models for medical image analysis. *IEEE Transactions on Medical Imaging*, 17(4), 634–641. doi:10.1109/42.730407
- Nigay, L. (1994). *Conception et modélisation logicielles des systèmes interactifs : application aux interfaces multimodales*. Université Joseph Fourier - Grenoble 1.
- Nigay, L., Bouchet, J., Juras, D., Mansoux, B., Ortega, M., Serrano, M., & Lawson, J.-Y. L. (2008). Software Engineering for Multimodal Interactive Systems. In D. Tzovaras (Ed.), *Multimodal User Interfaces* (Vol. 2, pp. 201–218). Springer Berlin Heidelberg. doi:10.1007/978-3-540-78345-9_9
- Nigay, L., & Coutaz, J. (1993). A design space for multimodal systems: concurrent processing and data fusion. In *INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (pp. 172–178). ACM Press. doi:10.1145/169059.169143
- Nigay, L., & Coutaz, J. (1995). A generic platform for addressing the multimodal challenge. In *SIGCHI Conference on Human Factors in Computing Systems (CHI'95)* (pp. 98–105). Denver, USA: ACM Press/Addison-Wesley Publishing. doi:10.1145/223904.223917
- Nixon, M. S., & Aguado, A. S. (2002). *Feature Extraction and Image Processing*. Oxford: Newnes.
- Nodine, M., Fowler, J., Ksiezzyk, T., Perry, B., Taylor, M., & Unruh, A. (2000). Active information gathering in InfoSleuth. *International Journal of Cooperative Information Systems*, 9(1-2), 1–27. doi:10.1142/S021884300000003X
- Noël, V. (2012). *Multi-Agent Systems and Component-based Software Architectures: Mutual and Complementary Contributions for Supporting Software Development*. Université Toulouse III - Paul Sabatier.

- Noureddin, B., Lawrence, P. D., & Man, C. F. (2005). A non-contact device for tracking gaze in a human computer interface. *Computer Vision and Image Understanding*, 98(1), 52–82. doi:10.1016/j.cviu.2004.07.005
- Novák, D., Kordík, P., Macaš, M., Vyhnaček, M., Brzezny, R., & Lhotská, L. (2004). School children dyslexia analysis using self organizing maps. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 1–4). San Francisco, CA, USA: IEEE. doi:10.1109/IEMBS.2004.1403075
- Novák, P., Rollo, M., Hodík, J., & Vlček, T. (2003). Communication Security in Multi-agent Systems. In V. Mařík, M. Pěchouček, & J. Müller (Eds.), *Multi-Agent Systems and Applications III* (pp. 454–463). Prague, Czech Republic: Springer Berlin Heidelberg. doi:10.1007/3-540-45023-8_44
- Nuance. (2015). Dragon NaturallySpeaking - world's best-selling speech recognition software - Nuance. Retrieved from <http://www.nuance.com/dragon>
- Nurcan, S. (1996). Analyse et conception de systèmes d'information coopératifs. *Techniques et Science Informatiques*, 15(9), 1287–1315.
- Nwana, H. S. (1996). Software agents: An overview. *Knowledge Engineering Review*, 11(03), 205–244. doi:10.1017/S026988890000789X
- Nwana, H. S., Lee, L. C., & Jennings, N. R. (1996). Co-ordination in software agent systems. *BT Technology Journal*, 14(4), 79–89.
- Nwana, H. S., Ndumu, D. T., Lee, L. C., & Collis, J. C. (1999). ZEUS: a toolkit for building distributed multiagent systems. *Applied Artificial Intelligence*, 13(1-2), 129–185. doi:10.1080/088395199117513
- Nyström, M., & Holmqvist, K. (2010). An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior Research Methods*, 42(1), 188–204. doi:10.3758/BRM.42.1.188
- O'Regan, S., & Marnane, W. (2013). Multimodal detection of head-movement artefacts in EEG. *Journal of Neuroscience Methods*, 218(1), 110–120. doi:10.1016/j.jneumeth.2013.04.017
- Object Management Group. (2014a). Business Process Model & Notation (BPMN): Resource Page. Retrieved from <http://www.omg.org/bpmn/>
- Object Management Group. (2014b). Data Distribution Service Portal: The Open, Multiplatform, Interoperable Publish-Subscribe Middleware Standard. Retrieved from <http://portals.omg.org/dds/>
- Object Management Group. (2014c). MDA - The Architecture of Choice for a Changing World. Retrieved from <http://www.omg.org/mda/>
- Object Management Group. (2014d). OMG's CORBA Website. Retrieved from <http://www.corba.org/>
- Object Management Group. (2014e). Unified Modeling Language™ (UML®): Resource Page. Retrieved from <http://www.uml.org/>
- Obrenović, Ž., & Starčević, D. (2004). Modeling multimodal human-computer interaction. *Computer*, 37(9), 65–72. doi:10.1109/MC.2004.139

- Obrenović, Ž., & Starčević, D. (2006). Adapting the unified software development process for user interface development. *Computer Science and Information Systems*, 3(1), 33–52. doi:10.2298/CSIS0601033O
- Occello, M. (2003). *Méthodologie et Architectures pour la Conception de Systèmes Multi-Agents*. UNIVERSITÉ JOSEPH FOURIER DE GRENOBLE.
- Occello, M., Baeijs, C., Koning, J.-L., & Demazeau, Y. (2003). La plate-forme MASK. Retrieved from <http://www-poleia.lip6.fr/~guessoum/asa/afia/mask.html>
- Odell, J., Parunak, H. V. D., & Bauer, B. (2000). Representing Agent Interaction Protocols in UML. In *AAAI Agents 2000 conference*. Barcelona, SPain: AAAI Press.
- Odell, J., Parunak, H. V. D., & Bauer, B. (2003). Extending UML for Agent-Based Systems. In H. Kilov & K. Baclawski (Eds.), *Practical Foundations of Business System Specifications* (pp. 245–270). Springer Netherlands. doi:10.1007/978-94-017-2740-2_13
- Ohene-Djan, J., & Begum, R. (2008). Multisensory Games for Dyslexic Children. In *Advanced Learning Technologies for Disabled and Non-Disabled People* (pp. 1040–1041). IEEE. doi:10.1109/ICALT.2008.98
- Ohno-Machado, L., Gennari, J., Murphy, S. N., Jain, N. L., Tu, S. W., Oliver, D. E., ... Barnett, G. O. (1998). The GuideLine Interchange Format: A Model for Representing Guidelines. *Journal of the American Medical Informatics Association*, 5(4), 357–372. doi:10.1136/jamia.1998.0050357
- Oja, M., Tamm, B., & Taveter, K. (2001). Agent-based software design. *Proceedings of the Estonian Academy of Sciences, Engineering*, 7(1), 5–21.
- Olson, C. K. (2010). Children's motivations for video game play in the context of normal development. *Review of General Psychology*, 14(2), 180–187. doi:10.1037/a0018984
- Olsson, P. (2007). *Real-time and Offline Filters for Eye Tracking*. KTH Royal Institute of Technology (Stockholm).
- Oostenveld, R., & Praamstra, P. (2001). The five percent electrode system for high-resolution EEG and ERP measurements. *Clinical Neurophysiology*, 112(4), 713–719. doi:10.1016/S1388-2457(00)00527-7
- openEHR Foundation. (2015). Welcome to openEHR. Retrieved from <http://www.openehr.org/>
- Oracle Corporation. (2015). MySQL's homepage. Retrieved from <http://www.mysql.com/>
- Orban de Xivry, J.-J., & Lefèvre, P. (2007). Saccades and pursuit: two outcomes of a single sensorimotor process. *The Journal of Physiology*, 584(1), 11–23. doi:10.1113/jphysiol.2007.139881
- Organization for the Advancement of Structured Information Standards. (2014). The online community gathering place and information resource for the WS-BPEL OASIS Standard and related specifications. Retrieved from <http://bpel.xml.org/>
- Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1), 12–49. doi:10.1016/0021-9991(88)90002-2

- Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62–66.
- Oviatt, S. (2007). Multimodal Interfaces. In A. Sears & J. A. Jacko (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications* (2nd ed., pp. 413–432). CRC Press. doi:10.1201/9781410615862.ch21
- Oviatt, S., & Cohen, P. (2000). Perceptual User Interfaces: Multimodal Interfaces That Process What Comes Naturally. *Communications of the ACM*, 43(3), 45–53. doi:10.1145/330534.330538
- Ow, S. H. (2012). Detecting Dyslexia in Children Using a Computer-Aided. *New Educational Review*, 30(4), 259–270.
- Owen, M., & Raj, J. (2003). *BPMN and Business Process Management: Introduction to the New Business Process Modeling Standard*.
- Padgham, L., & Winikoff, M. (2003). Prometheus: A methodology for developing intelligent agents. In F. Giunchiglia, J. Odell, & G. Weiß (Eds.), *Agent-Oriented Software Engineering III* (pp. 174–185). Bologna, Italy: Springer Berlin Heidelberg. doi:10.1007/3-540-36540-0_14
- Pammer, K. (2014). Brain Mechanisms and Reading Remediation: More Questions Than Answers. *Scientifica*, 2014(802741), 1–9. doi:10.1155/2014/802741
- Pashler, H. (1993). Doing two things at the same time. *American Scientist*, 81(1), 48–55.
- Pasqualotto, E., Federici, S., & Olivetti Belardinelli, M. (2012). Toward functioning and usable brain-computer interfaces (BCIs): A literature review. *Disability and Rehabilitation: Assistive Technology*, 7(2), 89–103. doi:10.3109/17483107.2011.589486
- Pattison-Gordon, E., Cimino, J. J., Hripcsak, G., Tu, S. W., Gennari, J. H., Jain, N. L., & Greenes, R. A. (1996). *Requirements of a sharable guideline representation for computer applications (SMI-96-0628)*.
- Paurobally, S., & Cunningham, J. (2003). Achieving common interaction protocols in open agent environments. In *2nd international workshop on Challenges in Open Agent Environments at AAMAS'03* (pp. 1–4). Melbourne, Australia.
- Pavón, J., & Gómez-Sanz, J. (2003). Agent oriented software engineering with INGENIAS. In V. Mařík, M. Pěchouček, & J. Müller (Eds.), *Multi-Agent Systems and Applications III* (pp. 394–403). Springer Berlin Heidelberg. doi:10.1007/3-540-45023-8_38
- Peleg, M., Boxwala, A. A., Bernstam, E., Tu, S. W., Greenes, R. A., & Shortliffe, E. H. (2001). Sharable Representation of Clinical Guidelines in GLIF: Relationship to the Arden Syntax. *Journal of Biomedical Informatics*, 34(3), 170–81. doi:10.1006/jbin.2001.1016
- Peleg, M., & González-Ferrer, A. (2014). Guidelines and Workflow Models. In R. A. Greenes (Ed.), *Clinical decision support: the road ahead* (2nd ed., pp. 435–464). Elsevier. doi:10.1016/B978-0-12-398476-0.00016-6
- Peng, Y., Finin, T., Labrou, Y. K., Chu, B., Long, J., Tolone, W., & Boughannam, A. (1998). A multi-agent system for enterprise integration. *International Journal of Agile Manufacturing*, 1(2), 213–229.

- Pérez, A., L.Córdoba, M., García, A., Méndez, R., Muñoz, M. L., Pedraza, J. L., & Sánchez, F. (2003). A precise eye-gaze detection and tracking system. *The 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2003*.
- Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence*, 12(7), 629–639. doi:10.1109/34.56205
- Perry, C., Ziegler, J. C., & Zorzi, M. (2007). Nested incremental modeling in the development of computational theories: the CDP+ model of reading aloud. *Psychological Review*, 114(2), 273–315. doi:10.1037/0033-295X.114.2.273
- Perry, C., Ziegler, J. C., & Zorzi, M. (2010). Beyond single syllables: large-scale modeling of reading aloud with the Connectionist Dual Process (CDP++) model. *Cognitive Psychology*, 61(2), 106–51. doi:10.1016/j.cogpsych.2010.04.001
- Persky, S. (2011). Employing immersive virtual environments for innovative experiments in health care communication. *Patient Education and Counseling*, 82(3), 313–317. doi:10.1016/j.pec.2010.12.007
- Picard, G. (2004). *Méthodologie de développement de systèmes multi-agents adaptatifs et conception de logiciels à fonctionnalité émergente*. Université Toulouse III - Paul Sabatier.
- Picard, G., & Gleizes, M.-P. (2004). The ADELFE Methodology: Designing Adaptive Cooperative Multi-Agent Systems. In F. Bergenti, M.-P. Gleizes, & F. Zambonelli (Eds.), *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering handbook* (pp. 157–176). Kluwer Publishing.
- Picard, R. W. (2000). Perceptual user interfaces: Affective Perception. *Communications of the ACM*, 43(3), 50–51. doi:10.1145/330534.330539
- Pichler, F. (2000). Modeling complex systems by multi-agent holarchies. In *Computer Aided Systems Theory-EUROCAST'99* (pp. 154–168). Springer Berlin Heidelberg. doi:10.1007/10720123_14
- Poggi, A., Rimassa, G., & Tomaiuolo, M. (2001). Multi-user and security support for multi-agent systems. In A. Omicini & M. Viroli (Eds.), *WOA 2001 Workshop* (pp. 8–13). Modena, Italy: Pitagora Editrice Bologna.
- Porumb, C., Porumb, S., Orza, B., & Budura, D. (2010). Computer-Supported Collaborative Work and Its Application to E-Health. In *2010 Third International Conference on Advances in Mesh Networks* (pp. 75–80). IEEE. doi:10.1109/MESH.2010.22
- Poslad, S. (2007). Specifying protocols for multi-agent systems interaction. *ACM Transactions on Autonomous and Adaptive Systems*, 2(4 (15)), 1–24. doi:10.1145/1293731.1293735
- Poslad, S., Charlton, P., & Calisti, M. (2003). Specifying standard security mechanisms in multi-agent systems. In R. Falcone, S. Barber, L. Korba, & M. Singh (Eds.), *Trust, Reputation, and Security: Theories and Practice* (pp. 163–176). Bologna, Italy,: Springer Berlin Heidelberg. doi:10.1007/3-540-36609-1_13
- Potel, M. (1996). *MVP: Model-View-Presenter the Taligent programming model for C++ and Java*.

- Powell, S. M., & Hohenhaus, S. M. (2006). Multidisciplinary Team Training and the Art of Communication. *Clinical Pediatric Emergency Medicine*, 7(4), 238–240. doi:10.1016/j.cpem.2006.08.007
- Prado, C., Dubois, M., & Valdois, S. (2007). The eye movements of dyslexic children during reading and visual search: impact of the visual attention span. *Vision Research*, 47(19), 2521–30. doi:10.1016/j.visres.2007.06.001
- Preston, K. J. (1981). Some notes on cellular logic operators. *Pattern Analysis and Machine Intelligence*, PAMI-3(4), 476–481.
- Prezi Inc. (2014). Prezi - Logiciel de présentation. Prezi Inc.
- Priestley, M., & Utt, M. H. (2000). A unified process for software and documentation development. In *2000 Joint IEEE International and 18th Annual Conference on Computer Documentation* (pp. 221–238). Cambridge, MA: IEEE. doi:10.1109/IPCC.2000.887279
- Prokosch, H.-U., Ries, M., Beyer, A., Schwenk, M., Seggewies, C., Köpcke, F., ... Bürkle, T. (2011). IT infrastructure components to support clinical care and translational research projects in a comprehensive cancer center. *Studies in Health Technology and Informatics*, 169, 892–896. doi:10.3233/978-1-60750-806-9-892
- Protti, D. (2009). Integrated care needs integrated information management and technology. *Healthcare Quarterly*, 13(Sp), 24–29. doi:10.12927/hcq.2009.21093
- Purves, I. N., Sugden, B., Booth, N., & Sowerby, M. (1999). The PRODIGY project - the iterative development of the release one model. In *AMIA Annual Symposium Proceedings* (pp. 359–363). American Medical Informatics Association.
- Purvis, M., Purvis, M., Haidar, A., & Savarimuthu, B. T. R. (2005). A distributed workflow system with autonomous components. In M. W. Barley & N. Kasabov (Eds.), *7th Pacific Rim International Workshop on Multi-Agents (PRIMA 2004)* (pp. 193–205). Auckland, New Zealand: Springer Berlin Heidelberg. doi:10.1007/978-3-540-32128-6_15
- Purvis, M., Savarimuthu, B. T. R., & Purvis, M. (2004). A multi-agent based workflow system embedded with web services. In A. Ghobani & S. Marsh (Eds.), *2nd international workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments (COLA 2004)* (pp. 55–62). Beijing, China: IEEE/WIC Press.
- Qiu, X. (2004). Building desktop applications with Web services in a message-based MVC paradigm. In *IEEE International Conference on Web Services* (pp. 765–768). IEEE. doi:10.1109/ICWS.2004.1314812
- Qiu, X. (2005). *Message-based MVC architecture for distributed and desktop applications*. Syracuse University.
- Qiu, X., Pallickara, S., & Uyar, A. (2004). Making SVG a web service in a message-based MVC architecture. In *SVG Open Conference* (pp. 1–12). Tokyo, Japan.
- Quaglini, S., Stefanelli, M., Cavallini, A., Micieli, G., Fassino, C., & Mossa, C. (2000). Guideline-based careflow systems. *Artificial Intelligence in Medicine*, 20(1), 5–22. doi:10.1016/S0933-3657(00)00050-6
- Quality Systems Inc. (2014). Mirth Connect. Quality Systems Inc. Retrieved from <http://www.mirthcorp.com/products/mirth-connect>

- Raberger, T., & Wimmer, H. (2003). On the automaticity/cerebellar deficit hypothesis of dyslexia: balancing and continuous rapid naming in dyslexic and ADHD children. *Neuropsychologia*, *41*(11), 1493–1497. doi:10.1016/S0028-3932(03)00078-2
- Rahimi, B., Vimarlund, V., & Timpka, T. (2009). Health information system implementation: A qualitative meta-analysis. *Journal of Medical Systems*, *33*(5), 359–368. doi:10.1007/s10916-008-9198-9
- Ram, P., Berg, D., Tu, S. W., Mansfield, G., Ye, Q., Abarbanel, R., & Beard, N. (2004). Executing Clinical Practice Guidelines using the SAGE Execution Engine. In M. Fieschi, E. Coiera, & Y.-C. J. Li (Eds.), *MEDINFO 2004: Proceedings of the 11th World Congress on Medical Informatics* (pp. 251–255). IOS Press. doi:10.3233/978-1-60750-949-3-251
- Ramírez-Cortes, J. M., Alarcon-Aquino, V., Rosas-Cholula, G., Gomez-Gil, P., & Escamilla-Ambrosio, J. (2010). P-300 Rhythm Detection Using ANFIS Algorithm and Wavelet Feature Extraction in EEG Signals. In *The World Congress on Engineering and Computer Science (WCECS 2010)* (pp. 619–623). San Francisco, CA, USA: Newswood Limited.
- Ramsin, R., & Paige, R. F. (2008). Process-Centered Review of Object Oriented Software Development Methodologies. *ACM Computing Surveys*, *40*(1 (article 3)), 1–89. doi:10.1145/1322432.1322435
- Ramus, F. (2003). Developmental dyslexia: specific phonological deficit or general sensorimotor dysfunction? *Current Opinion in Neurobiology*, *13*(2), 212–8. doi:10.1016/S0959-4388(03)00035-7
- Ramus, F. (2010). Génétique de la dyslexie développementale. In S. Chokron & J.-F. Démonet (Eds.), *Approche neuropsychologique des troubles des apprentissages* (pp. 67–90). Marseille: Solal.
- Ramus, F. (2012). Les troubles spécifiques de la lecture. *L'Information Grammaticale*, *133*, 34–40. doi:10.2143/IG.133.0.2159025
- Ramus, F., Pidgeon, E., & Frith, U. (2003). The relationship between motor control and phonology in dyslexic children. *Journal of Child Psychology and Psychiatry*, *44*(5), 712–722. doi:10.1111/1469-7610.00157
- Rao, A. S., & Georgeff, M. P. (1991). *Modeling rational agents within a BDI-architecture*.
- Rao, A. S., & Georgeff, M. P. (1995). *BDI Agents: From Theory to Practice*.
- Rational Software. (2001). *Rational Unified Process: Best Practices for Software Development Teams*.
- Ray, A., & Srivastava, D. C. (2008). Non-linear least squares ellipse fitting using the genetic algorithm with applications to strain analysis. *Journal of Structural Geology*, *30*(12), 1593–1602. doi:10.1016/j.jsg.2008.09.003
- Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, *124*(3), 372–422. doi:10.1037/0033-2909.124.3.372
- Rebeyrol, A., Bablet, M., Bisson-Vaivre, C., Caraglio, M., Carsin, C., Costa-Lascoux, J., ... Negrel, I. (2010). *Discriminations à l'École*.

- Reeves, L. M., Lai, J., Larson, J. A., Oviatt, S., Balaji, T. S., Buisine, S., ... Wang, Q. Y. (2004). Guidelines for multimodal user interface design. *Communications of the ACM*, 47(1), 57–59. doi:10.1145/962081.962106
- Reilent, E. (2012). *Whiteboard Architecture for the Multi-agent Sensor Systems*. Tallinn University of Technology.
- Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., ... Lécuyer, A. (2010). OpenViBE: An Open-Source Software Platform to Design, Test, and Use Brain–Computer Interfaces in Real and Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 19(1), 35–53. doi:10.1162/pres.19.1.35
- Riaño, D. (2007). The SDA* model: A set theory approach. In P. Kokol, V. Podgorelec, D. Mičetić-Turk, M. Zorman, & M. Verlić (Eds.), *Twentieth IEEE International Symposium on Computer-Based Medical Systems (CBMS'07)* (pp. 563–568). Maribor, Slovenia: IEEE. doi:10.1109/CBMS.2007.110
- Richard, N., Dojat, M., & Garbay, C. (2004). Automated segmentation of human brain MR images using a multi-agent approach. *Artificial Intelligence in Medicine*, 30(2), 153–175. doi:10.1016/j.artmed.2003.11.006
- Richlan, F., Kronbichler, M., & Wimmer, H. (2012). Structural abnormalities in the dyslexic brain: A meta-analysis of voxel-based morphometry studies. *Human Brain Mapping*, 000. doi:10.1002/hbm.22127
- Ricordel, P.-M., & Demazeau, Y. (2002). Volcano, a vowels-oriented multi-agent platform. In B. Dunin-Keplicz & E. Nawarecki (Eds.), *From Theory to Practice in Multi-Agent Systems* (pp. 253–262). Cracow, Poland: Springer Berlin Heidelberg. doi:10.1007/3-540-45941-3_27
- Rimrodt, S. L., Peterson, D. J., Denckla, M. B., Kaufmann, W. E., & Cutting, L. E. (2010). White matter microstructural differences linked to left perisylvian language network in children with dyslexia. *Cortex*, 46(6), 739–749. doi:10.1016/j.cortex.2009.07.008
- Ringard, J.-C. (2000). *A propos de l'enfant dysphasique et de l'enfant dyslexique*.
- Ringbeck, T., & Hagebecker, B. (2007). A 3d time of flight camera for object detection. In *8th conference on Optical 3-D Measurement Techniques*.
- Rizzo, A., Buckwalter, J. G., van der Zaag, C., Neumann, U., Thiebaut, M., Chua, C., ... Larson, P. (2000). Virtual Environment Applications in Clinical Neuropsychology. In *IEEE Virtual Reality* (pp. 63–70). New Brunswick, NJ: IEEE. doi:10.1109/VR.2000.840364
- Robertson, D. (2004). A lightweight method for coordination of agent oriented web services. In *AAAI Spring Symposium on Semantic Web Services* (pp. 76–83). Menlo Park, California, USA: AAAI Press.
- Robertson, D. (2005). A lightweight coordination calculus for agent systems. In J. Leite, A. Omicini, P. Torroni, & Pi. Yolum (Eds.), *Declarative agent languages and technologies II* (pp. 183–197). New York, NY, USA: Springer Berlin Heidelberg. doi:10.1007/11493402_11
- Robertson, D. (2012). Lightweight coordination calculus for agent systems: retrospective and prospective. In C. Sakama, S. Sardina, W. Vasconcelos, & M. Winikoff (Eds.), *Declarative*

Agent Languages and Technologies IX (pp. 84–89). Taipei, Taiwan: Springer Berlin Heidelberg. doi:10.1007/978-3-642-29113-5_7

- Robichon, F., & Habib, M. (1998). Abnormal callosal morphology in male adult dyslexics: relationships to handedness and phonological abilities. *Brain and Language*, 62(1), 127–46. doi:10.1006/brln.1997.1891
- Rodríguez, A., Fernández-Medina, E., & Piattini, M. (2006). Towards a UML 2.0 extension for the modeling of security requirements in business processes. In S. Fischer-Hübner, S. Furnell, & C. Lambrinoudakis (Eds.), *Trust and Privacy in Digital Business* (pp. 51–61). Kraków, Poland: Springer Berlin Heidelberg. doi:10.1007/11824633_6
- Rodríguez, M. D., Favela, J., Preciado, A., & Vizcaíno, A. (2005). Agent-based ambient intelligence for healthcare. *Ai Communications*, 18(3), 201–216.
- Rodríguez, S., Gaud, N., & Galland, S. (2014). SARL: a general-purpose agent-oriented programming language. In D. Ślęzak, B. Dunin-Kępicz, M. Lewis, & T. Terano (Eds.), *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)* (pp. 103–110). Warsaw, Poland: IEEE. doi:10.1109/WI-IAT.2014.156
- Rodríguez, S., Pérez-Lancho, B., De Paz, J. F., Bajo, J., & Corchado, J. M. (2009). Ovamah: Multiagent-based adaptive virtual organizations. In *12th International Conference on Information Fusion (FUSION'09)* (pp. 990–997). Seattle, Washington, USA: IEEE.
- Rodríguez-Domínguez, C., Benghazi, K., Noguera, M., Garrido, J. L., Rodríguez, M. L., & Ruiz-López, T. (2012). A Communication model to integrate the Request-Response and the publish-subscribe paradigms into ubiquitous systems. *Sensors*, 12(6), 7648–7668. doi:10.3390/s120607648
- Rosenfeld, A. (1970). Connectivity in digital pictures. *Journal of the ACM*, 17(1), 146–160. doi:10.1145/321556.321570
- Rosenfeld, A. (1974). Adjacency in digital pictures. *Information and Control*, 26(1), 24–33. doi:10.1016/S0019-9958(74)90696-2
- Rosenfeld, A., Kong, T. Y., & Nakamura, A. (1998). Topology-Preserving Deformations of Two-Valued Digital Pictures. *Graphical Models and Image Processing*, 60(1), 24–34. doi:10.1006/gmip.1997.0459
- Roser, S., & Bauer, B. (2005). A categorization of collaborative business process modeling techniques. In A. Wombacher, C. Huemer, & M. Stolze (Eds.), *Seventh IEEE International Conference on E-Commerce Technology Workshops (CEC 2005)* (pp. 43–51). Munich, Germany: IEEE. doi:10.1109/CECW.2005.1
- Rousseau-Giral, A.-C., Strohl, H., Bizot, C., Ravary, Y., & Gossot, B. (2002). *Enquête sur le rôle des dispositifs médico-social sanitaire et pédagogique dans la prise en charge des troubles complexes du langage*.
- Royce, W. W. (1970). Managing the development of large software systems. In *IEEE WESCON* (pp. 328–338). IEEE.
- Rudenko, D., & Borisovs, A. (2007). An Overview of Blackboard Architecture Application for Real Tasks. *Computer Science*, 31, 50–56.

- Sabouret, N. (2009). *Interactions sur le fonctionnement dans les systèmes multi-agents ouverts et hétérogènes*. Université Pierre et Marie CURIE.
- Sachdeva, S., & Bhalla, S. (2012). Semantic interoperability in standardized electronic health record databases. *ACM Journal of Data and Information Quality*, 3(1 (1)), 1–37. doi:10.1145/2166788.2166789
- Sadek, M. D. (1991). Dialogue acts are rational plans. In *Second VENACO Workshop - The Structure of Multimodal Dialogue* (pp. 19–47). Acquafredda di Maratea, Italy: International Speech Communication Association.
- Sainfort, F., Jacko, J., Edwards, P. J., & Booske, B. C. (2007). Human-Computer Interaction in Health Care. In A. Sears & J. A. Jacko (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications* (2nd ed., pp. 661–678). CRC Press. doi:10.1201/9781410615862.ch33
- Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. In *Symposium on Eye tracking research & applications (ETRA'00)* (pp. 71–78). New York, USA: ACM Press. doi:10.1145/355017.355028
- Sampath, H., Sivaswamy, J., & Indurkha, B. (2010a). Assistive systems for children with dyslexia and autism. In *ACM SIGACCESS Accessibility* (pp. 32–36). ACM. doi:10.1145/1731849.1731855
- Sampath, H., Sivaswamy, J., & Indurkha, B. (2010b). Role of technology in assisting children with Developmental Disorders. In *IGAI Workshop on Emerging Research Trends in Artificial Intelligence (ERTAI - 2010)*.
- San Agustin Lopez, J. (2009). *Off-the-Shelf Gaze Interaction*. The IT University of Copenhagen.
- San Agustin Lopez, J., Skovsgaard, H., Mollenbach, E., Barret, M., Tall, M., Hansen, D. W., & Hansen, J. P. (2010). Evaluation of a low-cost open-source gaze tracker. In *The 2010 Symposium on Eye-Tracking Research & Applications (ETRA'10)* (pp. 77–80). Austin, TX, USA: ACM Press. doi:10.1145/1743666.1743685
- Sánchez, D., Isern, D., & Moreno, A. (2010). Agent Technology and Healthcare: Possibilities, Challenges and Examples of Application. In I. Bichindaritz, S. Vaidya, A. Jain, & L. C. Jain (Eds.), *Computational Intelligence in Healthcare 4* (pp. 25–48). Springer Berlin Heidelberg. doi:10.1007/978-3-642-14464-6_3
- Sanchis, A., Julián, V., Corchado, J. M., Billhardt, H., & Carrascosa, C. (2014). Using Natural Interfaces for Human-Agent Immersion. In J. M. Corchado, J. Bajo, J. Kozlak, P. Pawlewski, J. M. Molina, B. Gaudou, ... P. G. Teodoro (Eds.), *PAAMS 2014 International Workshops* (pp. 358–367). Salamanca, Spain: Springer International Publishing. doi:10.1007/978-3-319-07767-3_32
- Sandhu, K. K. (1992). Specification and description language (SDL). In *IEE Tutorial Colloquium on Formal Methods and Notations Applicable to Telecommunications* (pp. 3/1–3/4). Londres, United Kingdom: IET.
- Santos, A., Joly-Pottuz, B., Moreno, S., Habib, M., & Besson, M. (2007). Behavioural and event-related potentials evidence for pitch discrimination deficits in dyslexic children: Improvement after intensive phonic intervention. *Neuropsychologia*, 45(5), 1080–1090. doi:10.1016/j.neuropsychologia.2006.09.010

- Sauer, S., & Engels, G. (1999). MVC-Based Modeling Support for Embedded Real-Time Systems. In P. Hofmann & A. Schrr (Eds.), *OMER Workshop* (pp. 11–14). Herrsching, Germany.
- Sauvage, S. (2004). Agent oriented design patterns: A case study. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '04)* (pp. 1496–1497). IEEE. doi:10.1109/AAMAS.2004.48
- Savarimuthu, B. T. R., & Purvis, M. (2004). A collaborative multi-agent based workflow system. In M. G. Negoita, R. J. Howlett, & L. C. Jain (Eds.), *8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES 2004)* (pp. 1187–1193). Wellington, New Zealand: Springer Berlin Heidelberg. doi:10.1007/978-3-540-30133-2_158
- Savarimuthu, B. T. R., Purvis, M., & Fleurke, M. (2004). Monitoring and controlling of a multi-agent based workflow system. In J. Hogan, P. Montague, M. Purvis, & C. Steketee (Eds.), *Australasian Workshop on Data Mining and Web Intelligence (DMWI2004)* (pp. 127–132). Bedford Park, Australia: Australian Computer Society.
- Sayyad Shirabad, J., Wilk, S., Michalowski, W., & Farion, K. (2012). Implementing an integrative multi-agent clinical decision support system with open source software. *Journal of Medical Systems*, 36(1), 123–137. doi:10.1007/s10916-010-9452-9
- Schadow, G., Russler, D. C., Mead, C. N., & McDonald, C. J. (2000). Integrating medical information and knowledge in the HL7 RIM. In *AMIA Annual Symposium Proceedings* (pp. 764–768). Los Angeles, CA: American Medical Informatics Association.
- Schell, J. (2010). *L'art du game design*. Pearson.
- Schillo, M., Fischer, K., & Klein, C. T. (2000). The micro-macro link in DAI and sociology. In S. Moss & P. Davidsson (Eds.), *Second International Workshop on Multi-Agent-Based Simulation (MABS 2000)* (pp. 133–148). Boston, MA, USA: Springer Berlin Heidelberg. doi:10.1007/3-540-44561-7_10
- Schneider, N. (2011). *A low-cost head-mounted eye-tracking system for automotive applications*. University of Lübeck.
- Schneider, N., Bex, P., Barth, E., & Dorr, M. (2011). An open-source low-cost eye-tracking system for portable real-time and offline tracking. In *The 1st Conference on Novel Gaze-Controlled Applications, NGCA '11*. ACM Press.
- Schulte-Körne, G., & Bruder, J. (2010). Clinical neurophysiology of visual and auditory processing in dyslexia: a review. *Clinical Neurophysiology: Official Journal of the International Federation of Clinical Neurophysiology*, 121(11), 1794–809. doi:10.1016/j.clinph.2010.04.028
- Schulte-Körne, G., Deimel, W., Bartling, J., & Remschmidt, H. (1998). Auditory processing and dyslexia: evidence for a specific speech processing deficit. *NeuroReport*, 9(2), 337–340.
- Schwarte, R., Häusler, G., & Malz, R. W. (2000). Three-Dimensional Imaging Techniques. In B. Jähne & H. Haußecker (Eds.), *Computer Vision and Applications: A Guide for Students and Practitioners* (pp. 177–208). Academic Press. doi:10.1016/B978-012379777-3/50008-X

- Sears, A., & Jacko, J. A. (2007). *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications* (2nd ed.). CRC Press. doi:10.1201/9781410615862
- Seffah, A., & Forbrig, P. (2002). Multiple User Interfaces: Towards a Task-Driven and Patterns-Oriented Design Model. In P. Forbrig, Q. Limbourg, J. Vanderdonckt, & B. Urban (Eds.), *Interactive Systems: Design, Specification, and Verification* (pp. 118–132). Rostock, Germany: Springer Berlin Heidelberg. doi:10.1007/3-540-36235-5_9
- Sethian, J. A. (1999). *Level set methods and fast marching methods (2nd edition)*. (University of California, Ed.). Cambridge: Cambridge University Press.
- Sethian, J. A. (2002). Computational methods for advancing interfaces. In *Conference on Interfaces in the 21st Century*.
- Seymour, P. H. K., Aro, M., & Erskine, J. M. (2003). Foundation literacy acquisition in European orthographies. *British Journal of Psychology*, 94(2), 143–74. doi:10.1348/000712603321661859
- Sezgin, M., & Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1), 146–165.
- Shahar, Y., Miksch, S., & Johnson, P. D. (1996). An intention-based language for representing clinical guidelines. In J. J. Cimino (Ed.), *AMIA Annual Symposium Proceedings* (pp. 592–596). Washington, DC: American Medical Informatics Association.
- Shahar, Y., Miksch, S., & Johnson, P. D. (1998). The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine*, 14(1-2), 29–51. doi:10.1016/S0933-3657(98)00015-3
- Shakshuki, E., Luo, Z., Gong, J., & Chen, Q. (2004). Multi-agent system for security service. In L. Barolli (Ed.), *18th International Conference on Advanced Information Networking and Application (AINA'04)* (Vol. 1, pp. 303–308). Fukuoka, Japan: IEEE. doi:10.1109/AINA.2004.1283928
- Shaul, S. (2008). Event-related potentials (ERPs) in the study of dyslexia. In Z. Breznitz (Ed.), *Brain Research in Language* (pp. 51–92). Springer US. doi:10.1007/978-0-387-74980-8_2
- Sheela, S. V., & Vijaya, P. A. (2011). Mapping Functions in Gaze Tracking. *International Journal of Computer Applications*, 26(3), 36–42. doi:10.5120/3080-4216
- Shehory, O. (1998). *Architectural Properties of Multi-Agent Systems*. Pittsburgh, Pennsylvania.
- Shehory, O. (2001). Software architecture attributes of multi-agent systems. In P. Ciancarini & M. J. Wooldridge (Eds.), *Agent-Oriented Software Engineering* (pp. 77–90). Springer Berlin Heidelberg. doi:10.1007/3-540-44564-1_5
- Shi, Y., & Karl, W. C. (2005a). *A fast implementation of the level set method without solving partial differential equations*. Boston.
- Shi, Y., & Karl, W. C. (2005b). A fast level set method without solving PDEs. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)* (Vol. 2, pp. 97–100). IEEE. doi:10.1109/ICASSP.2005.1415350

- Shi, Y., & Karl, W. C. (2005c). Real-time tracking using level sets. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2, 34–41. doi:10.1109/CVPR.2005.294
- Shi, Y., & Karl, W. C. (2008). A real-time algorithm for the approximation of level-set-based curve evolution. *IEEE Transactions on Image Processing*, 17(5), 645–656. doi:10.1109/TIP.2008.920737
- Shiffman, R. N., Karras, B. T., Agrawal, A., Chen, R., Marengo, L., & Nath, S. (2000). GEM: a proposal for a more comprehensive guideline document model using XML. *Journal of the American Medical Informatics Association*, 7(5), 488–498. doi:10.1136/jamia.2000.0070488
- Shiffman, R. N., Liaw, Y., Brandt, C. A., & Corb, G. J. (1999). Computer-based Guideline Implementation Systems: A Systematic Review of Functionality and Effectiveness. *Journal of the American Medical Informatics Association*, 6(2), 104–114. doi:10.1136/jamia.1999.0060104
- Shindo, K., Kawashima, K., Ushiba, J., Ota, N., Ito, M., Ota, T., ... Liu, M. (2011). Effects of neurofeedback training with an electroencephalogram-based brain-computer interface for hand paralysis in patients with chronic stroke: A preliminary case series study. *Journal of Rehabilitation Medicine*, 43(10), 951–957. doi:10.2340/16501977-0859
- Shneiderman, B., & Plaisant, C. (2010). *Designing the user interface: Strategies for Effective Human-Computer Interaction* (5th ed.). Addison-Wesley.
- Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, 60(1), 51–92. doi:10.1016/0004-3702(93)90034-9
- Shtiliyanova, A. (2012). *Modélisation et simulation informatique de l'innovation en médecine - Conception d'un outil d'aide à l'évaluation médico-économique pour les centres de radiothérapie*. Université d'Auvergne.
- Shtiliyanova, A., Feschet, F., & Pommier, P. (2011). Tool for the Evaluation of Innovative Therapies: Multi-agent based System. In *1st International Conference on Simulation and Modeling Methodologies, Technologies and Applications - SIMULTECH 2011* (pp. 148–154). Noordwijkerhout, Netherlands: SciTePress Digital Library.
- Singh, C., Sachan, S., & Gangwar, M. (2012). MAS based Selection and Composition Process of SWS's for Medical Health Care Planning System. *International Journal of Computer Applications*, 60(16), 40–44. doi:10.5120/9779-4394
- Singh, S., Ismail, B. I., Haron, F., & Yong, C. H. (2005). Architecture of agent-based healthcare intelligent assistant on grid environment. In K.-M. Liew, H. Shen, S. See, W. Cai, P. Fan, & S. Horiguchi (Eds.), *Parallel and Distributed Computing: Applications and Technologies* (pp. 58–61). Springer Berlin Heidelberg. doi:10.1007/978-3-540-30501-9_15
- Siok, W. T., Perfetti, C. A., Jin, Z., & Tan, L. J. (2004). Biological abnormality of impaired reading is constrained by culture. *Nature*, 431, 71–76. doi:10.1038/nature02709.1
- Smart, S., & Purves, I. N. (2001). The Problems of Large-scale Knowledge Authoring and the PRODIGY Solutions. In *AMIA Annual Symposium Proceedings* (p. 835). American Medical Informatics Association.

- Smith, I. A., & Cohen, P. R. (1996). Toward a semantics for an agent communications language speech-acts. In *13th National Conference on Artificial Intelligence* (pp. 24–31). Portland, Oregon: AAAI/MIT Press.
- Snowling, M. J. (2001). From Language to Reading and Dyslexia. *Dyslexia*, 7(1), 37–46. doi:10.1002/dys.185
- Society for Medical Decision Making Committee on Standardization of Clinical Algorithms. (1992). Proposal for Clinical Algorithm Standards. *Medical Decision Making*, 12(2), 149–154. doi:10.1177/0272989X9201200208
- Software AG. (2012). ARIS Express. Software AG. Retrieved from <http://www.ariscommunity.com/aris-express>
- Sørensen, E., & Mihailesc, M. I. (2010). Model-View-ViewModel (MVVM) Design Pattern using Windows Presentation Foundation (WPF) Technology. *MegaByte Journal*, 9(4), 1–19.
- Špakov, O. (2012). Comparison of eye movement filters used in HCI. In S. N. Spencer (Ed.), *Symposium on Eye Tracking Research and Applications (ETRA'12)* (pp. 281–284). Santa Barbara, CA, USA: ACM Press. doi:10.1145/2168556.2168616
- Spanoudakis, N., & Moraitis, P. (2008). The Agent Modeling Language (AMOLA). In D. Dochev, M. Pistore, & P. Traverso (Eds.), *Artificial Intelligence: Methodology, Systems, and Applications* (pp. 32–44). Varna, Bulgaria: Springer Berlin Heidelberg. doi:10.1007/978-3-540-85776-1_4
- Sparrow, L., & Mielle, S. (2002). Activation of Phonological Codes during Reading: Evidence from Errors Detection and Eye Movements. *Brain and Language*, 81(1-3), 509–516. doi:10.1006/brln.2001.2543
- Spragins, W. A., & Lorenzetti, D. L. (2008). *Public Expectation and Patient Experience of Integration of Health Care: A Literature Review*.
- Sprenger-Charolles, L., Colé, P., Béchennec, D., & Kipffer-Piquard, A. (2005). French normative data on reading and related skills from EVALEC, a new computerized battery of tests (end Grade 1, Grade 2, Grade 3, and Grade 4). *Revue Européenne de Psychologie Appliquée/European Review of Applied Psychology*, 55(3), 157–186. doi:10.1016/j.erap.2004.11.002
- Springer. (2014). SpringerLink. Retrieved from <http://link.springer.com/>
- SRI International. (1997). Open Knowledge Base Connectivity. Menlo Park, CA, USA: SRI International. Retrieved from <http://www.ai.sri.com/~okbc/>
- Srivastava, N., & Dev, H. (2013). Computer Vision Architecture using Fusion Technique. *International Journal of Computer Applications*, 76(4), 40–43. doi:10.5120/13238-0674
- Staccini, P., Joubert, M., Quaranta, J.-F., Fieschib, D., & Fieschib, M. (2000). Integration of health care process analysis in the design of a clinical information system: applying to the blood transfusion process. In *AMIA Annual Symposium Proceedings* (pp. 824–828).
- Stampe, D. M. (1993). Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems. *Behavior Research Methods, Instruments, & Computers*, 25(2), 137–142. doi:10.3758/BF03204486

- Stanford Center for Biomedical Informatics Research. (2014). Protégé: A free, open-source ontology editor and framework for building intelligent systems. Retrieved from <http://protege.stanford.edu/>
- Stanford University. (2008). Ontolingua Home Page. Retrieved from <http://www.ksl.stanford.edu/software/ontolingua/>
- Star, S. L., & Ruhleder, K. (1994). Steps towards an ecology of infrastructure: complex problems in design and access for large-scale collaborative systems. In J. B. Smith, F. D. Smith, & T. W. Malone (Eds.), *The 1994 ACM conference on Computer supported cooperative work (CSCW '94)* (pp. 253–264). Chapel Hill, North Carolina, USA: ACM Press. doi:10.1145/192844.193021
- Stein, J. (2001). The magnocellular theory of developmental dyslexia. *Dyslexia*, 7(1), 12–36. doi:10.1002/dys.186
- Stoufflet, P. E., Ohno-Machado, L., Deibel, S. R. A., Lee, D., & Greenes, R. A. (1996). GEODE-CM: a state-transition framework for clinical management. In *AMIA Annual Symposium Proceedings* (Vol. 9, p. 924). American Medical Informatics Association.
- Su, M.-C., Wang, K.-C., & Chen, G.-D. (2006). An Eye Tracking System and Its Application in Aids for People With Severe Disabilities. *Biomedical Engineering: Applications, Basis and Communications*, 18(06), 319–327. doi:10.4015/S1016237206000476
- Sugden, B. (1999). A Temporal Protocol Model for Chronic Disease Management in the PRODIGY Project. In R. W. Milne, A. L. Macintosh, & M. Bramer (Eds.), *Applications and Innovations in Expert Systems VI* (pp. 165–173). Cambridge, United Kingdom: Springer London. doi:10.1007/978-1-4471-0575-6_12
- Sulaiman, R., Sharma, D., Ma, W., & Tran, D. (2009). A Multi-agent Security Architecture. In *2009 Third International Conference on Network and System Security* (pp. 184–191). IEEE. doi:10.1109/NSS.2009.78
- Sullivan, F., & Mitchell, E. (1995). Has general practitioner computing made a difference to patient care? A systematic review of published reports. *British Medical Journal*, 311, 848–852. doi:10.1136/bmj.311.7009.848
- Sutherland, I. E. (2003). *Sketchpad: A man-machine graphical communication system. Seminal graphics*. Cambridge, United Kingdom.
- Sutton, D. R., & Fox, J. (2003). The syntax and semantics of the PROforma guideline modeling language. *Journal of the American Medical Informatics Association*, 10(5), 433–443. doi:10.1197/jamia.M1264
- Świrski, L., Bulling, A., & Dodgson, N. (2012). Robust real-time pupil tracking in highly off-axis images. In *Symposium on Eye Tracking Research and Applications (ETRA'12)* (pp. 173–176). ACM Press. doi:10.1145/2168556.2168585
- Sybase Inc. (2013). PowerAMC. Sybase Inc. Retrieved from <http://www.sybase.com/products/modelingdevelopment/powerdesigner>
- Sycara, K., Giampapa, J. A., Langley, B. K., & Paolucci, M. (2002). The RETSINA MAS, a Case Study. In A. Garcia, C. Lucena, F. Zambonelli, A. Omicini, & J. Castro (Eds.), *Software Engineering for Large-Scale Multi-Agent Systems: Research Issues and Practical Applications* (pp. 232–250). Springer Berlin Heidelberg. doi:10.1007/3-540-35828-5_15

- Sycara, K., Paolucci, M., van Velsen, M., & Giampapa, J. A. (2003). The RETSINA MAS Infrastructure. *Autonomous Agents and Multiagent Systems*, 7(1-2), 29–48. doi:10.1023/A:1024172719965
- Tanaka, H., Black, J. M., Hulme, C., Stanley, L. M., Kesler, S. R., Whitfield-Gabrieli, S., ... Hoeft, F. (2011). The brain basis of the phonological deficit in dyslexia is independent of IQ. *Psychological Science*, 22(11), 1442–51. doi:10.1177/0956797611419521
- Tapia, D. I., de la Prieta, F., Rodríguez González, S., Bajo, J., & Corchado, J. M. (2011). Organizations of Agents in Information Fusion Environments. In L. Antunes & H. S. Pinto (Eds.), *Progress in Artificial Intelligence* (pp. 59–70). Springer Berlin Heidelberg. doi:10.1007/978-3-642-24769-9_5
- Tardieu, H., Rochfeld, A., & Colletti, R. (2000). *La méthode Merise : Principes et outils*. (Editions d'Organisation, Ed.) (1ère ed.). Eyrolles.
- Taveter, K., & Tamm, B. (2001). A new approach to the modelling, design, and implementation of business information systems. In J. Barzdins & A. Caplinskas (Eds.), *Databases and Information Systems* (pp. 317–335). Springer Netherlands. doi:10.1007/978-94-015-9636-7_25
- Telecom Italia SpA. (2014). JAVA Agent Development Framework. Telecom Italia SpA. Retrieved from <http://jade.tilab.com/>
- Temple, E., Deutsch, G. K., Poldrack, R. A., Miller, S. L., Tallal, P., Merzenich, M. M., & Gabrieli, J. D. E. (2003). Neural deficits in children with dyslexia ameliorated by behavioral remediation: evidence from functional MRI. *Proceedings of the National Academy of Sciences of the United States of America*, 100(5), 2860–2865. doi:10.1073/pnas.0030098100
- Teplan, M. (2002). Fundamentals of EEG measurement. *Measurement Science Review*, 2(2), 1–11.
- Terenziani, P., Molino, G., & Torchio, M. (2001). A modular approach for representing and executing clinical guidelines. *Artificial Intelligence in Medicine*, 23(3), 249–276. doi:10.1016/S0933-3657(01)00087-2
- Terenziani, P., Montani, S., Bottrighi, A., Molino, G., & Torchio, M. (2008). Applying Artificial Intelligence to Clinical Guidelines: the GLARE Approach. In A. Ten Teije, S. Miksch, & P. Lucas (Eds.), *Computer-based Medical Guidelines and Protocols: A Primer and Current Trends* (pp. 273–282). IOS Press. doi:10.3233/978-1-58603-873-1-273
- Terenziani, P., Montani, S., Torchio, M., Molino, G., & Anselma, L. (2003). Temporal consistency checking in clinical guidelines acquisition and execution: the GLARE's approach. In *AMIA Annual Symposium Proceedings* (pp. 659–663). American Medical Informatics Association.
- Tessendorf, B., Bulling, A., Roggen, D., Stiefmeier, T., Tröster, G., Feilner, M., & Derleth, P. (2010). Towards multi-modal context recognition for hearing instruments. In *Wearable Computers (ISWC) 2010* (pp. 1–2). Seoul: IEEE. doi:10.1109/ISWC.2010.5665855
- The Cochrane Collaboration. (2015). The Cochrane Collaboration | Trusted evidence. Informed decisions. Better health. Retrieved from <http://www.cochrane.org/>

- The Eclipse Foundation. (2014a). BPEL Designer Project. The Eclipse Foundation. Retrieved from <https://eclipse.org/bpel/>
- The Eclipse Foundation. (2014b). Eclipse - The Eclipse Foundation open source community website. Retrieved from <http://eclipse.org>
- The Internet Engineering Task Force. (2014a). LDAP search on IETF.org. Retrieved from <http://datatracker.ietf.org/doc/search/?name=LDAP&activedrafts=on&rfcs=on>
- The Internet Engineering Task Force. (2014b). SSL on IETF.org. Retrieved from <http://datatracker.ietf.org/doc/search/?name=SSL&rfcs=on&activedrafts=on&sort=>
- The LimeSurvey project team. (2012). Limesurvey. Retrieved from <http://www.limesurvey.org>
- The Pennsylvania State University. (2015). CiteSeer. Retrieved from <http://citeseerx.ist.psu.edu/>
- Together Teamsolutions Co. Ltd. (2013a). Enhydra JaWE - Java Workflow Editor. Retrieved from <http://www.together.at/prod/workflow/twe>
- Together Teamsolutions Co. Ltd. (2013b). Enhydra Shark - Java Open Source XPDL Workflow. Retrieved from <http://www.together.at/prod/workflow/tws>
- Trenčanský, I., & Červenka, R. (2005). Agent Modeling Language (AML): A Comprehensive Approach to Modeling MAS. *Informatica*, 29(4), 391–400.
- Treuil, J.-P., Drogoul, A., & Zucker, J.-D. (2008). *Modélisation et simulation à base d'agents : Exemples commentés, outils informatiques et questions théoriques*. Dunod.
- Trolès, N., Le Jan, G., Le Bouquin-Jeannès, R., Faucon, G., Pichancourt, D., Scalart, P., & Gombert, J.-E. (2007). Elaboration d'un outil de diagnostic pour la dyslexie développementale. In *Congrès de la Société Française de Psychologie* (pp. 1–8).
- Trolès, N., Le Jan, G., Le Bouquin-Jeannès, R., Faucon, G., Scalart, P., Pichancourt, D., & Gombert, J.-E. (2008). Projet READY, une approche multidisciplinaire dans l'élaboration d'un outil d'aide au diagnostic de la dyslexie. In *Colloque bilan neuropsychologique et démarches pédagogiques* (p. 2). Lyon, France.
- Tu, S. W., Campbell, J. R., Glasgow, J., Nyman, M. A., McClure, R., McClay, J., ... Abarbanel, R. M. (2007). The SAGE Guideline Model: Achievements and Overview. *Journal of the American Medical Informatics Association*, 14(5), 589–598. doi:10.1197/jamia.M2399
- Tu, S. W., Eriksson, H., Gennari, J. H., Shahar, Y., & Musen, M. A. (1995). Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: application of PROTEGE-II to protocol-based decision. *Artificial Intelligence in Medicine*, 7(3), 257–289. doi:10.1016/0933-3657(95)00006-R
- Tu, S. W., & Musen, M. A. (1996). The EON model of intervention protocols and guidelines. In *AMIA Annual Symposium Proceedings* (pp. 587–591). American Medical Informatics Association.
- Tu, S. W., & Musen, M. A. (2000). From guideline modeling to guideline execution: defining guideline-based decision-support services. In *AMIA Annual Symposium Proceedings* (pp. 863–867). American Medical Informatics Association.

- Tu, S. W., & Musen, M. A. (2001). Modeling data and knowledge in the EON guideline architecture. In V. L. Patel, R. Rogers, & R. Haux (Eds.), *MEDINFO 2001: Proceedings of the 10th World Congress on Medical Informatics* (pp. 280–284). IOS Press. doi:10.3233/978-1-60750-928-8-280
- Turk, M. (1998). *Moving from GUIs to PUIs. Symposium on Intelligent Information Media*.
- Turk, M. (2014). Multimodal interaction: A review. *Pattern Recognition Letters*, 36, 189–195. doi:10.1016/j.patrec.2013.07.003
- Turk, M., & Kölsch, M. (2003). *Perceptual interfaces*. Santa Barbara.
- U.S. National Library of Medicine. (2013). Fact Sheet: Medical Subject Headings (MeSH®). Retrieved from <http://www.nlm.nih.gov/pubs/factsheets/mesh.html>
- U.S. National Library of Medicine. (2014). RxNorm. Retrieved from <http://www.nlm.nih.gov/research/umls/rxnorm/>
- U.S. National Library of Medicine. (2015). PubMed. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed>
- United Nations. (2014). Introducing UN/EDIFACT - Trade - UNECE. Retrieved from <http://www.unece.org/trade/untdid/welcome.html>
- United States Congress. Assistive Technology Act (Public Law 108 – 364) (2004).
- Unity Technologies. (2010). Unity. Retrieved from <https://unity3d.com/>
- Valdois, S. (2008). Dyslexies développementales : Théorie de l’empan visuo-attentionnel. *A.N.A.E.*, 96-97(20), 213–219.
- Valdois, S. (2010). Evaluation des difficultés d’apprentissage de la lecture. *Revue Française de Linguistique Appliquée*, 15(1), 89–103.
- Van Breemen, A. J. N. (2003). Integrating agents in software applications. In J. G. Carbonell, J. Siekmann, R. Kowalczyk, J. P. Müller, H. Tianfield, & R. Unland (Eds.), *Agent Technologies, Infrastructures, Tools, and Applications for E-Services* (Vol. 4, pp. 343–354). Erfurt, Germany: Springer Berlin Heidelberg. doi:10.1007/3-540-36559-1_25
- Van der Aalst, W. M. P. (2000). Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques. In W. M. P. van der Aalst, J. Desel, & A. Oberweis (Eds.), *Business Process Management* (pp. 161–183). Springer Berlin Heidelberg. doi:10.1007/3-540-45594-9_11
- Van der Aalst, W. M. P., Aldred, L., Dumas, M., & ter Hofstede, A. H. M. (2004). Design and Implementation of the YAWL System. In A. Persson & J. Stirna (Eds.), *Advanced Information Systems Engineering* (pp. 142–159). Riga, Latvia: Springer Berlin Heidelberg. doi:10.1007/978-3-540-25975-6_12
- Van der Aalst, W. M. P., ter Hofstede, A. H. M., & Weske, M. (2003). Business process management: A survey. In W. M. P. van der Aalst & M. Weske (Eds.), *Business Process Management* (pp. 1–12). Springer Berlin Heidelberg. doi:10.1007/3-540-44895-0_1
- Van der Aalst, W. M. P., & van Hee, K. (2000). *Workflow Management: Models, Methods, and Systems*. MIT Press.

- Van Vliet, L. J., & Verwer, B. J. H. (1988). A contour processing method for fast binary neighbourhood operations. *Pattern Recognition Letters*, 7(1), 27–36. doi:10.1016/0167-8655(88)90041-4
- Van Welie, M., & van der Veer, G. C. (2003). Pattern Languages in Interaction Design: Structure and Organization. In *Human-Computer Interaction (Interact'03)* (pp. 527–534). Zürich, Switzerland: IOS Press.
- Vasserman, E. Y., & Hatcliff, J. (2014). Foundational Security Principles for Medical Application Platforms. In *Information Security Applications* (pp. 213–217). Springer International Publishing. doi:10.1007/978-3-319-05149-9_13
- Vernier, F., & Nigay, L. (2000). Interfaces multimodales : composition et caractérisation des modalités de sortie. In *Conférence ErgoIHM'2000* (pp. 203–210). Biarritz, France.
- Vidal, M. (2014). *Exploring Eye Movements for Natural Human-Computer Interaction*. Lancaster University, UK.
- Vidal, M., Bulling, A., & Gellersen, H. (2012). Detection of smooth pursuits using eye movement shape features. In S. N. Spencer (Ed.), *Symposium on Eye Tracking Research and Applications (ETRA'12)* (pp. 177–180). Santa Barbara, CA, USA: ACM Press. doi:10.1145/2168556.2168586
- Vidyasagar, T. R., & Pammer, K. (2010). Dyslexia: a deficit in visuo-spatial attention, not in phonological processing. *Trends in Cognitive Sciences*, 14(2), 57–63. doi:10.1016/j.tics.2009.12.003
- Vieira-Marques, P. M., Cruz-Correia, R. J., Robles, S., Cucurull, J., Navarro, G., & Martí, R. (2006). Secure integration of distributed medical data using mobile agents. *IEEE Intelligent Systems*, 21(6), 47–54. doi:10.1109/MIS.2006.120
- Villanueva, A., Cabeza, R., & Porta, S. (2006). Eye tracking: Pupil orientation geometrical modeling. *Image and Vision Computing*, 24(7), 663–679. doi:10.1016/j.imavis.2005.06.001
- Villanueva, A., Cerrolaza, J. J., & Cabeza, R. (2007). Geometry Issues of a Gaze Tracking System. In C. Stephanidis (Ed.), *Universal Access in Human-Computer Interaction: Ambient Interaction* (pp. 1006–1015). Springer Berlin Heidelberg. doi:10.1007/978-3-540-73281-5_110
- Vion-Dury, J. (2010). Remarques épistémologiques sur l'exploration cérébrale: à propos de l'IRM et l'EEG. *L'Évolution Psychiatrique*, 75(4), 621–632. doi:10.1016/j.evopsy.2010.09.013
- Votruba, P., Miksch, S., & Kosara, R. (2004). Facilitating knowledge maintenance of clinical guidelines and protocols. In M. Fieschi, E. Coiera, & Y.-C. J. Li (Eds.), *MEDINFO 2004: Proceedings of the 11th World Congress on Medical Informatics* (pp. 57–61). IOS Press. doi:10.3233/978-1-60750-949-3-57
- Wagner, G. (2003a). A UML profile for External Agent-Object-Relationship (AOR) Models. In F. Giunchiglia, J. Odell, & G. Weiß (Eds.), *Agent-Oriented Software Engineering III* (pp. 138–149). Bologna, Italy: Springer Berlin Heidelberg. doi:10.1007/3-540-36540-0_11

- Wagner, G. (2003b). The Agent–Object-Relationship metamodel: towards a unified view of state and behavior. *Information Systems*, 28(5), 475–504. doi:10.1016/S0306-4379(02)00027-3
- Wagner, P., Bartl, K., Günthner, W., Schneider, E., Brandt, T., & Ulbrich, H. (2006). A Pivotal Head Mounted Camera System that is Aligned by Three-Dimensional Eye Movements. In K.-J. Rähä & A. T. Duchowski (Eds.), *Symposium on Eye Tracking Research & Applications (ETRA'06)* (pp. 117–124). San Diego, CA, USA: ACM Press. doi:10.1145/1117309.1117354
- Walker, J., Pan, E., Johnston, D., Adler-Milstein, J., Bates, D. W., & Middleton, B. (2005). The value of health care information exchange and interoperability. *Health Affairs, Suppl. Web*, W5–10–W5–18. doi:10.1377/hlthaff.w5.10
- Walston, S. L., Kimberly, J. R., & Burns, L. R. (1996). Owned vertical integration and health care: Promise and performance. *Health Care Management Review*, 21(1), 83–92.
- Wang, D., Peleg, M., Bu, D., Cantor, M., Landesberg, G., Lunenfeld, E., ... Shortliffe, E. H. (2003). GESDOR – a generic execution model for sharing of computer-interpretable clinical practice guidelines. In *AMIA Annual Symposium Proceedings* (pp. 694–698). American Medical Informatics Association.
- Wang, D., Peleg, M., Tu, S. W., Boxwala, A. A., Greenes, R. A., Patel, V. L., & Shortliffe, E. H. (2002). Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: A literature review of guideline representation. *International Journal of Medical Informatics*, 68(1-3), 59–70. doi:10.1016/S1386-5056(02)00065-5
- Wang, D., Peleg, M., Tu, S. W., Boxwala, A. A., Ogunyemi, O., Zeng, Q., ... Shortliffe, E. H. (2004). Design and implementation of the GLIF3 guideline execution engine. *Journal of Biomedical Informatics*, 37(5), 305–318. doi:10.1016/j.jbi.2004.06.002
- Wang, D., Peleg, M., Tu, S. W., Shortliffe, E. H., & Greenes, R. A. (2001). Representation of clinical practice guidelines for computer-based implementations. In V. L. Patel, R. Rogers, & R. Haux (Eds.), *MEDINFO 2001: Proceedings of the 10th World Congress on Medical Informatics* (pp. 285–289). IOS Press. doi:10.3233/978-1-60750-928-8-285
- Wang, D., & Shortliffe, E. H. (2002). GLEE – a model-driven execution system for computer-based implementation of clinical practice guidelines. In *AMIA Annual Symposium Proceedings* (pp. 855–859). American Medical Informatics Association.
- Wang, J.-G., Sung, E., & Venkateswarlu, R. (2005). Estimating the eye gaze from one eye. *Computer Vision and Image Understanding*, 98(1), 83–103. doi:10.1016/j.cviu.2004.07.008
- Wang, J.-W., Li, C.-C., & Wang, F.-J. (2005). Dynamic activities on an agent-based workflow management system. In *The 3rd ACS/IEEE International Conference on Computer Systems and Applications* (pp. 648–651). IEEE. doi:10.1109/AICCSA.2005.1387111
- Wang, S. J., Middleton, B., Prosser, L. A., Bardon, C. G., Spurr, C. D., Carchidi, P. J., ... Bates, D. W. (2003). A cost-benefit analysis of electronic medical records in primary care. *The American Journal of Medicine*, 114(5), 397–403. doi:10.1016/S0002-9343(03)00057-3

- Wangler, B., Åhlfeldt, R.-M., & Perjons, E. (2003). Process oriented information systems architectures in healthcare. *Health Informatics Journal*, 9(4), 253–265. doi:10.1177/1460458203094003
- Wangler, B., & Backlund, A. (2005). Information Systems Engineering: What Is It? In *CAiSE Workshops* (pp. 427–437). Porto, Portugal.
- Wangler, B., & Paheerathan, S. (2000). Horizontal and vertical integration of organizational IT systems. In S. Brinkkemper, E. Lindencrona, & A. Sjølvberg (Eds.), *Information Systems Engineering: State of the Art and Research Themes*. Springer London.
- Weibel, N., Emmenegger, C., Lyons, J., Dixit, R., Hill, L. L., & Hollan, J. D. (2013). Interpreter-Mediated Physician-Patient Communication: Opportunities for Multimodal Healthcare Interfaces. In *7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)* (pp. 113–120). IEEE. doi:10.4108/pervasivehealth.2013.252026
- Weiser, M. (1993). Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7), 75–84. doi:10.1145/159544.159617
- Weiss, A. (2009). Health and biomedical informatics. *netWorker*, 13(4), 18–25. doi:10.1145/1655737.1655743
- Weiss, B., Scheffler, T., Möller, S., & Reithinger, N. (2012). Describing Multimodal Human-Computer Interaction. In *Workshop at NordiCHI: Assessing Multimodal Interaction (aMMi)* (pp. 33–36).
- Westermann, G., Mareschal, D., Johnson, M. H., Sirois, S., Spratling, M. W., & Thomas, M. S. C. (2007). Neuroconstructivism. *Developmental Science*, 10(1), 75–83. doi:10.1111/j.1467-7687.2007.00567.x
- White, S. A. (2004). *Introduction to BPMN*.
- Wikimedia. (2015). Wikimedia Commons, la médiathèque libre des projets Wikimedia. Retrieved from <http://commons.wikimedia.org>
- Wikipedia. (2015). Wikipedia, L'encyclopédie libre. Retrieved from <http://www.wikipedia.org/>
- Wilk, S., Michalowski, W., O'Sullivan, D., Farion, K., & Matwin, S. (2008). Engineering of a clinical decision support framework for the point of care use. In *AMIA Annual Symposium Proceedings* (pp. 814–818). American Medical Informatics Association.
- Wise, B. W., Ring, J., & Olson, R. K. (2000). Individual differences in gains from computer-assisted remedial reading. *Journal of Experimental Child Psychology*, 77(3), 197–235. doi:10.1006/jecp.1999.2559
- Witton, C. (2010). Childhood auditory processing disorder as a developmental disorder: the case for a multi-professional approach to diagnosis and management. *International Journal of Audiology*, 49(2), 83–87. doi:10.3109/14992020903289808
- WONCA International Classification Committee. (2005). *International Classification of Primary Care (ICPC-2-R)* (2nd revise.). Oxford University Press.
- Wooldridge, M. J. (1997). Agent-based software engineering. *IEE Proceedings-Software Engineering*, 144(1), 26–37. doi:10.1049/ip-sen:19971026

- Wooldridge, M. J., & Jennings, N. R. (1995a). Agent theories, architectures, and languages: a survey. In M. J. Wooldridge & N. R. Jennings (Eds.), *Intelligent agents* (pp. 1–39). Springer Berlin Heidelberg. doi:10.1007/3-540-58855-8_1
- Wooldridge, M. J., & Jennings, N. R. (1995b). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2), 115–152. doi:10.1017/S0269888900008122
- Wooldridge, M. J., Jennings, N. R., & Kinny, D. (2000). The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3), 285–312. doi:10.1023/A:1010071910869
- Wootton, R. (2001). Telemedicine. *British Medical Journal*, 323, 557 – 560. doi:0.1136/bmj.323.7312.557
- Workflow Management Coalition. (2012). XML Process Definition Language (XPDL): a standard of the WfMC. Retrieved from <http://www.xpdl.org/>
- Workflow Management Coalition. (2014). Home - Workflow Management Coalition. Retrieved from <http://www.wfmc.org/>
- World Health Organization. (2011). *International Statistical Classification of Diseases and Related Health Problems (ICD-10)*. World Health Organization.
- World Health Organization. (2015). WHO | International Classification of Health Interventions (ICHI). Retrieved from <http://www.who.int/classifications/ichi>
- World Wide Web Consortium. (2001). Web Services Description Language (WSDL). Retrieved from <http://www.w3.org/TR/wsdl>
- World Wide Web Consortium. (2007). Simple Object Access Protocol (SOAP) Specifications. Retrieved from <http://www.w3.org/TR/soap/>
- World Wide Web Consortium. (2013). Extensible Markup Language (XML). Retrieved from <http://www.w3.org/XML/>
- World Wide Web Consortium. (2014). Resource Description Framework (RDF). Retrieved from <http://www.w3.org/RDF/>
- Wright, A., Goldberg, H., Hongsermeier, T., & Middleton, B. (2007). A Description and Functional Taxonomy of Rule-based Decision Support Content at a Large Integrated Delivery Network. *Journal of the American Medical Informatics Association*, 14(4), 489–496. doi:10.1197/jamia.M2364
- Wright, A., & Sittig, D. F. (2008). SANDS: A service-oriented architecture for clinical decision support in a National Health Information Network. *Journal of Biomedical Informatics*, 41(6), 962–981. doi:10.1016/j.jbi.2008.03.001
- Wright, A., Sittig, D. F., Ash, J. S., Sharma, S., Pang, J. E., & Middleton, B. (2009). Clinical decision support capabilities of commercially-available clinical information systems. *Journal of the American Medical Informatics Association*, 16(5), 637–644. doi:10.1197/jamia.M3111
- Wu, J. (2008). Robust real-time ellipse detection by Direct Least-Square-Fitting. In *International Conference on Computer Science and Software Engineering (CSSE 2008)* (pp. 923–927). Wuhan, Hubei: IEEE. doi:10.1109/CSSE.2008.789

- Wynn, T. J., & Stewart, S. A. (2005). Comparative testing of ellipse-fitting algorithms: implications for analysis of strain and curvature. *Journal of Structural Geology*, 27(11), 1973–1985. doi:10.1016/j.jsg.2005.06.010
- Xiao, M., Wong, M., Umali, M., & Pomplun, M. (2007). Using eye-tracking to study audio – visual perceptual integration. *Perception*, 36(9), 1391–1395. doi:10.1068/p5731
- Xu, C., Yezzi, A., & Prince, J. L. (2001). A summary of geometric level-set analogues for a general class of parametric active contour and surface models. In *IEEE Workshop on Variational and Level Set Methods in Computer Vision* (pp. 104–111). IEEE Computer Soc. doi:10.1109/VLSM.2001.938888
- Xu, Y., Sauquet, D., Degoulet, P., & Jaulent, M.-C. (2003). Component-based mediation services for the integration of medical applications. *Artificial Intelligence in Medicine*, 27(3), 283–304. doi:10.1016/S0933-3657(03)00007-1
- Yan, J., Pidgeon, P., Krishna, A., & Yong, J. (2007). An Agent-based Decentralised Process Management Framework for Web service Composition. In J. Li, W.-C. Lee, & F. Silvestri (Eds.), *2nd International Conference on Scalable Information Systems (Infoscale 2007)* (pp. 1–4). Suzhou, China: Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering. doi:10.4108/infoscale.2007.904
- Yanhong, W., Fang, H., Caowan, Y., & Yantao, T. (2002). A multi-agent and holonic approach for intelligent scheduling in agile manufacturing environments. In *4th World Congress on Intelligent Control and Automation* (pp. 3071–3075). Shanghai, China: IEEE. doi:10.1109/WCICA.2002.1020094
- Yellowlees, P. M. (2005). Successfully developing a telemedicine system. *Journal of Telemedicine and Telecare*, 11(7), 331–335. doi:10.1258/135763305774472024
- Yilmaz, A., Li, X., & Shah, M. (2004). Object contour tracking using level sets. In *Asian Conference on Computer Vision, ACCV 2004* (pp. 1–7). doi:10.1.1.3.5486
- Yip, R. K. K., Tam, P. K. S., & Leung, D. N. K. (1995). Modification of Hough transform for object recognition using a 2-dimensional array. *Pattern Recognition*, 28(11), 1733–1744. doi:10.1016/0031-3203(95)00031-T
- Yoo, S. K., Kim, D. K., Kim, J. C., Park, Y. J., & Chang, B. C. (2008). Implementation of a large-scale hospital information infrastructure for multi-unit health-care services. *Journal of Telemedicine and Telecare*, 14(3), 164–166. doi:10.1258/jtt.2008.003022
- Young, I. T., Peverini, R. L., Verbeek, P. W., & van Otterloo, P. J. (1981). A new implementation for the binary and Minkowski operators. *Computer Graphics and Image Processing*, 17(3), 189–210. doi:10.1016/0146-664X(81)90001-0
- Yu, E. S. (1995). *Modeling Strategic Relationships for Process Reengineering*. University of Toronto, Canada.
- Yu, L. H., & Eizenman, M. (2004). A new methodology for determining point-of-gaze in head-mounted eye tracking systems. *IEEE Transactions on Biomedical Engineering*, 51(10), 1765–1773. doi:10.1109/TBME.2004.831523
- Yu, L., & Schmid, B. F. (1999). A conceptual framework for agent oriented and role based workflow modelling. In G. Wagner & E. Yu (Eds.), *1st International Workshop on Agent-Oriented Information Systems (AOIS'99)* (pp. 1–19). Heidelberg, Germany: MIT Press.

- Zambonelli, F., Jennings, N. R., & Wooldridge, M. J. (2003). Developing multiagent systems: The Gaia methodology. *ACM Transactions on Software Engineering and Methodology*, 12(3), 317–370. doi:10.1145/958961.958963
- Zeman, M. (2013). *Application for Feature Extraction from Eye Movement Signal Analysis*. Czech Technical University in Prague, Faculty of Electrical Engineering.
- Zeng, L., Benatallah, B., Nguyen, P., & Ngu, A. H. H. (2001). Agflow: Agent-based cross-enterprise workflow management system. In P. M. G. Apers, P. Atzeni, S. Ceri, S. Paraboschi, K. Ramamohanarao, & R. T. Snodgrass (Eds.), *27th International Conference on Very Large Data Bases (VLDB'01)* (pp. 697–698). Roma, Italy: Morgan Kaufmann Publishers Inc.
- Zhang, K., Zhang, L., Song, H., & Zhou, W. (2010). Active contours with selective local or global segmentation: A new formulation and level set method. *Image and Vision Computing*, 28(4), 668–676. doi:10.1016/j.imavis.2009.10.009
- Zhang, R. (2007). *Design of Head mounted displays: Tutorial report for Optics 521*. Tucson, AZ, USA.
- Zhang, S. (2006). A fast level set implementation method for image segmentation and object tracking. In A. G. Tescher (Ed.), *Applications of Digital Image Processing XXIX (SPIE 6312)* (p. (63121R) 1–8). San Diego, California, USA: SPIE. doi:10.1117/12.682228
- Zhang, T., & Gomes, H. M. (2014). Technology survey on video face tracking. In Q. Lin, J. P. Allebach, & Z. Fan (Eds.), *Imaging and Multimedia Analytics in a Web and Mobile World 2014 (SPIE 9027)* (p. (90270F) 1–12). San Francisco, California, USA: SPIE. doi:10.1117/12.2048518
- Zhang, Z. (2004). Camera Calibration. In G. Medioni & S. B. Kang (Eds.), *Emerging Topics in Computer Vision* (pp. 4–43). Prentice Hall Professional Technical Reference.
- Zhu, S. C., & Yuille, A. (1996). Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9), 884–900. doi:10.1109/34.537343
- Zhu, Z., & Ji, Q. (2005). Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. *Computer Vision and Image Understanding*, 98(1), 124–154. doi:10.1016/j.cviu.2004.07.012
- Zhu, Z., & Wang, J.-Y. (2006). Multi-agent based approach to support HCI. In *Machine Learning and Cybernetics* (pp. 188–191). IEEE. doi:10.1109/ICMLC.2006.258951
- Ziegler, J. C. (2011). Understanding Developmental Dyslexia Through Computational Modeling. In P. D. McCardle (Ed.), *Dyslexia across languages: orthography and the brain-gene-behavior link* (pp. 169–183). Paul H. Brookes Publishing Company.
- Ziegler, J. C., Castel, C., Pech-Georgel, C., George, F., Alario, F.-X., & Perry, C. (2008). Developmental dyslexia and the dual route model of reading: simulating individual differences and subtypes. *Cognition*, 107(1), 151–178. doi:10.1016/j.cognition.2007.09.004
- Ziegler, J. C., & Goswami, U. (2006). Becoming literate in different languages: similar problems, different solutions. *Developmental Science*, 9(5), 429–36. doi:10.1111/j.1467-7687.2006.00509.x

- Ziegler, J. C., Perry, C., & Coltheart, M. (2003). Speed of lexical and nonlexical processing in French: The case of the regularity effect. *Psychonomic Bulletin & Review*, *10*(4), 947–953. doi:10.3758/BF03196556
- Ziegler, J. C., Perry, C., Ma-Wyatt, A., Ladner, D., & Schulte-Körne, G. (2003). Developmental dyslexia in different languages: Language-specific or universal? *Journal of Experimental Child Psychology*, *86*(3), 169–193. doi:10.1016/S0022-0965(03)00139-5
- Zielstorff, R. D., Teich, J. M., Paterno, M. D., Segal, M., Kuperman, G. J., Hiltz, F. L., & Fox, R. L. (1998). P-CAPE: A High-Level Tool for Entering and Processing Clinical Practice Guidelines. In *AMIA Annual Symposium Proceedings* (pp. 478–482). American Medical Informatics Association.
- Zorzi, M. (2010). The connectionist dual process (CDP) approach to modelling reading aloud. *European Journal of Cognitive Psychology*, *22*(5), 37–41. doi:10.1080/09541440903435621
- Zorzi, M., Barbiero, C., Facoetti, A., Lonciari, I., Carrozzi, M., Montico, M., ... Ziegler, J. C. (2012). Extra-large letter spacing improves reading in dyslexia. *Proceedings of the National Academy of Sciences of the United States of America*, *109*(28), 11455–11459. doi:10.1073/pnas.1205566109
- Zunic, J., & Sladoje, N. (2000). Efficiency of characterizing ellipses and ellipsoids by discrete moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(4), 407–414. doi:10.1109/34.845384

Annexes

Ces annexes contiennent différents matériels utilisés ou produits durant ces travaux de thèse.

I) Les ressources utilisées pour cette recherche

La mise en œuvre d'un projet informatique requiert des moyens et des ressources (Guérin, 2012). C'est pourquoi, nous avons listé ici tous les outils que nous avons utilisé au long de ce projet de thèse. Listons d'abord les moyens techniques dont nous avons disposé :

- D'un poste de développement équipé d'un processeur Intel® Core™2 Duo (2.8 GHz + 4Go RAM) et des périphériques d'acquisition de données suivants :
 - o De différentes webcams bon marché (maximum 50€ l'unité).
 - o D'un casque EEG Emotiv EPOC® (Emotiv Inc., 2015).
- Des serveurs de documentation constitués :
 - o D'un serveur dédié, et en ligne, hébergeant l'instance d'un CMS (DotNetNuke Corporation, 2012) et disposant entre autres d'un forum et d'un Wiki.
 - o D'un serveur dédié, et en ligne, hébergeant une instance de LimeSurvey (The LimeSurvey project team, 2012).
 - o Des services de Cloud de Mendeley (Mendeley Ltd., 2014), de Dropbox (Dropbox Inc., 2014) et de Google Drive (Google Inc., 2015b).

Sur le poste de développement ont été utilisés notablement les logiciels suivants :

- Paint .NET (Brewster & dotPDN LLC, 2011) et Pencil (Evolus, 2013).
- Visual studio 2008 SP1 (Microsoft, 2008b), avec les bibliothèques suivantes :
 - o OpenCV 2.4.6 (Intel & Willow Garage, 2013).
 - o Emotiv SDK 1.0.0.5 (Emotiv Inc., 2013).
- Unity 3.0 (Unity Technologies, 2010).
- Microsoft Office Visio 2007 (Microsoft, 2007c), Word 2007 (Microsoft, 2007d), PowerPoint 2007 (Microsoft, 2007b) et Excel 2007 (Microsoft, 2007a).
- Prezi (Prezi Inc., 2014).

A des fins bibliographiques et rédactionnelles, nous avons eu recours, de manière significative, aux ressources en ligne listées ci-dessous :

- Wikipédia (Wikipedia, 2015) et Wikimedia Commons (Wikimedia, 2015).
- Le dictionnaire de Français Larousse (Larousse, 2015).
- Encyclopædia Universalis (Encyclopædia Universalis, 2015).
- Les moteurs de recherche scientifiques suivants :
 - o ACM Digital Library (ACM Inc., 2015).
 - o SpringerLink (Springer, 2014).
 - o ScienceDirect (Elsevier B.V., 2015).
 - o IEEE Xplore Digital Library (IEEE Computer Society, 2015).
 - o MEDLINE/PubMed (U.S. National Library of Medicine, 2015).
 - o EBSCOhost Online Research Databases (EBSCO Industries Inc., 2015).
 - o CiteSeer^X (The Pennsylvania State University, 2015).
 - o Cochrane (The Cochrane Collaboration, 2015).
 - o IOS Press (IOS Press, 2015).
 - o Google Scholar (Google Inc., 2015c).
- Les moteurs de recherche généralistes Google (Google Inc., 2015a) et Bing (Microsoft, 2015).

II) Enquête auprès des parents d'enfants dyslexiques

Face à la complexité de fonctionnement du système de santé dans le cadre de la prise en charge d'enfants dyslexiques, il nous a paru pertinent de nous intéresser au ressenti des parents de ces enfants. Nous voulions ainsi en savoir plus sur la compréhension qu'ils avaient de ce système et mieux appréhender leur rapport à la prise en charge de leur enfant. De plus nous étions intéressés par leur connaissance des outils informatiques entourant cette prise en charge.

A) Le questionnaire

Ci-après se trouvent les 18 questions, telles qu'elles étaient présentées sur la version papier de 8 pages du questionnaire, utilisées pour réaliser notre enquête. Une version en ligne, implémentée sur le logiciel Limesurvey (The LimeSurvey project team, 2012), était aussi mise à disposition. C'est d'ailleurs via cette dernière version du questionnaire que nous avons collecté l'intégralité des réponses obtenues.

L'enquête a été ouverte aux réponses de juillet 2013 à fin octobre de la même année. Un appel à participation avait été transmis à des dizaines d'associations afin de pouvoir atteindre la cible de notre enquête, les parents d'enfants souffrant de dyslexie. Ceci nous a d'ailleurs permis de recueillir des réponses exploitables pour 81 personnes.

Concernant la version en ligne, l'ensemble des questions d'une même section y étaient simultanément présentés de telle sorte qu'il fallait avoir répondu à toutes les questions d'une section avant de pouvoir passer à la section suivante. Ainsi les personnes n'ayant pas souhaité remplir l'ensemble du questionnaire nous ont quand même fourni des réponses utilisables pour certaines sections dans leur intégralité.



ENQUETE :

La prise en charge de la dyslexie vue par les parents

Ce questionnaire a pour but de nous aider à appréhender au mieux votre rapport à la prise en charge de votre enfant dyslexique. Nous souhaiterions que vous répondiez de la manière la plus spontanée possible, et nous vous remercions par avance pour le temps que vous allez y consacrer.

A la fin de chaque question vous trouverez, entre parenthèses, une note précisant le type de réponse attendue.

Les recherches, dans le cadre desquelles se tient cette enquête, sont financées par :

- *La Région Auvergne*
- *Le Fond Européen de Développement Régional en Auvergne (Feder)*
- *L'Association Nationale de la Recherche et de la Technologie (ANRT)*



Remarque sur la protection de la vie privée :

- Ce questionnaire est anonyme.
- L'enregistrement de vos réponses ne contient aucune information d'identification sur vous, à moins qu'une question ne vous ait été posée dans ce sens. Si vous avez répondu à un questionnaire utilisant des invitations, vous pouvez être assuré(e) que le code de l'invitation n'est pas enregistré avec votre réponse. Les invitations sont gérées dans une base de données séparée qui n'est mise à jour que pour indiquer si vous avez ou non utilisé votre code pour remplir le questionnaire. Il n'y a aucun moyen de faire le lien entre les invitations et les réponses enregistrées pour ce questionnaire.

Pour nous retourner le questionnaire rempli, vous pouvez (au choix) nous l'expédier :

- Par mail, à l'adresse électronique suivante : recherche@medialexie.com
- Par fax, au numéro suivant : +33 (0)4 73 25 82 50
- Par courrier, à l'adresse postale suivante :

Logolexie
6, rue Nicolas Joseph CUGNOT
63100 CLERMONT-FERRAND
FRANCE

I- Informations générales

1- Dans quel pays résidez-vous ? (*Préciser si besoin*)

France

Belgique

Suisse

Canada

Autre pays :

2- Naturellement les réponses à ce questionnaire sont anonymes, cependant si vous le souhaitez nous pouvons prendre contact avec vous :

non merci

oui, contactez-moi

2- (bis) Si vous souhaitez être contacté, merci de préciser les informations suivantes :
(*Remplir seulement les champs voulus*)

Nom :

Prénom :

Adresse:

Ville :

Code postal :

Téléphone :

Adresse électronique :

II- Parcours de soins de l'enfant et ressenti

3- Depuis combien de temps votre enfant fait-il l'objet d'une prise en charge pour ses troubles dyslexiques ? (*Détailler si besoin*)

- mon enfant n'est pas encore pris en charge
- < 1 an
- 1 à 2 ans
- > 2 ans

Texte libre :

4- Comment le diagnostique a-t-il été posé ? Par qui ? (*Détailler le plus possible*)

Texte libre :

5- Face aux difficultés de votre enfant, quelles sont les solutions que les spécialistes vous ont préconisées ? (*Détailler le plus possible*)

Texte libre :

6- Comment évaluez-vous, en tant que parents d'enfant dyslexique, l'efficacité de ces solutions ? (*Détailler le plus possible*)

Texte libre :

7- Quelles sont vos attentes vis-à-vis de ces solutions ? *(Détaillez le plus possible)*

Texte libre :



8- Quelles améliorations souhaitez-vous dans la prise en charge de la dyslexie par les professionnels de santé ? *(Détaillez le plus possible)*

Texte libre :



III- La dyslexie et les outils informatiques

9- Savez vous que des logiciels sont utilisés à des fins de compensation pour pallier à la dyslexie ? *(Détaillez si besoin)*

- oui
- non

Texte libre :

10- Dans quel cas pensez-vous qu'un outil informatique peut s'avérer utile à un enfant dyslexique ? *(Détaillez le plus possible)*

Texte libre :

11- Votre enfant travaille t-il à l'aide de tels outils ? Si oui, lesquels et pourquoi ? *(Enumérez le plus possible)*

- oui
- non
- je ne sais pas

Texte libre :

11- (bis) A-t-il bénéficié d'une formation concernant ce ou ces logiciels ? Vous a-t-elle parue nécessaire ? efficace ? *(Expliquer votre choix)*

- oui
- non
- je ne sais pas

Texte libre :

12- Pensez-vous que ce type d'aide logicielle est efficace ? *(Expliquer votre choix)*

- oui
- non
- je ne sais pas

Texte libre :

13- Seriez-vous prêts à conseiller à d'autres familles, dans une situation similaire à la votre, l'utilisation d'un logiciel d'aide ? *(Expliquer votre choix)*

- oui
- non
- je ne sais pas

Texte libre :

IV- Les possibilités des nouvelles technologies

14- Seriez-vous favorable à l'idée d'utiliser un ordinateur dans une perspective de rééducation ? *(Expliquer votre choix)*

- oui
- non
- sans avis

Texte libre :

15- Trouveriez-vous qu'il y a une différence entre le travail effectué chez un professionnel de santé selon qu'il est équipé, ou non, d'un ordinateur doté d'un logiciel ? *(Expliquer votre choix)*

- oui
- non
- sans avis

Texte libre :

16- A vos yeux quels avantages ou inconvénients peut avoir l'utilisation d'un ordinateur pour le diagnostique et la rééducation de la dyslexie ? *(Enumérer ces différents aspects)*

Avantages :

Inconvénients :

17- Trouveriez-vous judicieux la possibilité pour un praticien de pouvoir intervenir à distance avec un ordinateur, via internet par exemple ? (*Expliquer votre choix*)

- oui
- non
- sans avis

Texte libre :

18- Auriez-vous des idées de logiciels ou fonctionnalités de logiciels qui permettraient à votre enfant de vivre avec sa dyslexie ? (*Détailler le plus possible*)

Texte libre :

Merci pour votre participation !

Les résultats finaux de cette enquête seront publiés sur notre site :

<http://recherche.medialexie.fr>

B) L'analyse des résultats

Cette partie des annexes correspond à l'analyse qui a été réalisée par des étudiants (Florian BARDET, Julie FAFOURNOUX, Cédric FAURON et Emilie ROBERT) de notre questionnaire. Ces derniers appartenaient tous les quatre au DUT GEA de l'Université d'Auvergne et ont effectué ce travail sous notre supervision durant leur année scolaire 2013-2014.

Dans notre questionnaire deux types de questions avaient été proposés :

- Les questions dites à choix qualitatifs, ne présentant aucune réponse suggérée (c'est-à-dire à texte libre) dans le but de ne pas influencer les personnes interrogées.
- Les questions, présentant plusieurs réponses à choix exclusifs, qui ne nécessitent qu'une seule réponse de la part des personnes interrogées.

Dans le cas des questions dites à choix quantitatifs, les étudiants ont dû définir des métriques objectives afin de pouvoir effectuer un recensement puis une analyse numérique des réponses données librement par les personnes interrogées. Ainsi leur travail a consisté en la création de métriques objectives, le recensement des réponses, la réalisation de représentations graphiques ainsi que la rédaction de commentaires plus personnels.

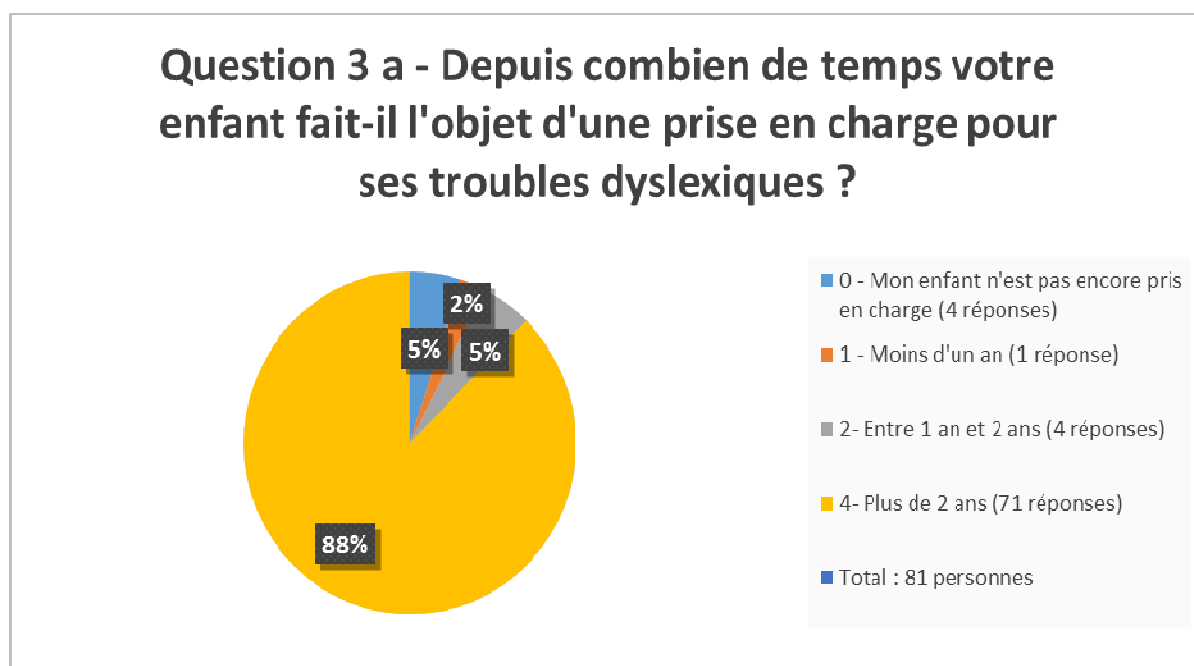
Ci-après se trouve rapportée à l'identique la partie de leur rapport de projet contenant le travail que nous leur avons confié.

Les dénominations chiffrées correspondent aux caractères quantitatifs et les dénominations alphabétiques correspondent aux caractères qualitatifs.

II. Parcours de soin de l'enfant et ressenti (81 personnes)

Question 3 : Depuis combien de temps votre enfant fait-il l'objet d'une prise en charge pour ses troubles dyslexiques ?

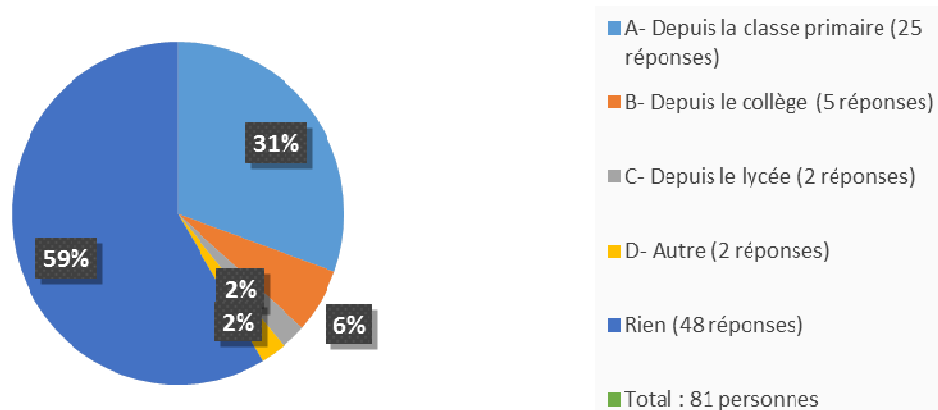
Réponse à la question	Dénomination	Nombre de réponses
Mon enfant n'est pas encore pris en charge	0	4
Moins d'un an	1	2
Entre 1 an et 2 ans	2	4
Plus de 2 ans	3	71



Depuis la classe primaire	A	25
Depuis le collège	B	5
Depuis le lycée	C	2
Autre	D	2
Rien		48

Commentaire : Sur une question normalement exclusive, comptabilisation de deux réponses pour l'identifiant 74 suite à la réponse donnée : « sur trois enfants, nous en avons deux qui sont pris en charge : l'aîné, 16 ans qui est en 1ere STI2D et le second, 12 ans qui est en 5eme »

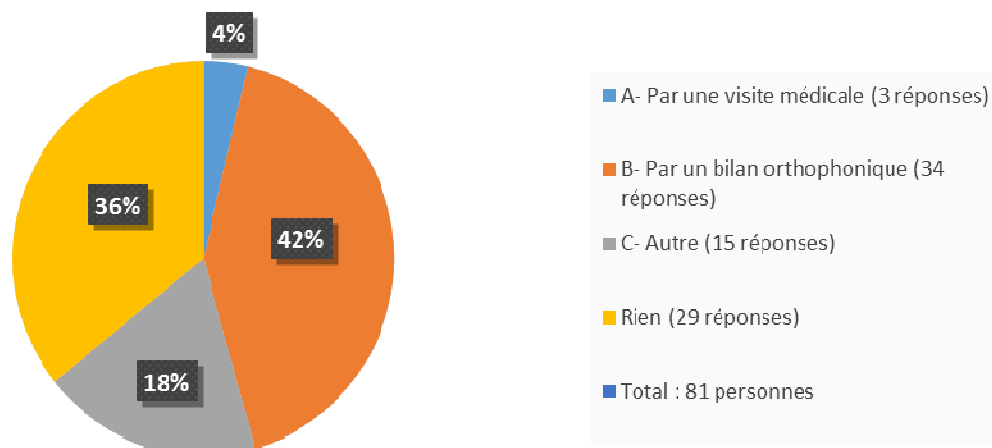
Question 3 b - Depuis combien de temps votre enfant fait-il l'objet d'une prise en charge pour ses troubles dyslexiques ?



Question 4 : Comment le diagnostic a-t-il été posé ? Par qui ?

Par une visite médicale (à l'école)	A	3
Par un bilan orthophonique (lors d'un rdv chez l'orthophoniste)	B	34
Autre	C	15
Rien		29

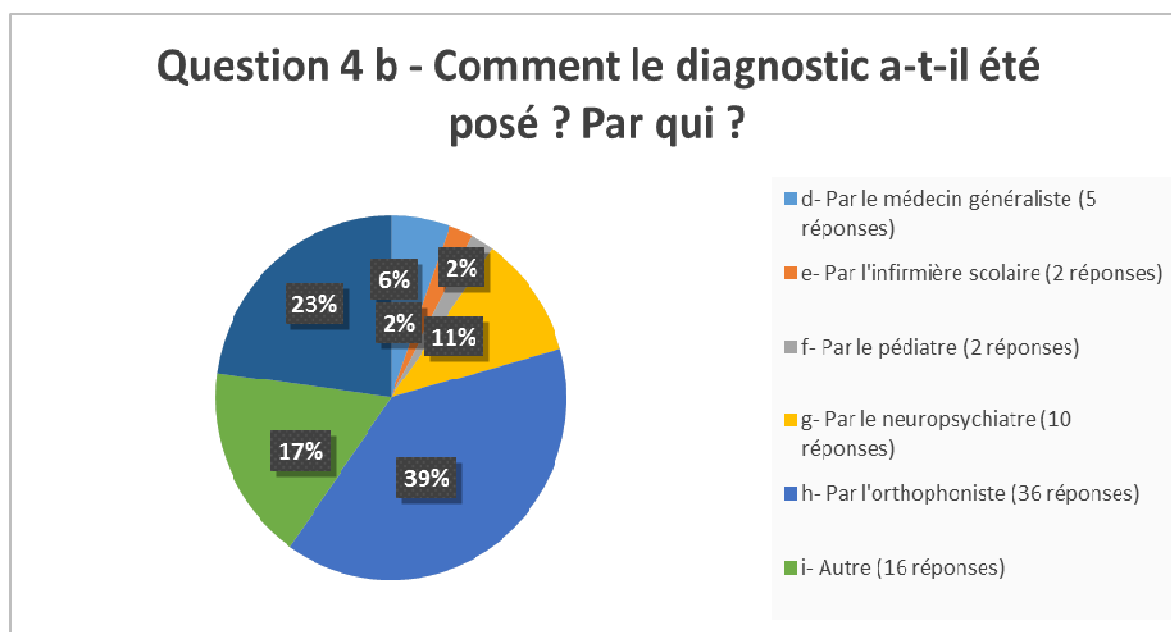
Question 4 a - Comment le diagnostic a-t-il été posé ? Par qui ?



Par le médecin généraliste	d	5
Par l'infirmière scolaire (à l'école)	e	2
Par le pédiatre	f	2
Par le neuropsychiatre	g	10
Par l'orthophoniste	h	36
Autre	i	16
Rien		21

Commentaire : quatre personnes ont répondu à la question 3 mais pas à l'ensemble des questions suivantes de la section 1, ce qui est normalement impossible. Ces personnes sont les identifiants 21, 28, 38 et 107

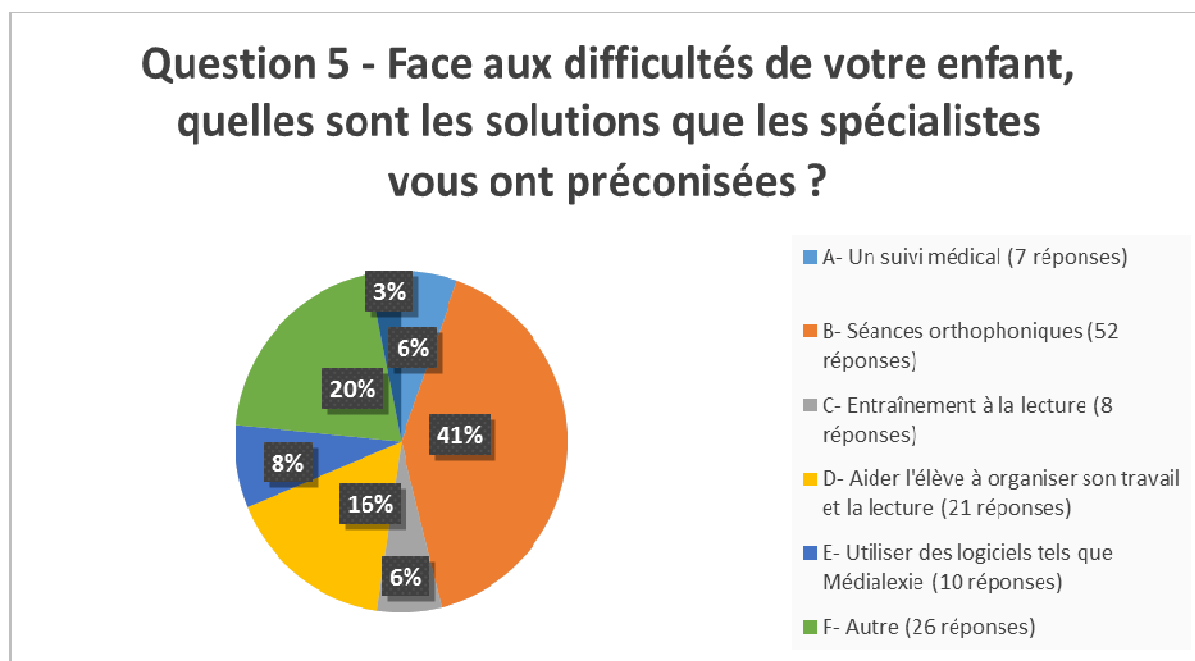
Exemples de réponses types qui ne correspondent pas à la question : « dysorthographique et dysorthographique » ou « En regardant une émission par ma grand-mère dans les années 1980 je crois par le professeur Debrez Ritzen ». Il y a donc une grande difficulté à introduire ces réponses dans les métriques définies, sans avoir à les comptabiliser dans la catégorie « Rien » dans la mesure où ils ont donné une réponse.



Question 5 : Face aux difficultés de votre enfant, quelles sont les solutions que les spécialistes vous ont préconisées ?

Un suivi médical (suivre le patient régulièrement)	A	7
Séances orthophoniques	B	52
Entraînement à la lecture	C	8
Aider l'élève à organiser son travail et la lecture (dispositif scolaire)	D	21
Utiliser des logiciels tels que Médialexie ...	E	10
Autre	F	26
Rien		5

Commentaire : Une grande diversité de réponses qui a compliqué la recherche de variables récurrentes.

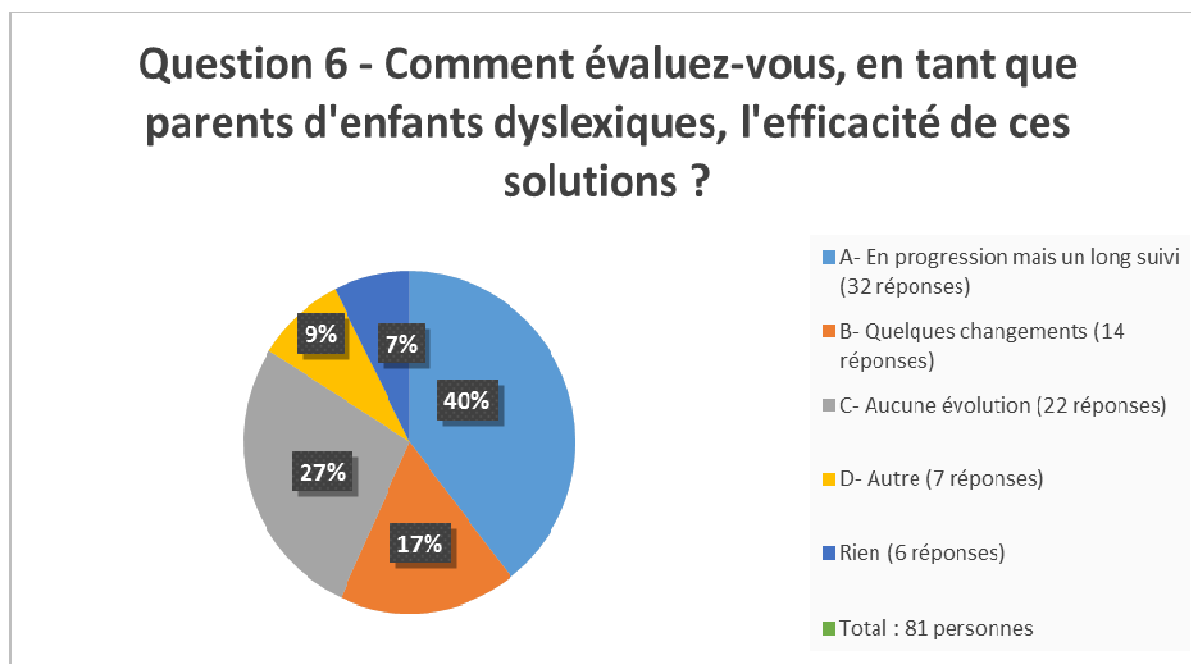


Question 6 : Comment évaluez-vous, en tant que parents d'enfant dyslexique, l'efficacité de ces solutions ?

En progression malgré un long suivi (solutions efficaces mais qui prennent du temps à se mettre en place)	A	32
Quelques changements	B	14
Aucune évolution	C	22
Autre	D	7
Rien		6

Commentaire 1 : Une nouvelle personne n'a répondu que jusqu'à la question 5. Il est donc comptabilisé dans les « rien » des questions 6, 7 et 8. Il s'agit de l'identifiant 29.

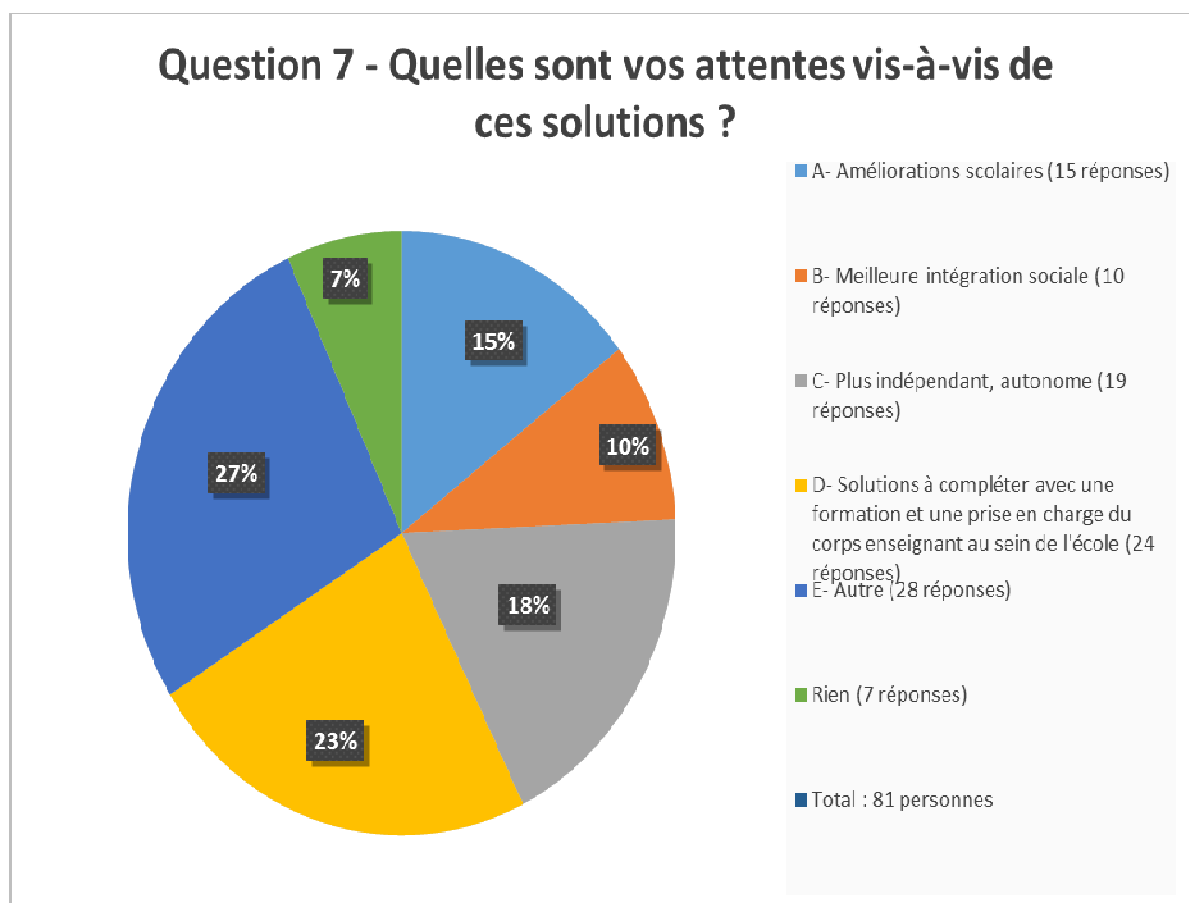
Commentaire 2 : Nous avons décidé de fusionner 2 réponses. En effet les 2 anciennes réponses qui étaient « En progression » et « Un long suivi » ne forment plus qu'une seule réponse puisque nous avons constaté que lorsque les parents répondaient « En progression » la majorité d'entre eux précisait que cette progression s'accompagnait avec un long suivi.



Question 7 : Quelles sont vos attentes vis-à-vis de ces solutions ?

Améliorations scolaires (résultats en hausse)	A	15
Meilleure intégration sociale (adaptation avec les autres enfants)	B	10
Plus indépendant, autonome	C	19
Solutions à compléter avec une formation et une prise en charge du corps enseignant au sein de l'école	D	24
Autre	E	28
Rien		7

Commentaire : L'identifiant 45 n'a exclusivement pas répondu à cette question. S'agit-il d'un oubli ?



Question 8 : Quelles améliorations souhaitez-vous dans la prise en charge de la dyslexie par les professionnels de santé ?

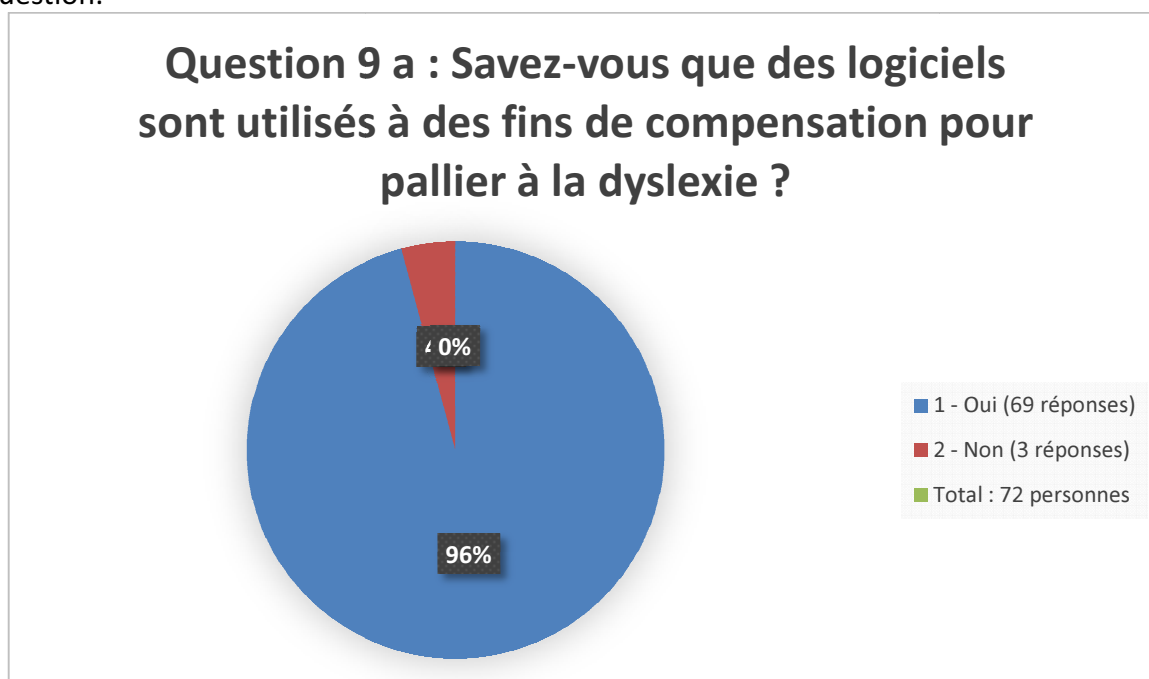
Davantage d'informations (manque de connaissances)	A	13
Réactivité dans la prise en charge (souvent long à se mettre en place)	B	16
Davantage de spécialistes	C	22
Réformes de l'Education nationale (aides financières, matérielles, reconnaissance du trouble)	D	6
Autre	E	25
Rien		6

III- La dyslexie et les outils informatiques (72 personnes)

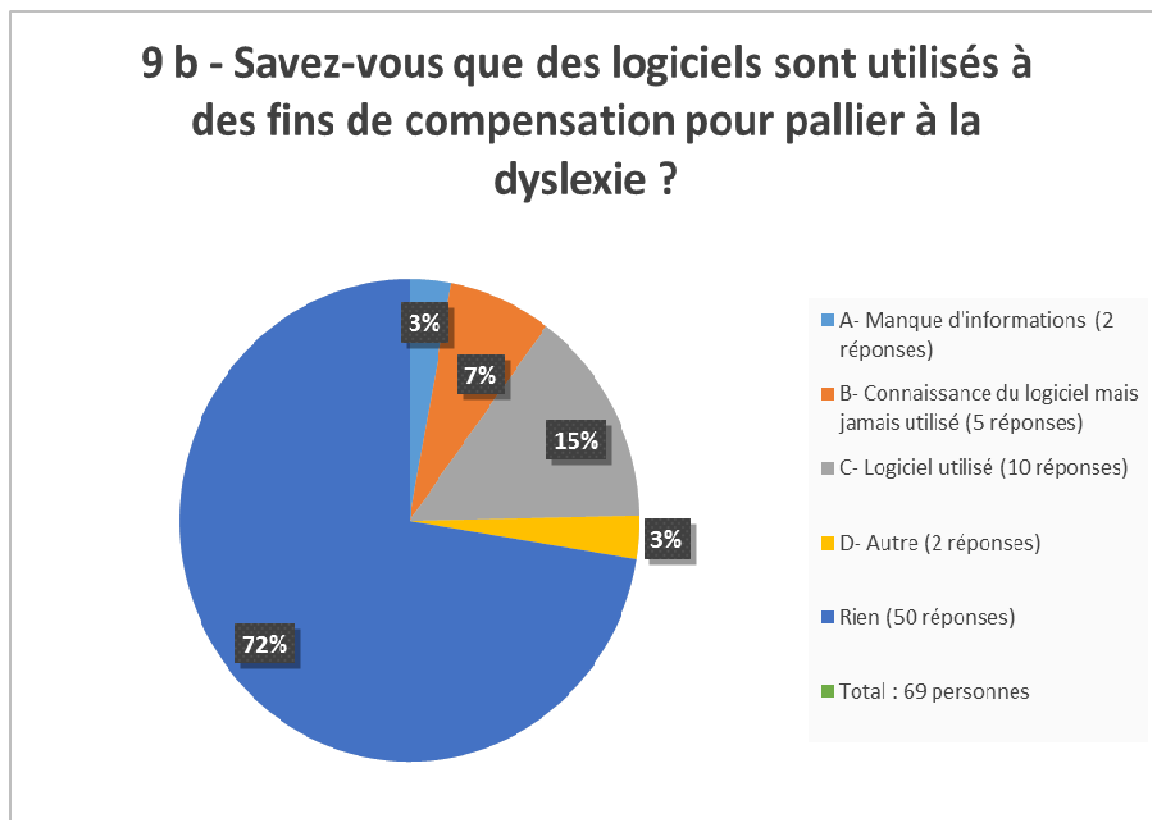
Question 9 : Savez-vous que des logiciels sont utilisés à des fins de compensation pour pallier à la dyslexie ?

Oui	1	69
Non	2	3

Les 69 personnes ayant répondues « Oui » peuvent accéder à la deuxième partie de la question.



Manque d'informations (concernant les logiciels et leurs utilisations)	A	2
Connaissance du logiciel mais jamais utilisé	B	5
Logiciel utilisé (régulièrement par les personnes dyslexiques)	C	10
Autre	D	2
Rien		50



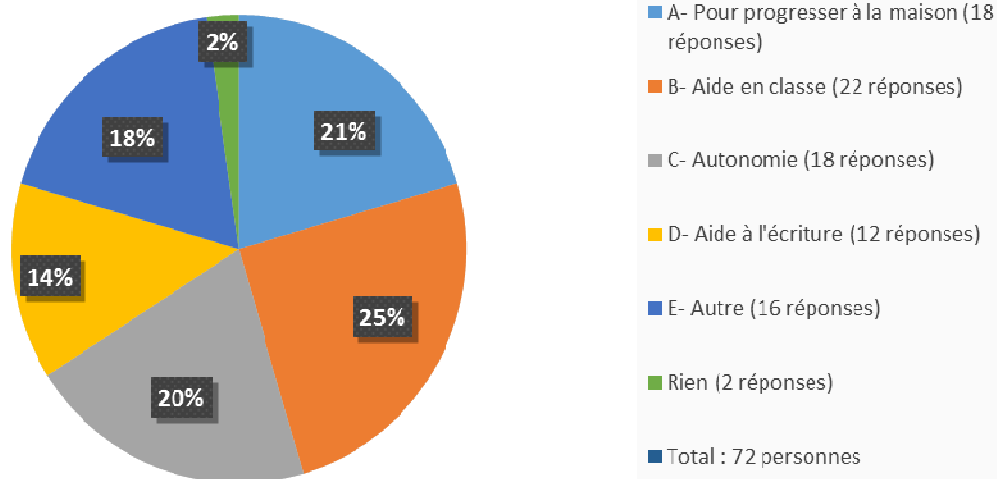
Question 10 : Dans quel cas pensez-vous qu'un outil informatique peut s'avérer utile à un enfant dyslexique ?

Pour progresser à la maison (pour faire les devoirs)	A	18
Aide en classe (permettre une meilleure compréhension et un appui pédagogique)	B	22
Autonomie	C	18
Aide à l'écriture (par les multiples fonctions que proposent les logiciels)	D	12
Autre	E	16
Rien		2

Commentaire : Une personne a répondu à la question 9 a mais pas à la question 10, ce qui n'est normalement pas possible car la personne est censée avoir répondu à toutes les questions obligatoires de la section 2. Cette personne est l'identifiant 102.

Commentaire 2 : Une personne a répondu à la question 9 a mais pas à la question 10 ainsi qu'aux questions suivantes. Cette personne est l'identifiant 35. Ces personnes sont comptabilisées dans les « Rien » des questions suivantes.

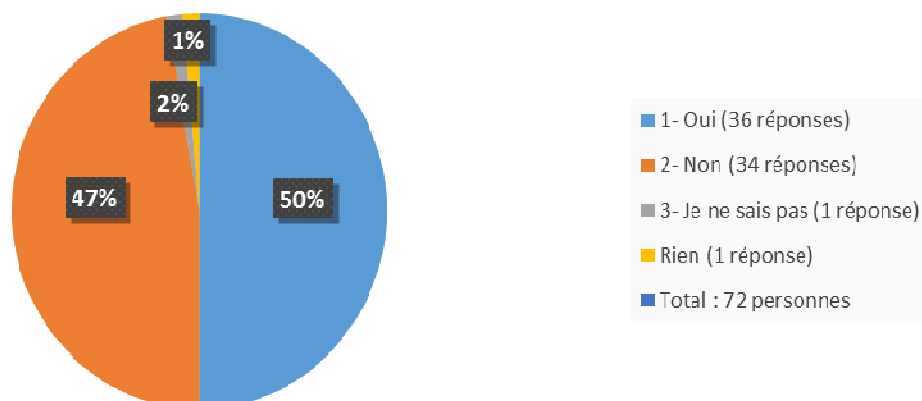
Question 10 - Dans quel cas pensez-vous qu'un outil informatique peut s'avérer utile à un enfant dyslexique ?



Question 11 : Votre enfant travaille-t-il à l'aide de tels outils ? Si oui, lesquels ?

Oui	1	36
Non	2	34
Je ne sais pas	3	1
Rien		1

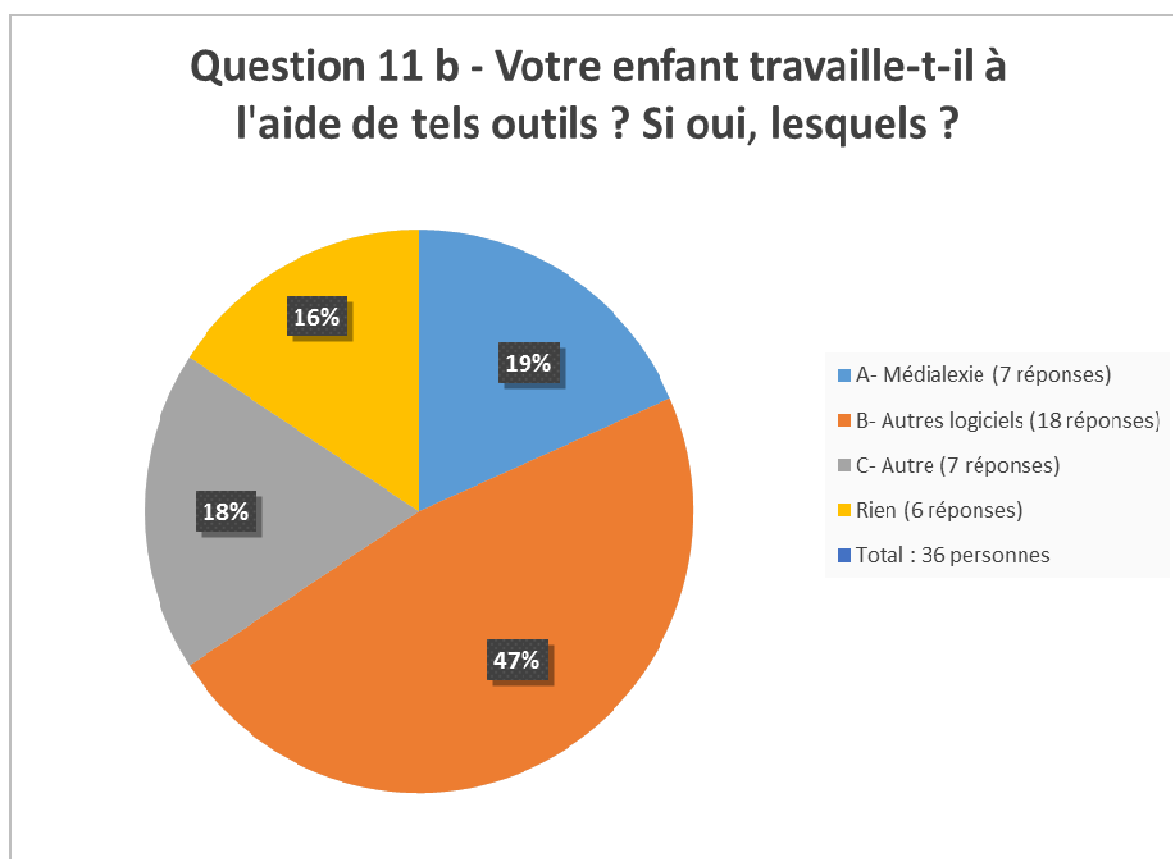
11 a - Votre enfant travaille-t-il à l'aide de tels outils ? Si oui, lesquels ?



Les 36 personnes ayant répondues « Oui » peuvent accéder à la deuxième partie de la question.

Médialexie	A	7
Autres logiciels (dragon, antidote, notability ...)	B	18
Autre	C	7
Rien		5

Exemples de réponses types qui ne correspondent pas à la question : « Seulement si je les installe » ou « Ce n'est pas faute de lui avoir proposé ». Il y a donc une difficulté à les classer dans une des catégories proposées.

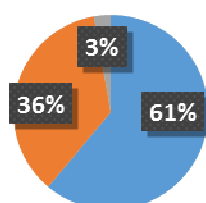


Question 11 bis : A-t-il bénéficié d'une formation concernant ce ou ces logiciels ? Vous a-t-elle parue nécessaire ? Efficace ?

Les 36 personnes ayant répondues « Oui » à la question 11 a peuvent accéder à l'ensemble de la question 11 bis.

Oui	1	22
Non	2	13
Je ne sais pas	3	1

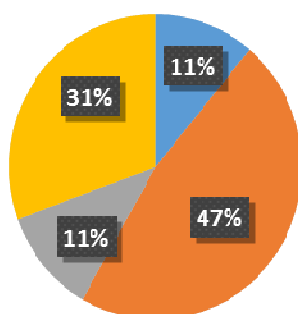
Question 11 bis a - A-t-il bénéficié d'une formation concernant ce ou ces logiciels ? Vous a-t-elle parue nécessaire ? Efficace ?



- 1- Oui (22 réponses)
- 2- Non (13 réponses)
- 3- Je ne sais pas (1 réponse)
- Total : 36 personnes

Aucune formation	A	4
Une formation utile	B	17
Une formation inefficace	C	4
Rien		11

Question 11 bis b - A-t-il bénéficié d'une formation concernant ce ou ces logiciels ? Vous-a-t-elle parue nécessaire ? Efficace ?

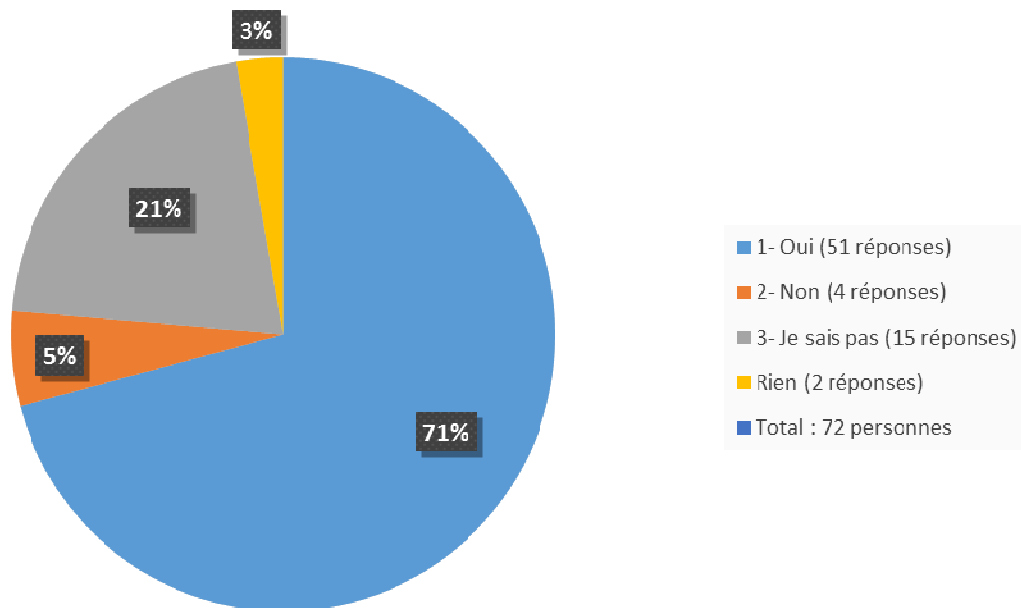


- A- Aucune formation (4 réponses)
- B- Une formation utile (17 réponses)
- C- Une formation inefficace (4 réponses)
- Rien (11 réponses)
- Total : 36 personnes

Question 12 : Pensez-vous que ce type d'aide logicielle est efficace ?

Oui	1	51
Non	2	4
Je ne sais pas	3	15
Rien		2

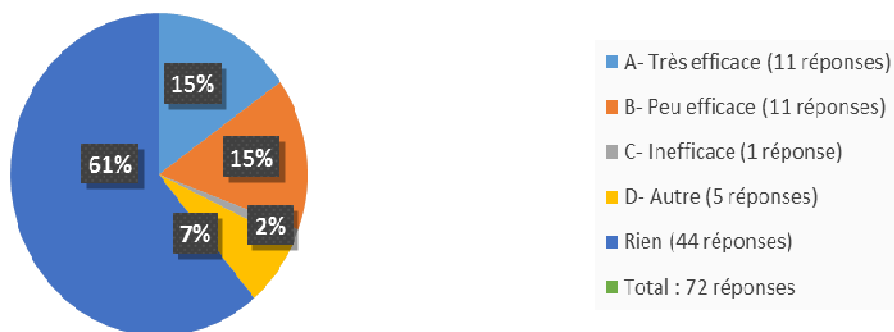
Question 12 a - Pensez-vous que ce type d'aide logicielle est efficace ?



Très efficace	A	11
Peu efficace	B	11
Inefficace	C	1
Autre	D	5
Rien		44

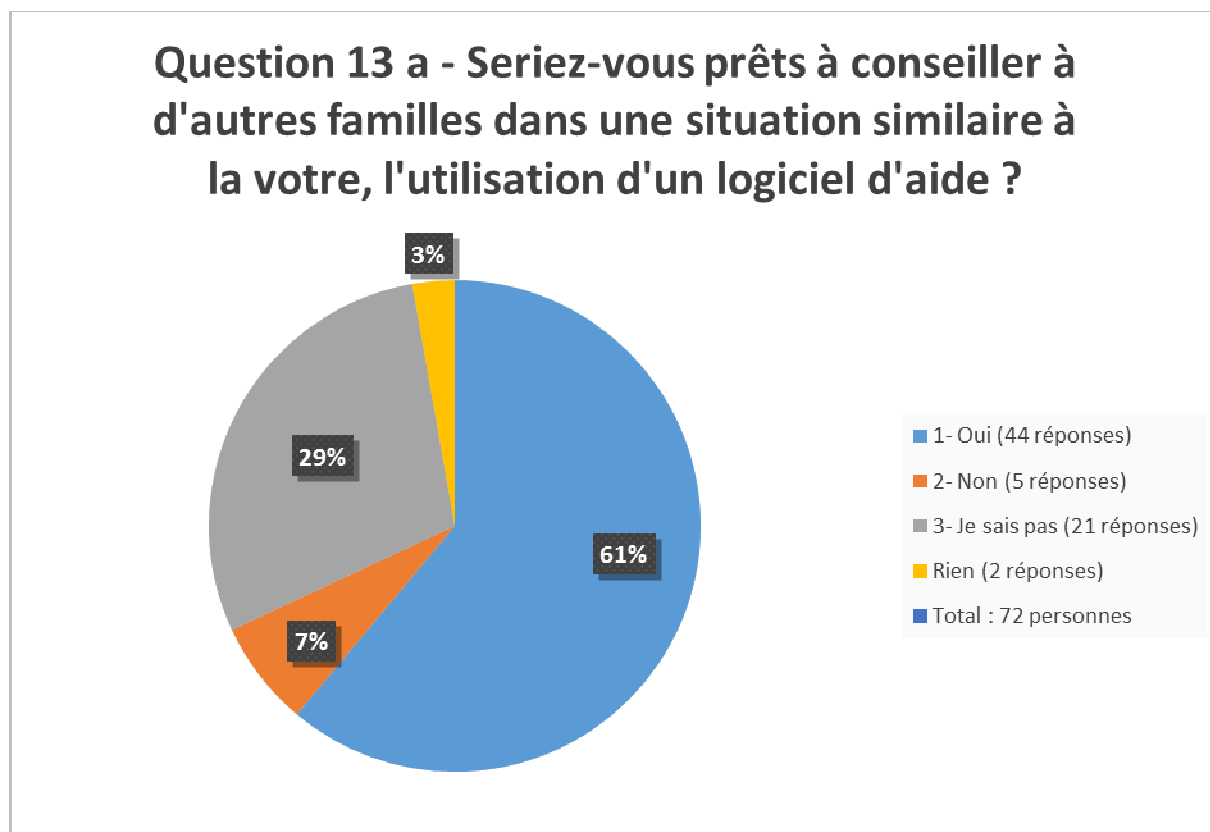
Exemple de réponse type qui ne correspond pas à la question : « Il va avoir Wody »

Question 12 b - Pensez-vous que ce type d'aide logicielle est efficace ?



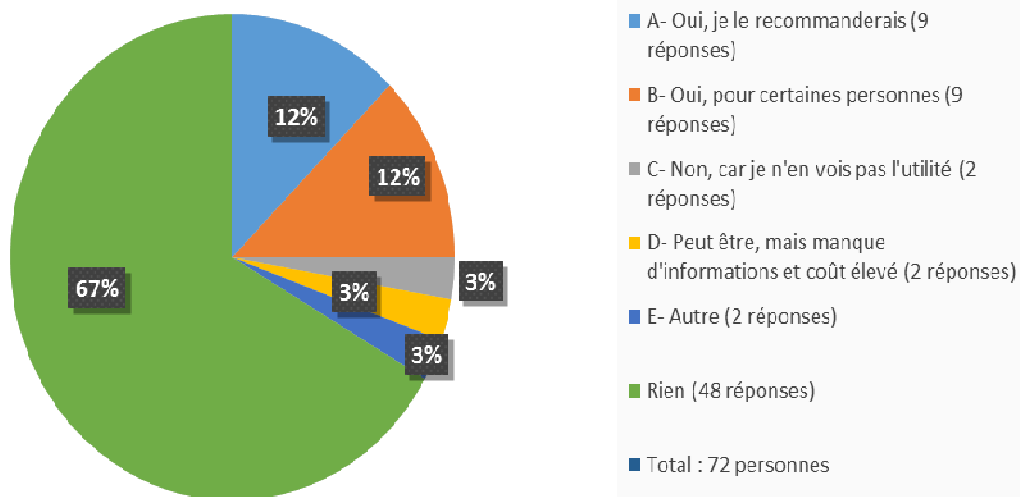
Question 13 : Seriez-vous prêts à conseiller à d'autres familles, dans une situation similaire à la vôtre, l'utilisation d'un logiciel d'aide ?

Oui	1	44
Non	2	5
Je ne sais pas	3	21
Rien		2



Oui, je le recommanderai	A	9
Oui, pour certaines personnes (cela n'est pas forcément efficace pour palier à tous les types de dyslexie)	B	9
Non, car je n'en vois pas l'utilité (logiciels pas vraiment adaptés et performants)	C	2
Peut-être mais manque d'informations et coût élevé (aucune aide financière de l'éducation nationale)	D	2
Autre	E	2
Rien		48

Question 13 b - Seriez-vous prêts à conseiller à d'autres familles, dans une situation similaire à la votre, l'utilisation d'un logiciel d'aide ?

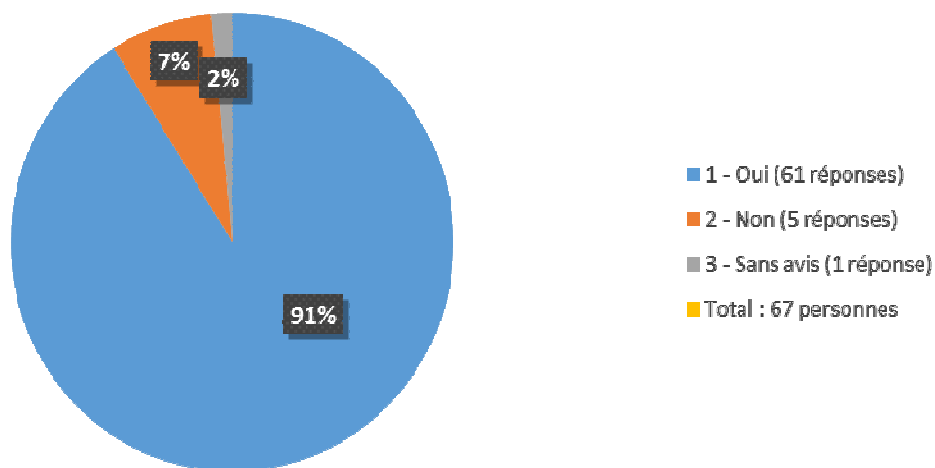


IV- Les possibilités des nouvelles technologies (67 personnes)

Question 14 : Seriez-vous favorable à l'idée d'utiliser un ordinateur dans une perspective de rééducation ?

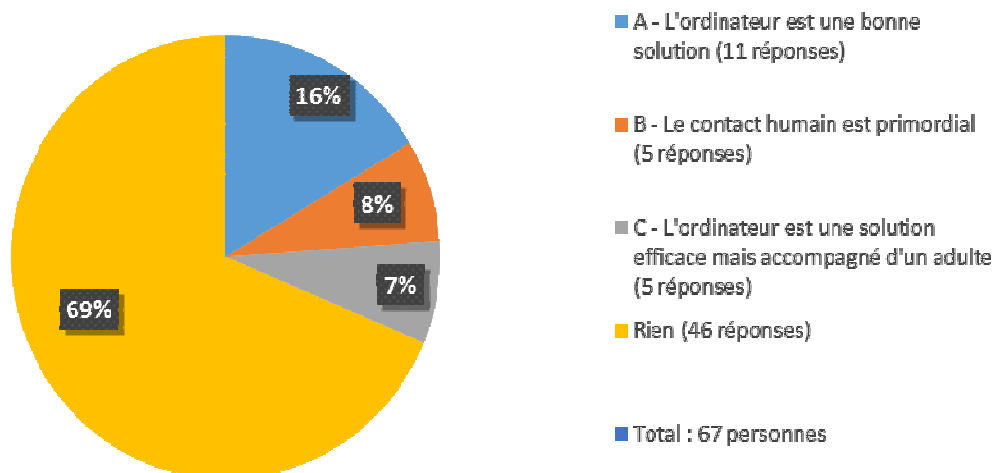
Oui	1	61
Non	2	5
Sans avis	3	1

Question 14 a : Seriez-vous favorable à l'idée d'utiliser un ordinateur dans une perspective de rééducation ?



L'ordinateur est une bonne solution	A	11
Le contact humain est primordial	B	5
L'ordinateur est une solution efficace mais accompagné d'un adulte	C	5
Rien		46

Question 14 b : Seriez-vous favorable à l'idée d'utiliser un ordinateur dans une perspective de rééducation ?



Question 15 : Trouveriez-vous qu'il y a une différence entre le travail effectué chez un professionnel de santé selon qu'il est équipé, ou non, d'un ordinateur doté d'un logiciel ?

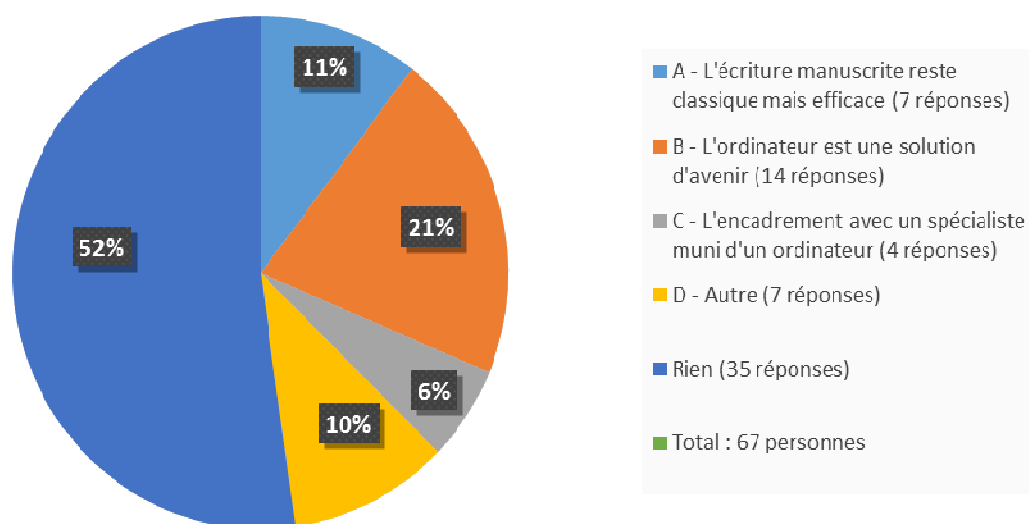
Oui	1	35
Non	2	6
Sans avis	3	26

Question 15 a : Trouveriez-vous qu'il y a une différence entre le travail effectué chez un professionnel de santé selon qu'il est équipé, ou non, d'un ordinateur doté d'un logiciel ?



L'écriture manuscrite reste classique mais efficace	A	7
L'ordinateur est une solution d'avenir	B	14
L'encadrement avec un spécialiste muni d'un ordinateur	C	4
Autre	D	7
Rien		35

Question 15 b : Trouveriez-vous qu'il y a une différence entre le travail effectué chez un professionnel de santé selon qu'il est équipé, ou non, d'un ordinateur doté d'un logiciel ?

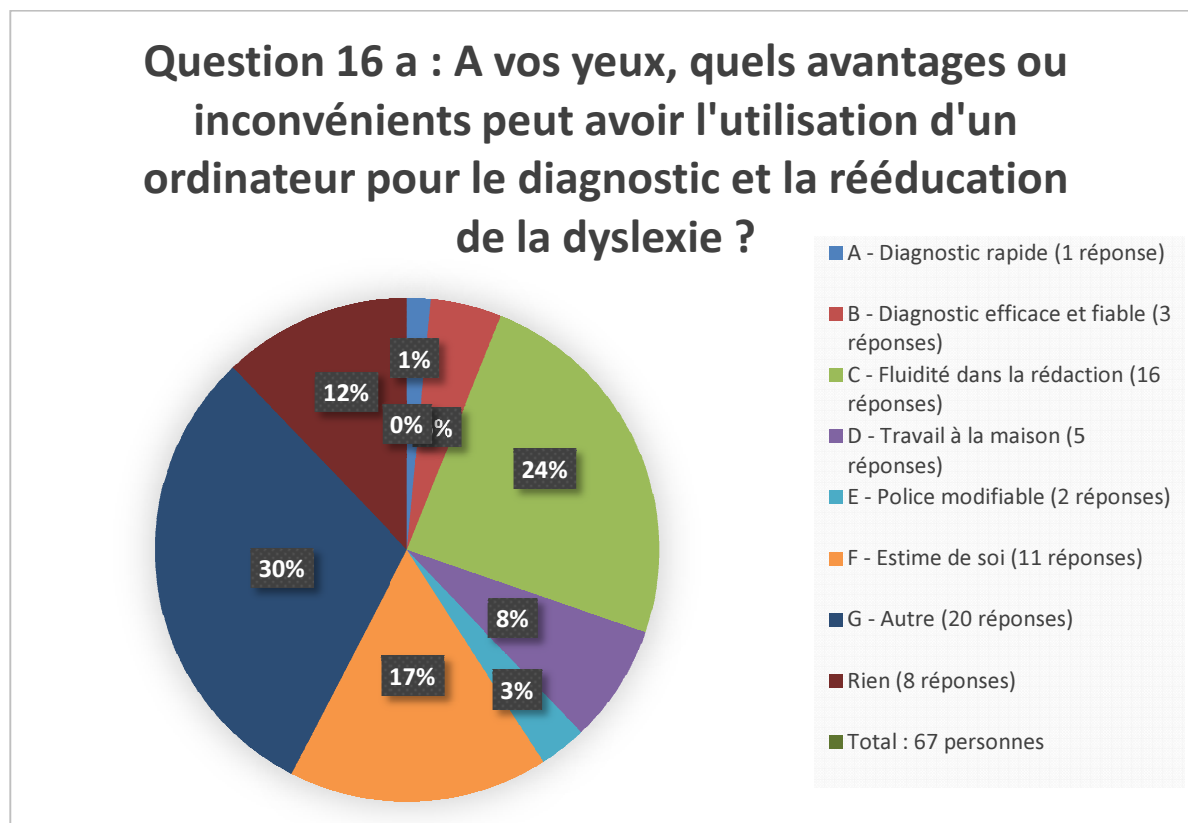


Question 16 : A vos yeux quels avantages ou inconvénients peut avoir l'utilisation d'un ordinateur pour le diagnostic et la rééducation de la dyslexie ?

Les avantages

Diagnostic rapide	A	1
Diagnostic efficace et fiable	B	3
Fluidité dans la rédaction (Orthographe, vocabulaire)	C	16
Travail à la maison (possibilité de progresser chez soi, à son rythme)	D	5
Police modifiable	E	2

(les différents caractères permettent de se s'adapter à chaque enfant)		
Estime de soi (redonner confiance)	F	11
Autre	G	20
Rien		8



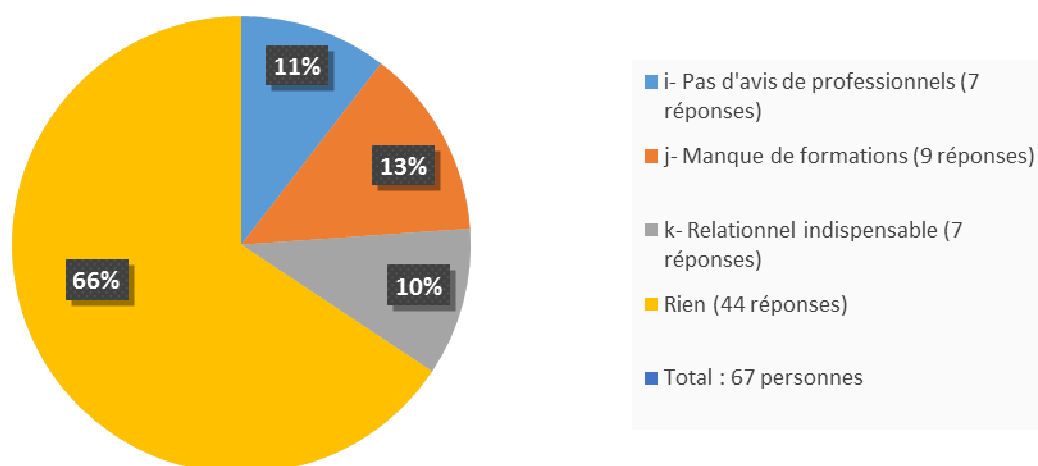
Les inconvénients

Pas d'avis de professionnel (familles livrées à elles-mêmes)	h	7
Manque de formations (pour le fonctionnement des logiciels)	i	9
Relationnel indispensable (l'ordinateur ne peut remplacer un être humain)	j	7
Rien		44

Commentaire : Cette question a été subdivisée en deux, c'est-à-dire que l'on a considéré qu'une personne devait répondre un avantage et un inconvénient. Cependant cela n'a pas toujours été le cas.

Exemple : L'identifiant 67 a écrit : « Quand ils sont petits les dyslexiques ont besoin d'une personne pour la rééducation mais après l'ordinateur peut être un complément ». Nous l'avons comptabilisé dans « Travail à la maison » et « Estime de soi ».

Question 16 b - A vos yeux, quels avantages ou inconvénients peut avoir l'utilisation d'un ordinateur pour le diagnostic et la rééducation de la dyslexie ?



Question 17 : Trouveriez-vous judicieux la possibilité pour un praticien de pouvoir intervenir à distance avec un ordinateur, via internet par exemple ?

Oui	1	32
Non	2	19
Sans avis	3	16

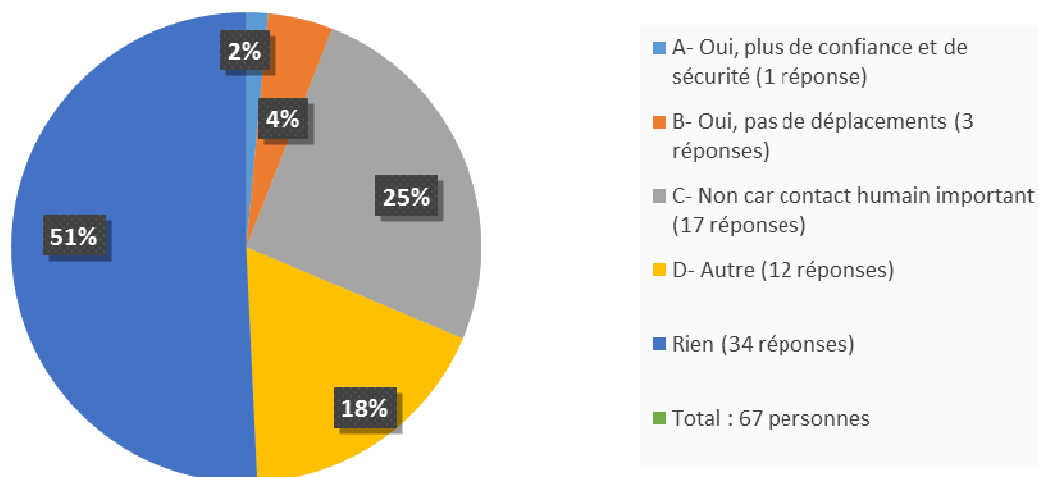
Question 17 a : Trouveriez-vous judicieux la possibilité pour un praticien de pouvoir intervenir à distance avec un ordinateur, via internet par exemple ?



Oui, plus de confiance et de sécurité (la présence d'un spécialiste est rassurante, même par internet)	A	1
Oui, pas de déplacements (pour éviter les temps de transports)	B	3
Non, car contact humain important	C	17
Autre	D	12
Rien		34

Exemple : L'identifiant 100 a répondu à cette question : « Cela permet de se sentir aidé si besoin ». Cette réponse ne rentrant dans aucune des catégories de métriques, elle a été comptabilisée dans « Autre ».

Question 17 b - Trouveriez-vous judicieux la possibilité pour un praticien de pouvoir intervenir à distance avec un ordinateur, via internet par exemple ?



Question 18 : Auriez-vous des idées de logiciels ou fonctionnalités de logiciels qui permettraient à votre enfant de vivre avec sa dyslexie ?

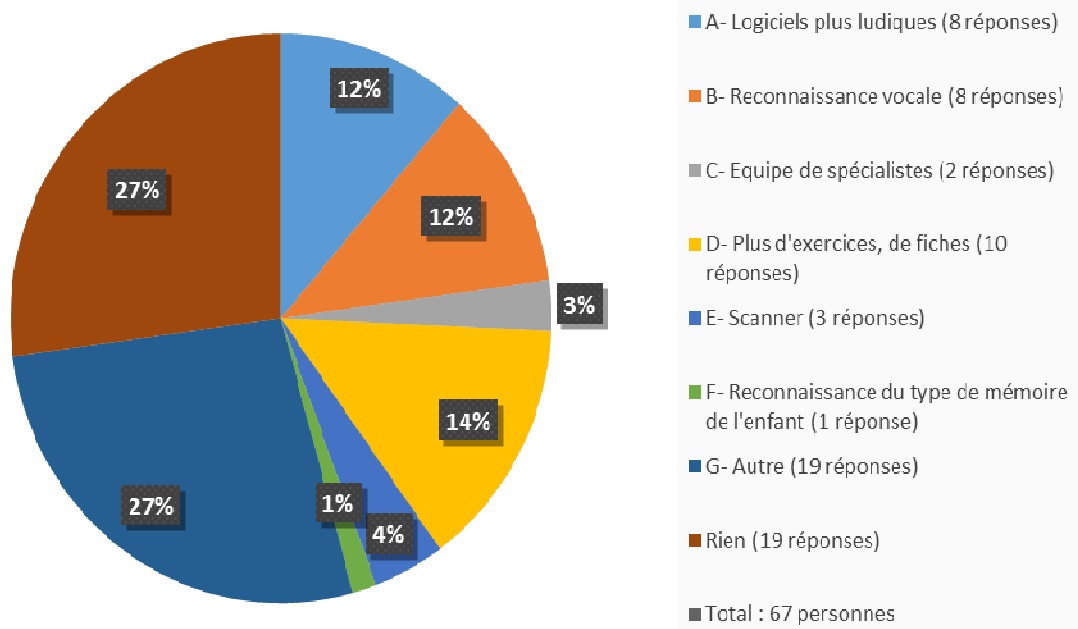
Logiciels plus ludique (progresser en s'amusant)	A	8
Reconnaissance vocale (méthode plus pratique pour l'enfant)	B	8
Equipe de spécialistes	C	2
Plus d'exercices, de fiches	D	10
Scanner (Bouton)	E	3
Reconnaissance du type de mémoire de l'enfant	F	1
Autre	G	19
Rien		19

Commentaire : Voici la réponse de l'identifiant 73 à la question 18 : « Ecriture intuitive, "reconnaissance vocale", proposition de fiches mentales selon le niveau scolaire ».

Cette réponse a donc été comptabilisé en deux réponses une pour la métrique B et une pour la métrique G car cette réponse est composé de plusieurs idées.

Ainsi le nombre de réponses à cette question est de 70 au lieu de 67 car deux autres personnes ont comptabilisé deux réponses. Ce sont les identifiants 57 et 68.

Question 18 - Auriez-vous des idées de logiciels ou fonctionnalités de logiciels qui permettraient à votre enfant de vivre avec sa dyslexie ?



C) Le tableau de recensement

Ci-dessous se trouve l'intégralité du tableau de recensement dans lequel a été évaluée, en accord avec les métriques détaillées dans la partie relative à l'analyse du questionnaire, l'intégralité des réponses données à notre enquête. On notera que chaque ligne correspond à l'enregistrement des réponses données par une personne, et que si cette personne n'a pas répondu à l'intégralité du questionnaire alors la partie de la ligne ne contenant plus de réponse est surlignée en bleu.

Question 3								Question 4								
0	1	2	3	A	B	C	D	A	B	C	d	e	f	g	h	i
			1		1					1	1					
			1				1	1							1	
		1								1						
			1	1					1						1	
			1					1	1		1					
		1		1					1					1	1	
			1		1					1						
			1			1				1						
			1	1											1	1
			1	1					1							
			1													
			1		1											1
			1													1
1																
			1													
			1						1						1	
			1							1				1		
			1	1												
			1	1												
			1													
			1													
			1	1												
			1	1												
			1	1												
			1													
			1													
			1	1				1				1			1	
			1												1	1
			1		1				1							

Question 5						Question 6				Question 7				
A	B	C	D	E	F	A	B	C	D	A	B	C	D	E
	1						1							1
				1		1						1		
	1				1	1						1		
	1				1	1				1	1	1		
			1				1						1	
	1		1			1					1	1		
					1				1					1
1	1					1		1					1	
	1					1							1	
	1		1		1	1					1		1	1
		1	1				1							1
	1			1		1								1
	1													
1		1			1	1				1		1		
		1			1				1		1		1	
	1	1				1							1	
	1			1		1		1						1
	1			1			1						1	
	1		1					1						1
					1	1						1		
					1	1							1	
					1			1					1	
	1		1			1	1							1
	1		1			1				1				
	1				1			1					1	1
					1			1					1	
	1				1	1	1			1				1
	1				1			1						1
			1					1						1
	1		1				1			1				1
	1				1	1						1		
	1				1	1				1	1	1		
	1				1				1		1	1		
	1	1	1			1	1						1	

	1				1		1								1
1	1				1	1									1
	1				1	1					1	1			
1	1								1	1				1	
1	1		1			1							1		
	1				1			1		1					
1	1						1					1			
1	1							1		1					
	1					1							1		
	1		1			1	1								1
	1		1			1	1								1
	1				1	1							1		
	1	1	1			1									1
	1							1				1			
					1		1								1
					1	1									1
					1	1									1
					1		1				1				
	1					1		1			1				1
	1					1		1							1
	1						1								1
	1					1						1			
	1		1				1								1
	1		1				1				1				1
					1			1		1		1			
					1			1		1		1			1
				1				1						1	1
	1			1				1				1	1		
7	52	8	21	10	26	39	19	23	6	15	10	19	24	28	

Question 8					Question 9					Question 10					
A	B	C	D	E	1	2	A	B	C	D	A	B	C	D	E
1					1				1		1				
			1		1								1		
				1	1							1			
	1				1				1			1	1		
				1	1							1			
	1				1								1		
		1			1				1					1	
	1		1		1				1						1
				1	1										1
		1			1						1	1			
			1		1							1			
1					1				1		1				
					1										
					1										
1					1								1		
				1	1			1							1
				1	1										1
	1				1							1			
			1		1						1				
				1	1				1			1			
				1	1										1
	1				1									1	
	1				1				1			1			
			1		1								1		
1	1				1										1
		1			1										1
		1			1				1					1	
				1	1						1				
				1	1							1			
	1				1									1	
	1	1			1										1
				1	1							1			
	1	1			1							1			
				1	1						1				

		1			1								1		
1					1							1			
				1	1								1		
		1				1								1	
				1	1									1	
				1	1								1		
1					1		1							1	
				1	1									1	
			1		1										1
1		1				1		1				1			
	1				1										1
				1	1								1		
				1	1							1	1	1	
		1			1							1			
		1				1									1
		1			1										
		1			1							1			
				1	1				1			1			
				1	1										
	1			1	1							1	1		
				1	1			1				1	1		
1					1									1	
				1	1										
		1			1				1					1	
		1	1		1			1							1
				1	1			1				1	1	1	
		1			1							1	1	1	
	1				1							1	1	1	
				1	1										1
		1													
1					1		1						1	1	
13	16	22	6	25	69	3	2	5	10	2	18	22	18	12	16

Question 11					Question 11 bis					Question 12								
1	2	3	A	B	C	1	2	3	A	B	C	1	2	3	A	B	C	D
1			1				1		1			1			1			
1			1			1				1		1			1			
1						1						1						
	1											1			1			
1						1						1				1		
	1											1						1
	1												1					
	1													1				
1				1		1				1		1			1			
	1											1			1			
	1											1						
	1													1				
1				1			1			1		1						
1				1				1			1	1				1		
1			1	1			1			1		1						
1				1		1				1		1				1		
1				1		1				1		1				1		
1			1			1					1	1						
	1											1						
	1													1				1
1				1			1			1		1						
1			1				1					1						
	1											1						
	1												1					1
1				1			1			1		1						1
1				1		1				1		1						1
1					1	1				1		1						1
1							1			1			1					
1				1		1				1		1			1			
	1											1			1			
	1											1						
	1													1				

	1											1						
1				1			1		1			1			1			
	1													1				
1				1			1		1			1				1		
	1											1						
	1													1				
	1											1						
	1													1				
	1													1				1
	1													1				
1				1			1							1				
1							1					1			1			
	1													1				
	1											1						
1						1												
1				1		1			1			1						
	1											1						
1				1	1	1			1			1			1			
1					1	1					1	1						
	1											1						
	1											1			1			
	1														1			
1					1		1						1					
1						1	1									1		
	1											1					1	
36	34	1	7	18	7	22	13	1	4	17	4	51	4	15	11	11	1	5

1															
1								1			1	1			
		1							1						
	1								1						
	1								1						
		1							1						
1									1						
1			1						1						
		1								1				1	
		1		1					1						
1										1					
		1								1					
1										1					
		1								1					
		1								1					
1															
1									1						
1									1			1			
1									1						
		1								1					1
1									1						
		1		1						1					
1									1						
		1								1					
		1								1					
1			1						1			1			
44	5	21	9	9	2	2	2	2	61	5	1	11	5	5	

Question 15							Question 16									
1	2	3	A	B	C	D	A	B	C	D	E	F	G	h	i	j
1				1												1
		1											1			
1				1										1		
1					1											
1				1										1		
1													1			
1				1					1		1					
1				1												
1																
1																
1							1		1							
1																
		1	1										1			
1											1			1		
		1												1		1
														1		
1						1								1		
	1			1									1			1
1									1							1
1														1		
		1												1	1	
	1									1				1		
1										1		1				
1				1									1			1
1														1		
1			1	1									1			
		1				1				1						
		1							1							1
1			1	1									1			
		1							1							1
		1				1								1		
		1						1								
		1							1					1		
		1								1						
		1												1		
		1														
		1														

		1											1			1	
	1																
1				1									1				
	1															1	
		1													1		
		1				1											1
		1											1				1
		1				1			1								1
1															1		
		1							1								
1															1	1	
	1			1					1								
		1														1	
1													1				
	1			1					1								
1			1	1					1								
		1	1						1								1
1																	1
1											1						
1				1					1								
1										1							
		1												1			
1											1						
		1															
1				1													
35	6	26	7	14	4	7	1	3	16	5	2	11	20	7	9	7	

Question 17						Question 18							
1	2	3	A	B	C	D	A	B	C	D	E	F	G
	1					1							1
1					1								
1							1				1		
	1					1							1
1							1						
1													
1									1				
1							1						
1								1					
1								1					
	1					1					1		
1								1					
		1				1						1	
1									1				
1													1
	1					1							1
		1				1				1			
1													
1													
		1					1		1				1
		1					1					1	
1						1					1		
1													1
		1				1							1
	1					1					1		
	1												
		1						1					

III) Éléments des différents prototypes

Dans ce chapitre des annexes, nous allons exposer des éléments appartenant aux différents prototypes que nous avons pu développer. Ceci a pour but de produire des éléments techniques représentatifs des implémentations réalisées.

A) Quelques fichiers d'entête

Dans la présente section, quelques uns des fichiers d'entêtes de nos prototypes en C++ utilisant les MFC 9.0 (Microsoft, 2008a) sont présentés. En particulier, certains de ces fichiers précisent les structures de données utilisées pour les différents traitements

1) DLL_constantes.h

Le premier fichier que nous souhaitons introduire est celui qui définit les différentes constantes et dépendances nécessaire au fonctionnement des différents composants, compilés sous la forme de DLL, du noyau fonctionnel. On signalera que ce fichiers provoque le chargement des constantes spécifiques :

- Au suivi oculaire, décrites dans le fichier d'entête **EYE_constantes.h**.
- A l'enregistrement, décrites dans le fichier d'entête **EEG_constantes.h**.

```
#ifndef CONSTANTES_H
#define CONSTANTES_H

/////////////////////////////////////////////////////////////////
// Standard includes
#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS
// some CString constructors will be explicit

#include <afxwin.h> // MFC core and standard components
#include <afxext.h> // MFC extensions
#include <afxole.h> // MFC OLE classes
#include <afxodlgs.h> // MFC OLE dialog classes
#include <afxdisp.h> // MFC Automation classes
#include <afxdtctl.h> // MFC support for Internet Explorer 4 Common Controls

/////////////////////////////////////////////////////////////////
// Others includes
#include "targetver.h"

#include <list>
#include <iostream>
#include <iomanip>
#include <fstream>
#include <conio.h>
#include <sstream>
#include <windows.h>
#include <map>
#include <math.h>

/////////////////////////////////////////////////////////////////
// Multimodal specific includes
#include "EEG_constants.h" // EEG
#include "NLP_constants.h" // Natural Language Processing
#include "EYE_constants.h" // Eye Tracking

/////////////////////////////////////////////////////////////////
// Core handled messages
#define WM_USER_CORE (WM_USER+1000)
#define MSG_CORE_EEG (WM_USER_CORE)
//WPARAM = message type ; LPARAM = associated value
#define MSG_CORE_NLP (WM_USER_CORE+1)
#define MSG_CORE_EYE (WM_USER_CORE+2)

/////////////////////////////////////////////////////////////////
// Main application
#define MAIN_APP ((CProtoERPEmotivApp *) ::AfxGetApp())

/////////////////////////////////////////////////////////////////
// String formats
#define FORMAT(arg) STR(arg)
#define STR(arg) #arg

#define TIME %.20lf

#endif
```

2) IHardwareReader.h

Ce fichier d'entête décrit l'interface que les modules de traitements des différentes modalités doivent implémenter. En particulier, on remarque que la communication par message utilise le système de messages de l'API Win32 (Microsoft, 2010).

```
#ifndef IHWAREREADER_H
#define IHWAREREADER_H

#include "..\Commun\DLL_constantes.h"

class IHardwareReader : public CWinThread
{
protected:
    IHardwareReader(void);
    virtual ~IHardwareReader(void);

    virtual BOOL InitInstance();
    virtual BOOL PumpMessages();

    DECLARE_MESSAGE_MAP()

    virtual afx_msg void OnMessageStart(WPARAM vParam, LPARAM lParam) = 0;
    virtual afx_msg void OnMessageStop(WPARAM vParam, LPARAM lParam) = 0;
};

#endif
```

3) EYE_constantes.h

Les contraintes et définitions spécifiques au composant de suivi oculaire disposent de leur propre fichier d'entête. On notera que ce fichier provoque l'inclusion de la bibliothèque OpenCV (Intel & Willow Garage, 2013).

```

#ifndef EYE_CONSTANTES_H
#define EYE_CONSTANTES_H

////////////////////////////////////
// OpenCV SDK
#include <opencv/cv.h>
#include <opencv2/objdetect/objdetect.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
using namespace cv;

////////////////////////////////////
// Eye Tracking recorded channels
#define EYE_CLOSE_CAMERA_NUMBER (1)
#define EYE_FAR_CAMERA_NUMBER (0)
#define EYE_CAMERA_NUMBER (EYE_CLOSE_CAMERA_NUMBER+EYE_FAR_CAMERA_NUMBER)
#define EYE_CAPTURE_WIDTH (160) //(320) //(640)
#define EYE_CAPTURE_HEIGHT (120) //(240) //(480)
#define EYE_GRAY_LEVEL_NUMBER (256)
#define EYE_SEGMENTATION_DARK_THRESHOLD ((EYE_GRAY_LEVEL_NUMBER-1)/(NCLASS-1))
//darkest threshold (=> supposed pupil)
#define EYE_SEGMENTATION_LEVELSET_Na (3) //(20)
#define EYE_SEGMENTATION_LEVELSET_Ng (3) //(5) //(7)
#define EYE_SEGMENTATION_GAUSS_NEIGHBOURS (EYE_SEGMENTATION_LEVELSET_Ng)
#define EYE_SEGMENTATION_GAUSS_SIGMA (1)
#define EYE_SEGMENTATION_IN (-1)
#define EYE_SEGMENTATION_OUT (1)
#define EYE_RESOLUTION_MAX_AMPLITUDE (10000)
#define EYE_REFRESH_TIMER (33) //ms

////////////////////////////////////
// Eye Tracking handled messages
#define WM_USER_EYE (WM_USER+300)

#define MSG_EYE_START (WM_USER_EYE)
#define MSG_EYE_STOP (WM_USER_EYE+1)
#define MSG_EYE_STARTRECORD (WM_USER_EYE+2)
#define MSG_EYE_STOPRECORD (WM_USER_EYE+3)
#define MSG_EYE_STOPALLRECORDS (WM_USER_EYE+4)
#define MSG_EYE_STOPANDSTARTRECORD (WM_USER_EYE+5)

////////////////////////////////////
// Eye Tracking handled parameters
#define EYE_LPARAM LPARAM
//LOWORD = EYE_VALUE_TYPE ; HIWORD = RECORD_LIST.recordID:

////////////////////////////////////
// Eye Tracking data structure types
#define EYE_VALUE_TYPE UINT
#define EYEVT_INVALID 0
#define EYEVT_RAW 1
#define EYEVT_INTERPOLATED 2
#define EYEVT_COORDINATES 3

```



```

////////////////////////////////////
// Eye Tracking raw data structures
typedef struct tEYE_RECORD
    //values
{
    double time;                //sample (absolute) time
    EYE_VALUE_TYPE type;        //sample type
    UINT weight;                //weight of averaged values
    UINT partialWeight;         //weight of averaged raw values
    IplImage * closeImages[MAX(EYE_CLOSE_CAMERA_NUMBER, 1)]; //raw images
    IplImage * farImages[MAX(EYE_FAR_CAMERA_NUMBER, 1)];
    double X, Y;                //extrapolated point of regard
    tEYE_RECORD * next;         //next value
} EYE_RECORD;

typedef struct tEYE_RECORD_LIST //records
{
    int recordID;                //record ID
    TCHAR name[256];             //record name
    EYE_VALUE_TYPE type;         //record type
    bool isRecording;            //false = record complete
    double startTime;            //starting (absolute) time
    double duration;             //duration
    double synchronizationTime; //stimulation (relative) time
    UINT maxWeight;              //max weight of averaged values
    UINT maxPartialWeight;       //max weight of averaged raw values
    double minX, minY;           //min values
    double maxX, maxY;           //max values
    int valueNumber;             //value number
    EYE_RECORD * recordHead;      //first value of the record
    EYE_RECORD * recordQueue;     //last updated value of the record
    tEYE_RECORD_LIST * next;      //next record
} EYE_RECORD_LIST;

////////////////////////////////////
// Eye Tracking function handlers
typedef EYE_RECORD_LIST * (WINAPI * tNewEYE_RECORD_LIST)
    (int, EYE_VALUE_TYPE, TCHAR *, bool, EYE_RECORD_LIST *);
typedef void (WINAPI * tFreeEYE_RECORD_LIST)(EYE_RECORD_LIST *);
typedef EYE_RECORD * (WINAPI * tNewEYE_RECORD)
    (EYE_RECORD_LIST *, double, IplImage **, IplImage **,
     double, double, UINT, UINT, EYE_VALUE_TYPE, EYE_RECORD **);
typedef void (WINAPI * tFreeEYE_RECORD)(EYE_RECORD *);

////////////////////////////////////
// Eye Tracking main application (thread)
#define          MAIN_EYE_APP          MAIN_APP

#endif

```

4) EYEReader.h

Ce fichier d'entête décrit l'implémentation de l'interface des composants du noyau fonctionnel, définie dans de le fichier d'entête **IHardwareReader.h**, pour le traitement du suivi oculaire. On remarquera que ce fichier provoque l'inclusion d'autres fichiers d'entête

spécifiques aux traitements mis en œuvre pour cette analyse (voir la section **Le suivi oculaire**).

```
#ifndef EYEREADER_H
#define EYEREADER_H

#include "IHardwareReader.h"

#include "Seuillage_Otsu_Liao.h"
#include "LevelSet_Shi.h"

class EYEReader : public IHardwareReader
{
public:
    EYEReader(void);
    virtual ~EYEReader(void);

    //Records
    EYE_RECORD_LIST * recordList;

protected:
    int nextRecordID;

    void LoggingRecord(EYE_RECORD_LIST * record, bool fullReport=false);

    DECLARE_MESSAGE_MAP()

    afx_msg void OnMessageStart(WPARAM vParam, LPARAM lParam);
    afx_msg void OnMessageStop(WPARAM vParam, LPARAM lParam);
    afx_msg void OnMessageStartRecord(WPARAM vParam, LPARAM lParam);
    afx_msg void OnMessageStopRecord(WPARAM vParam, LPARAM lParam);
    afx_msg void OnMessageStopAllRecords(WPARAM vParam, LPARAM lParam);
    afx_msg void OnMessageStopAndStartRecord(WPARAM vParam, LPARAM lParam);
};

#endif
```

5) EEG_constantes.h

Les contraintes et définitions spécifiques au composant d'enregistrement EEG disposent également de leur propre fichier d'entête. On notera que ce fichier provoque l'inclusion de l'Emotiv SDK (Emotiv Inc., 2013).

```

#ifndef EEG_CONSTANTES_H
#define EEG_CONSTANTES_H

/////////////////////////////////////////////////////////////////
// Emotiv Epoc SDK
#include "include\emotiv\EmoStateDLL.h"
#include "include\emotiv\edk.h"
#include "include\emotiv\edkErrorCode.h"

/////////////////////////////////////////////////////////////////
// EEG recorded channels
#define ED_CHANNELS ED_AF3, ED_F7, ED_F3, ED_FC5, ED_T7,
ED_P7, ED_O1, ED_O2, ED_P8, ED_T8, ED_FC6, ED_F4, ED_F8, ED_AF4
#define EEG_CHANNEL_NUMBER 14
#define EEG_VOLTAGE_MAX_AMPLITUDE (10000)
#define EEG_TIMER_EPSILON (0.0005) //based a 2kHz sampling rate

/////////////////////////////////////////////////////////////////
// EEG handled messages
#define WM_USER_EEG (WM_USER+100)

#define MSG_EEG_START (WM_USER_EEG)
#define MSG_EEG_STOP (WM_USER_EEG+1)
#define MSG_EEG_STARTRECORD (WM_USER_EEG+2)
#define MSG_EEG_STOPRECORD (WM_USER_EEG+3)
#define MSG_EEG_STOPALLRECORDS (WM_USER_EEG+4)
#define MSG_EEG_STOPANDSTARTRECORD (WM_USER_EEG+5)
#define MSG_EEG_CALCULATINGERPS (WM_USER_EEG+6)

/////////////////////////////////////////////////////////////////
// EEG handled parameters
#define EEG_LPARAM LPARAM
//LOWORD = EEG_VALUE_TYPE ; HIWORD = EEG_RECORD_LIST.recordID:

/////////////////////////////////////////////////////////////////
// EEG data structure types
#define EEG_VALUE_TYPE UINT

#define EEGVT_INVALID 0
#define EEGVT_RAW 1
#define EEGVT_INTERPOLATED 2
#define EEGVT_ERP 3

/////////////////////////////////////////////////////////////////
// EEG raw data structures
typedef struct tEEG_RECORD //values
{
    double time; //sample (absolute) time
    EEG_VALUE_TYPE type; //sample type
    UINT weight; //weight of averaged values
    UINT partialWeight; //weight of averaged raw values
    double raws[EEG_CHANNEL_NUMBER]; //raw datas
    tEEG_RECORD * next; //next value
} EEG_RECORD;

```

```
typedef struct tEEG_RECORD_LIST //records
{
    int recordID; //record ID
    TCHAR name[256]; //record name
    EEG_VALUE_TYPE type; //record type
    bool isRecording; //false = record complete
    double startTime; //starting (absolute) time
    double duration; //duration
    double synchronizationTime; //stimulation (relative) time
    UINT maxWeight; //max weight of averaged values
    UINT maxPartialWeight; //max weight of averaged raw values
    double minRaws[EEG_CHANNEL_NUMBER]; //min value for each channel recorded
    double maxRaws[EEG_CHANNEL_NUMBER]; //max value for each channel recorded
    int valueNumber; //value number
    EEG_RECORD * recordHead; //first value of the record
    EEG_RECORD * recordQueue; //last updated value of the record
    tEEG_RECORD_LIST * next; //next record
} EEG_RECORD_LIST;

////////////////////////////////////
// EEG function handlers
typedef EEG_RECORD_LIST * (WINAPI * tNewEEG_RECORD_LIST)(int, EEG_VALUE_TYPE,
TCHAR *, bool, EEG_RECORD_LIST *);
typedef void (WINAPI * tFreeEEG_RECORD_LIST)(EEG_RECORD_LIST *);
typedef EEG_RECORD * (WINAPI * tNewEEG_RECORD)(EEG_RECORD_LIST *, int, double,
double **, UINT, UINT, EEG_VALUE_TYPE, EEG_RECORD **);
typedef void (WINAPI * tFreeEEG_RECORD)(EEG_RECORD *);

////////////////////////////////////
// EEG main application (thread)
#define MAIN_EEG_APP MAIN_APP

#endif
```

6) EEGReader.h

Ce dernier fichier d'entête décrit l'implémentation de l'interface des composants du noyau fonctionnel, définie dans de le fichier d'entête **IHardwareReader.h**, pour l'enregistrement des signaux EEG.

```
#ifndef EEGREADER_H
#define EEGREADER_H

#include "IHardwareReader.h"

#include "Interpolation_CS.h"

class EEGReader : public IHardwareReader
{
public:
    EEGReader(void);
    virtual ~EEGReader(void);

    //Channel lists
    static EE_DataChannel_t rawChannels[EEG_CHANNEL_NUMBER];
    static EE_DataChannel_t targetChannelList[EEG_CHANNEL_NUMBER+1];

    //Records
    EEG_RECORD_LIST * recordList;

protected:
    EmoEngineEventHandle eEvent;
    EmoStateHandle eState;

    int nextRecordID, nextERPID;

    void LoggingRecord(EEG_RECORD_LIST * record, bool fullReport=false);
    void SynchronizingRecords();
    void CorrectingAmplitude(EEG_RECORD_LIST * record);
    EEG_RECORD_LIST * NormalizingRecords();
    EEG_RECORD_LIST * Interpolation_CS(EEG_RECORD_LIST * record,
        EEG_RECORD_LIST * erp, EE_DataChannel_t * targetChannelList);
    void InterpolatingRecord(EEG_RECORD_LIST * record,
        EEG_RECORD_LIST * erp);
    void AveragingERP(EEG_RECORD_LIST * erp);

    DECLARE_MESSAGE_MAP()

    afx_msg void OnMessageStart(WPARAM vParam, LPARAM lParam);
    afx_msg void OnMessageStop(WPARAM vParam, LPARAM lParam);
    afx_msg void OnMessageStartRecord(WPARAM vParam, LPARAM lParam);
    afx_msg void OnMessageStopRecord(WPARAM vParam, LPARAM lParam);
    afx_msg void OnMessageStopAllRecords(WPARAM vParam, LPARAM lParam);
    afx_msg void OnMessageStopAndStartRecord(WPARAM vParam, LPARAM lParam);
    afx_msg void OnMessageCalculatingERPs(WPARAM vParam, LPARAM lParam);
};

#endif
```

B) Les listes de mots du test de lecture de mots fréquents et peu fréquents

Ci-dessous sont rassemblées les différentes listes de mots utilisés pour notre implémentation du test de lecture de mots fréquents et peu fréquents extrait de la méthode

B.A.L.E. (Jacquier-Roux et al., 2010). Chaque liste contient vingt mots qui seront successivement affichés dans l'ordre dans lequel ils sont présentés au sein de la liste à laquelle ils appartiennent.

	Réguliers	Irréguliers	Pseudo-mots
Fréquents	Faute Nuit Vague Montagne Soin Soif Mal Sauvage Mission Fuite Elan Animé Talon Splendeur Maman Pardon Caravelle Electron Jaloux Envoyé	Femme Hier Ville Monsieur Sept Août Dix Seconde Million Fusil Echo Tronc Tabac Orchestre Moyen Parfum Cacahuète Equateur Gentil Examen	Sande Chon Givor Bondeuse Sule Toir Mic Taubage Mardion Fudin Esan Trane Tagin Splindron Modan Tandir Taparelle Abinqueur Gental Ontage
Peu fréquents	Sac Congé Dorade Rigueur Asile Approche Piège Bottine Hausse Astronome Alchimie Avanie Courroie Baril Cargo Esquif Cric Cagoule Acrobate Bise	Net Galop Dolmen Respect Bourg Aiguille Poêle Baptême Oignon Aquarelle Orchidée Agenda Compteur Stand Toast Escroc Cake Chorale Aquarium Paon	Rac Gavin Caldon Rigende Plour Vatriche Piscal Bertale Aivron Pacirande Anchoyée Agante Courlone Stipe Torac Casine Bate Coginte Abranise Glon

IV) Le prototypage de l'interface multimodale

Afin de mettre à la disposition des praticiens un outil qui permette la génération rapide de métriques objectives et fiables, nous avons imaginé une interface multimodale reposant sur l'utilisation d'un matériel unique rassemblant l'ensemble des capteurs nécessaires à l'acquisition en parallèle de différentes modalités. Une telle centralisation de tous les capteurs et transducteur n'est pas triviale et nécessite une étude des modalités prise isolément avant d'analyser les interactions qui peuvent subvenir entre elles. A termes, l'idéal serait d'obtenir un matériel aux branchements et à l'utilisation excessivement simple permettant de procéder en une seule fois à l'acquisition de données provenant de multiples modalités qui actuellement nécessitent différents examens réalisés par différents spécialistes. Nous espérons ainsi voir l'arsenal des outils de diagnostic s'étoffer d'un matériel bon marché qui en augmenterait l'établissement et la précision, tout en systématisant le recours à certains indicateurs de la pathologie qui sont bien souvent négligés faute de moyens.

Dans un premier temps, nous avons ainsi établi un cahier des charges dans le but de lister tous les besoins liés à notre prototype de casque permettant le suivi oculaire (comme, par exemple, ceci est réalisé succinctement dans (Schneider, 2011)). Ce travail constitue la base d'un travail destiné à une amélioration progressive et itérative du casque d'une manière similaire, par exemple, à ce qui a été fait dans le projet openEyes (Li et al., 2006).

Nous signalerons également que nous avons fait un cahier des charges global, c'est-à-dire qui englobait toutes les modalités envisagés pour notre interface multimodale (voir le chapitre **L'intégration métier verticale : support de la multimodalité**), mais que nous faisons apparaître ici uniquement la partie concernant la modalité de suivi oculaire qui est la plus aboutie à l'heure actuelle. Néanmoins, certains éléments de ce cahier des charges global peuvent être consultés dans la partie de ce manuscrit dédiée aux perspectives de notre travail (voir le chapitre **La réalisation d'un casque « tout-en-un »**).

A) Le cahier des charges fonctionnel du suivi oculaire

I- Informations générales

1- Description du projet

a- Contexte du projet

Dans le cadre d'un projet de thèse il est envisagé de développer un appareil permettant l'**acquisition multimodale** de données issues de patients atteints de **troubles neuropsychologiques**, tels que la dyslexie. Cet appareil aura pour but d'être utilisé pour la détection et la prise en charge de ce genre de pathologies, qui nécessite la réalisation de **bilans médicaux** (et paramédicaux) pluridisciplinaires.

En effet chacun de ses bilans s'attache, conformément à la discipline du praticien le faisant passer au patient, uniquement à recueillir des données selon certaines modalités. Les

données obtenues de la sorte peuvent provenir d'**observations comportementales ou physiologiques** du patient selon le type de bilan. Or on notera, de par la nature même des fonctions altérées pas les troubles neuropsychologiques, que la grande majorité des bilans réalisés s'appuient sur des paradigmes expérimentaux communs. Il paraît donc pertinent de vouloir mutualiser ce qui peut l'être dans une perspective d'efficacité, tout en réalisation de façon automatique (via l'électronique ou l'informatique) certaines tâches de bas niveau partiellement décorrélées de l'expertise du praticien (voir [Annexe 2](#)).

Evidemment, pour chacune de ces catégories de bilans, il existe bien souvent un matériel permettant l'acquisition des données provenant de la modalité appropriée. Cependant aucun appareil existant ne couvre, à lui seul, l'ensemble des modalités nécessaires et suffisantes à l'ensemble des bilans. D'autre part, l'acquisition de multiples matériels, très onéreuse, est bien souvent réservée au milieu de la recherche ou aux cliniques spécialisées. Dans le cas présent nous souhaitons analyser des modalités permettant des observations comportementales et physiologiques, par électroencéphalographie, suivi oculaire et enregistrement de productions verbales. Dans le même temps, nous souhaitons pouvoir proposer à l'utilisateur des stimuli de nature visuelle et auditive.

C'est dans ce contexte que nous nous sommes fixé pour objectif d'utiliser des technologies de pointe en matière d'**imagerie cérébrale**, de **réalité virtuelle**, de **traitement de signaux et données** et d'**ergonomie** afin de développer un dispositif de nouvelle génération dans le domaine du traitement des troubles neuropsychologiques. Ce projet devrait déboucher sur la réalisation d'une solution industrielle, d'aide diagnostique et de remédiation. Nous comptons ainsi proposer un dispositif plus performant en termes d'objectivité et de concision que ce qui existe actuellement.

Il nous est ainsi nécessaire de développer un nouveau matériel bon marché et embarquant, de manière efficace et élégante, les différentes fonctionnalités de matériels déjà sur le marché. Nos besoins n'étant d'ailleurs pas forcément les mêmes que ceux satisfaits par les dits matériels, il nous sera possible d'intégrer des versions « amoindries » de certaines fonctionnalités afin de les assurer à moindre coût. Nous souhaitons aussi anticiper sur les utilisations futures qui pourraient être faites de notre dispositif en le dotant d'une certaine flexibilité et de la possibilité d'évolutions/ajouts de fonctionnalités.

b- Objectif du projet

Le présent projet a pour objectif le **développement d'un prototype de lunettes** permettant le suivi oculaire de l'utilisateur pendant qu'il utilise un système informatique. Concernant le **suivi oculaire**, ce prototype devra pouvoir assurer les mêmes fonctions que celles dont l'interface multimodale devrait disposer à terme. Nous voulons ainsi pouvoir **éprouver** notre choix des capteurs ainsi que le support utilisé pour accomplir ce suivi oculaire.

c- Durée du projet

Nous évaluons la durée du développement d'un prototype fonctionnel entre 1 et 3 mois.

2- Partenaires du projet

a- Présentation et rôles des différents collaborateurs

Nous allons maintenant présenter les différents acteurs du projet et préciser leurs contributions respectives :



Logolexie (N° Siren : 444 849 301 00031)

6, rue Nicolas Joseph Cugnot

63100 CLERMONT-FERRAND

France

<http://www.medialexie.com/>

Présentation :

La société Logolexie/Médialexie est localisée sur le campus universitaire de Clermont-Ferrand (63) au sein de la plate-forme technologique de l'ISIMA. Elle conçoit et développe des applications dans le domaine du Traitement Automatique du Langage Naturel (TALN). Elle intervient en réalisant :

- des logiciels complets pouvant être installés sur les ordinateurs de type PC,
- des modules pour diverses applications (site internet, logiciels dédiés, ...)

Afin d'obtenir des produits aidant, le plus possible, les personnes ayant des troubles de la communication dans leur vie quotidienne, la société Logolexie/Médialexie a passé des accords avec des établissements scolaires et de santé pour connaître les fonctionnalités souhaitées et pour valider la pertinence de ceux-ci.

Toujours dans un souci d'aider le plus de personnes possibles, la société Logolexie /Médialexie réalise des projets en partenariat avec l'AFM, la Fondation France Télécom pour aider des personnes ayant des handicaps spécifiques tels que les myopathes, les aveugles ou les personnes présentant des handicaps moteurs.

Rôle :

Développement d'application d'aide diagnostique et de remédiation

Contact :

Gilles Vessière – gilles.vessiere@medialexie.com – 04 73 25 82 50



LIMOS (CNRS UMR 6158)

Complexe scientifique des Cézeaux

24, avenue des Landais

63173 AUBIERE cedex

France

<http://limos.isima.fr/>

Présentation :

Le LIMOS : Informatique, Modélisation et Optimisation des Systèmes, créé en 1995, en accompagnement de l'ISIMA : Institut d'Informatique, de Modélisation et des Applications, a alors rassemblé 23 enseignants

chercheurs, dont 7 nouvellement recrutés, autour des STIC pour l'organisation. Réparti sur 3 établissements (UBP : porteur, IFMA et UdA : secondaires), reconnu EA en 1996 et en expansion rapide, il s'est alors fixé quelques grands objectifs :

- fédérer les STIC sur le site Clermontois et être partie prenante du développement socio-économique du site ;
- acquérir un rayonnement au plan national et international ;

Afin de parvenir à ces objectifs, il a mis en place une politique de :

- incitation à la publication en revues, soutien aux jeunes vers l'HDR, mutualisation des équipements lourds ;
- ouverture des recrutements vers l'extérieur et ouverture internationale ;
- forte articulation avec les entités de formation professionnelle ;
- équilibre entre fondamental et appliqué et mise en place de liens forts avec le tissu socio-économique local.

Le LIMOS a été associé au CNRS en 2000, relevant successivement des départements SPI, STIC, MIPPU, ST2I et INS2I.

Depuis janvier 2012, le LIMOS a une antenne à l'Ecole Nationale Supérieure des Mines de Saint-Etienne (ENSM.SE, 13 enseignants chercheurs permanents). Maintenant il y a 4 tutelles suivantes : le CNRS, l'Université Blaise Pascal, l'Université d'Auvergne et l'ENSM.SE, et un partenaire institutionnel : IFMA.

Rôle :

Conception et implémentation des traitements logiciels

Contact :

Geoffrey Garcia – garcia@isima.fr – 07 60 57 76 35



Méc@Prod / IFMA

Complexe scientifique des Cézeaux

24, avenue des Landais

BP 265 - 63175 AUBIERE Cedex

France

www.ifma.fr/mecaprod

Présentation :

A destination des industriels, Méc@Prod mobilise des compétences pointues et des moyens performants pour réaliser des actions de transfert de technologie. Son savoir-faire et son matériel de haute technologie lui garantissent notamment dans les domaines de l'Usinage Grande Vitesse (UGV) et du décolletage, des compétences exclusives, très peu développées au niveau régional.

Notre volonté est de poursuivre notre démarche de mutualisation de moyens performants en élargissant nos domaines d'intervention pour la réalisation de prestations industrielles et formation continue de qualité.

Rôle :

Conception et prototypage mécanique

Contact :

José Alba – jose.alba@ifma.fr – 06 13 52 67 81



CAP'TRONIC / JESSICA FRANCE

12, rue Gémini
87068 LIMOGES Cedex
France

<http://www.captronic.fr/>

Présentation :

L'Association JESSICA FRANCE, fondée par le CEA et OSEO, est principalement financée par le Ministère de l'Economie, des Finances et de l'Industrie.

Elle porte le programme CAP'TRONIC, dont la mission est de faciliter l'innovation et la compétitivité des PME par l'électronique.

Forte de vingt Ingénieurs répartis sur le territoire et de plus de quatre cents adhérents, elle a aidé en 2009 plus de 1800 PME différentes de tous secteurs par de la sensibilisation au moyen de séminaires techniques (1100 PME), des conseils (494 PME), des contrats d'appui technique en collaboration avec des centres de compétence en électronique (260 PME) ainsi que le suivi des PME et de leurs projets (364 PME).

Rôle :

Expertise, conseil et mise en relation

Contact :

Dominique Gérard – gerard@captronic.fr – 06 60 77 06 70

b- Autres modalités de collaboration

La réalisation de ce projet fera l'objet d'un contrat de collaboration entre les différents partenaires. L'élaboration de ce contrat sera, par ailleurs, supervisée par la SATT Grand Centre.

II- Expression des besoins fonctionnels

Le besoin fondamental auquel doit répondre le dispositif, est représenté sur la figure ci-dessous :

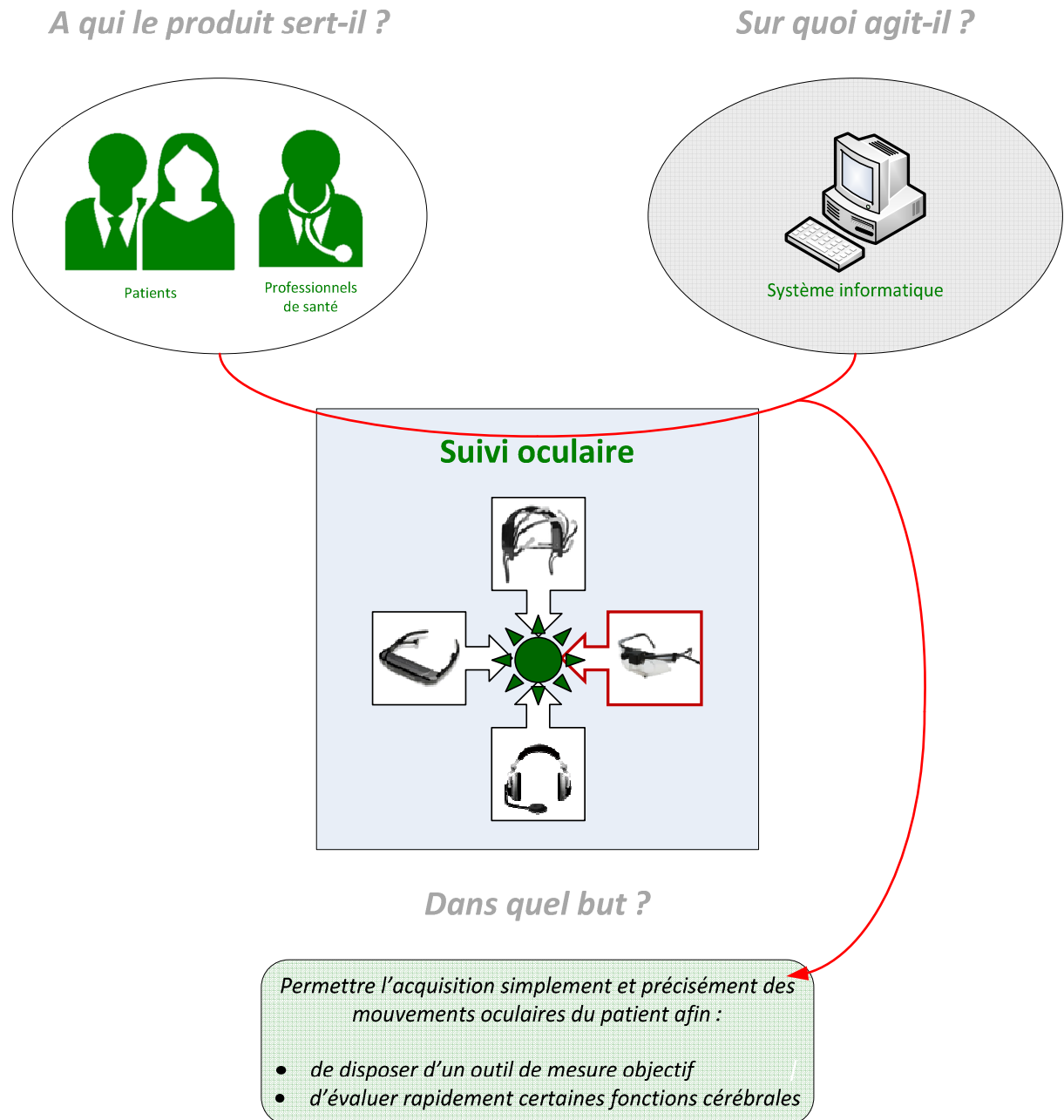


Figure 1 : Représentation fonctionnelle du besoin fondamental

L'interface multimodale sera composée d'un support de type « casque », de capteurs ainsi que de l'électronique nécessaire à l'intégration des mesures et de leur acheminement jusqu'à un système informatique. Il nous est d'ailleurs possible de clairement définir les acteurs et fonctions nécessaires à la satisfaction du besoin :

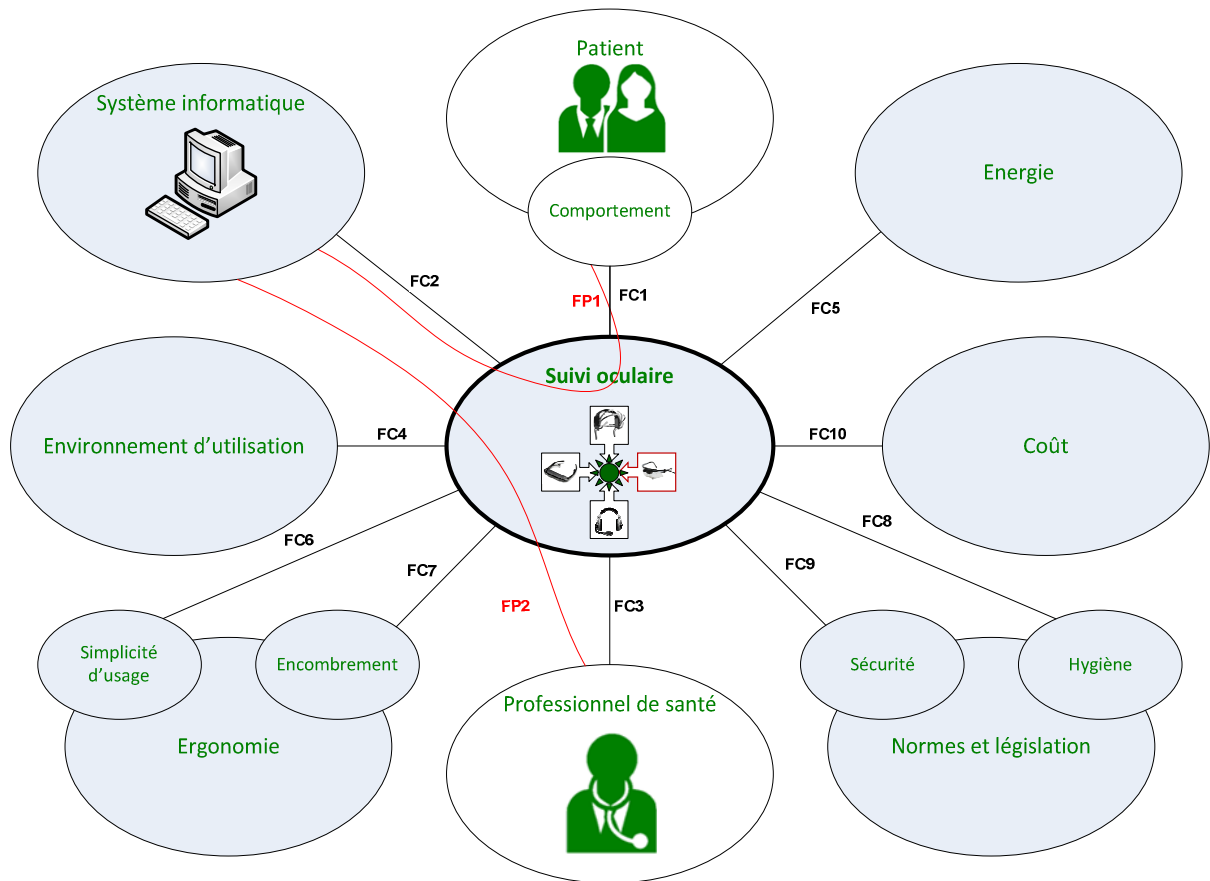


Figure 2 : Diagramme des interacteurs

Voici le détail des fonctions présentes sur ce diagramme :

Fonctions principales	Fonctions complémentaires ou contraintes
FP1 : Assurer l'acquisition sur le patient de données permettant son suivi oculaire	FC1 : S'adapter à la physionomie et la physiologie de l'utilisateur
FP2 : Permettre le pilotage du dispositif par un professionnel de santé	FC2 : Garantir une connectabilité avec le système informatique
	FC3 : Intégrer et couvrir les pratiques et processus de soins utilisés par le professionnel, en garantissant la qualité des résultats proposés
	FC4 : Fonctionner dans un environnement remplissant des conditions adéquates d'usage
	FC5 : Assurer une alimentation en électricité
	FC6 : Etre utilisable très simplement
	FC7 : Entraver le moins possible le porteur
	FC8 : Respecter les exigences concernant les appareils utilisés pour les examens médicaux
	FC9 : Respecter la réglementation à vigueur pour ce genre d'appareil électronique
	FC10 : Etre bon marché autant que possible

1- Les caméras embarquées

D'une manière générale chaque caméra, utilisée pour faire le suivi oculaire, devrait pouvoir fournir 30 images par seconde pour garantir la réalisation de **FP1**. D'autre part, conformément à **FC3**, l'ensemble des caméras embarquées doit être synchronisé de façon à ce que toutes les acquisitions d'image se fassent simultanément. Les caractéristiques de l'œil, utiles pour le suivi oculaire, doivent être visibles quelque soit l'utilisateur, comme précisé par **FC1**.

Ainsi nous avons décidé de réutiliser un matériel, répondant à nos exigences, qui a déjà été utilisé avec succès d'un projet similaire au notre (Projet Pupil : <http://code.google.com/p/pupil/>). Pour ce qui est de la synchronisation nous la ferons dans un premier temps à un niveau logiciel et nous avons choisi d'embarquer quatre caméras aux spécifications identiques afin de nous simplifier d'autant la tâche.

Ci-dessus sont reportées les informations relatives aux caméras que nous allons intégrer :

Référence	Description	Quantité
Microsoft HD-6000	http://www.microsoft.com/hardware/fr-fr/p/lifecam-hd-6000-for-notebooks#details	4

2- Support mécanique

a- *Forme générale et taille*

D'une manière globale, notre matériel devra respecter l'ensemble des contraintes précédemment citées :

- conformément à **FC1** : le support doit s'adapter aux différentes tailles et formes de têtes
- conformément à **FC3** : si le support comprend des marques ou graduations, elles devront se conformer aux nomenclatures utilisées par les professionnels de santé
- conformément à **FC4** : le casque doit être adapté aux conditions thermiques de bon fonctionnement de ses composants, ainsi que proposer une isolation électromagnétique pour ceux qui le requièrent
- conformément à **FC6** : les connectiques ainsi que les différentes articulations, permettant de modifier les positions d'éléments du casque, doivent être d'un usage trivial ou approchant
- conformément à **FC7** : le poids et le volume du dispositif devront être minimisés afin d'interférer le moins possible avec le comportement de l'utilisateur (en trouvant un compromis avec **FC10**)
- conformément à **FC8** : le casque doit présenter le moins de recoins et d'aspérités possibles, et d'autre part les parties exposées de ses matériaux ne doivent pas craindre les produits d'hygiène/entretien
- conformément à **FC9** : le produit doit répondre aux normes en vigueur concernant les interfaces homme-machine en informatiques
- conformément à **FC10** : en cas de possibilités multiples concernant le choix des matériaux ou de l'apparence, il faudra veiller à choisir les solutions les moins onéreuses

b- Connectique

Conformément à **FC2**, il comporter un passage de câble et proposer une connectique de type USB2 pour chaque caméra embarquée.

c- Intégration des caméras

Conformément à **FP1**, notre système doit permettre de déterminer ce que regarde le porteur du casque. Pour cela deux caméras seront braquées sur chaque œil de l'utilisateur. Les positions de ces caméras devront pouvoir être ajustées précisément (voir [Figure 3](#)) afin de s'assurer que la pupille de l'utilisateur est correctement filmée.

Ces dernières vont permettre, pour chaque œil, de déterminer un vecteur de regard de l'utilisateur comme schématisé sur l'[Annexe 1](#).

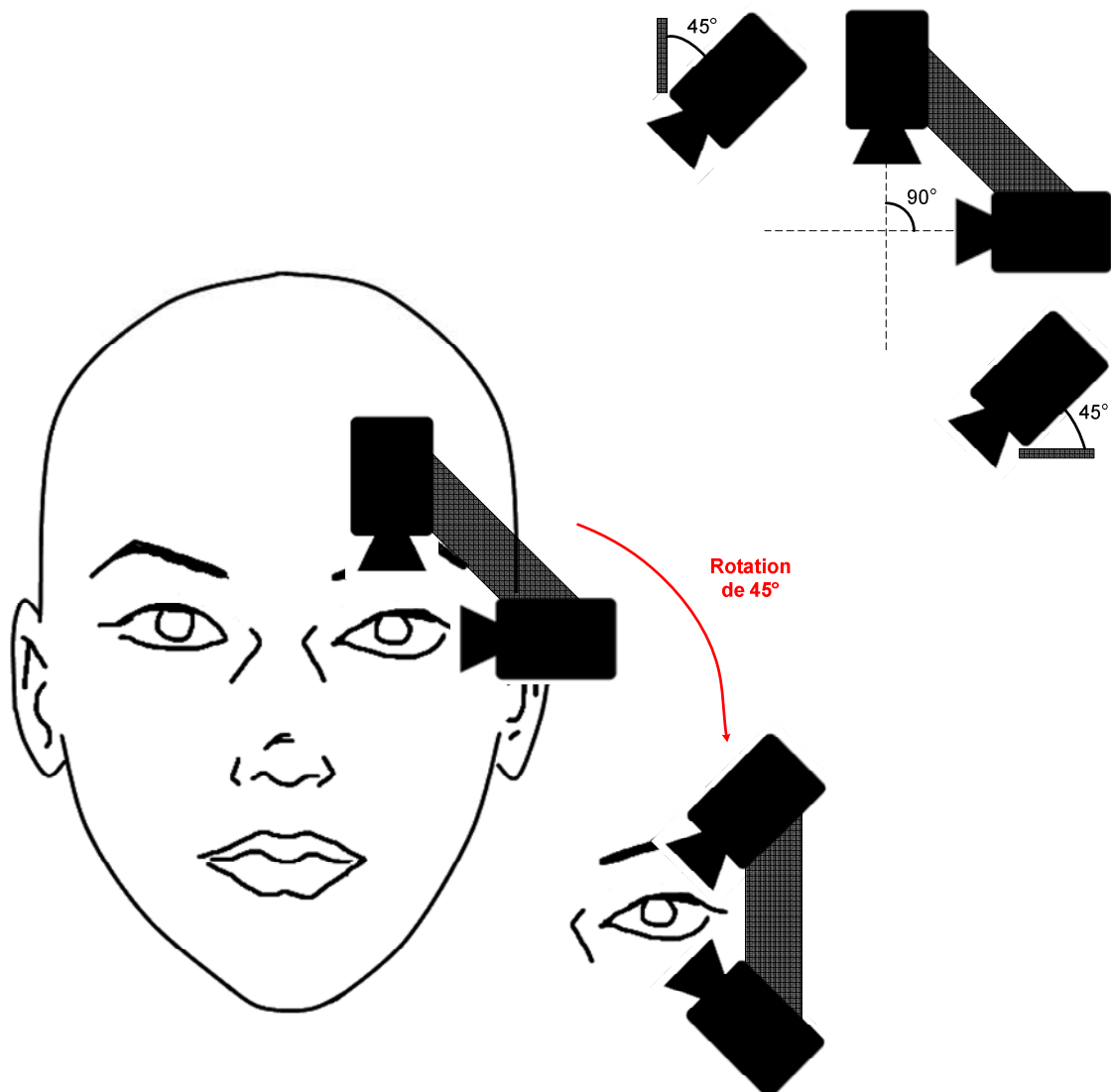


Figure 3 : Positions des caméras à proximité de l'œil

III- Annexes

1- Détermination du vecteur de regard de l'utilisateur

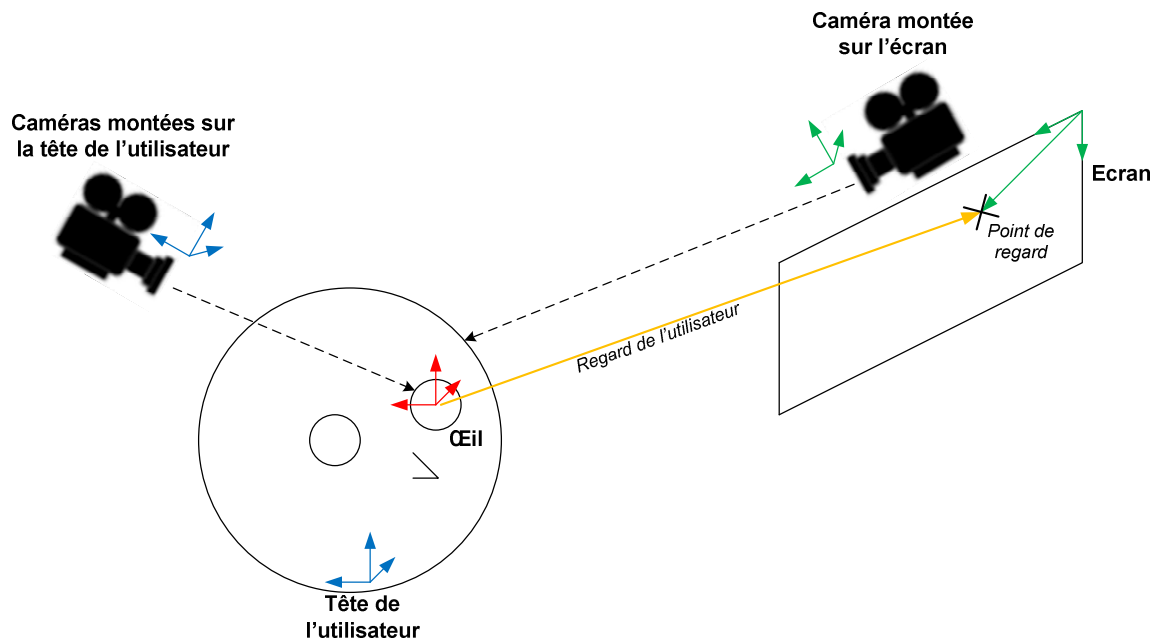


Figure 4 : Schéma général du dispositif de suivi oculaire et des repères manipulés

On peut préciser que les repères de même couleur (**rouge**, **bleu** et **vert**) sont liés. C'est-à-dire que leurs positions relatives sont constantes et, par la même occasion, seront connues consécutivement au paramétrage du système. Nous pouvons ainsi nous limiter à l'utilisation de 3 repères : le repère de l'œil de l'utilisateur (**rouge**), celui de la tête de ce dernier (**bleu**) et pour finir, celui de l'écran (**vert**).

2- Contexte applicatif de l'utilisation de l'interface multimodale

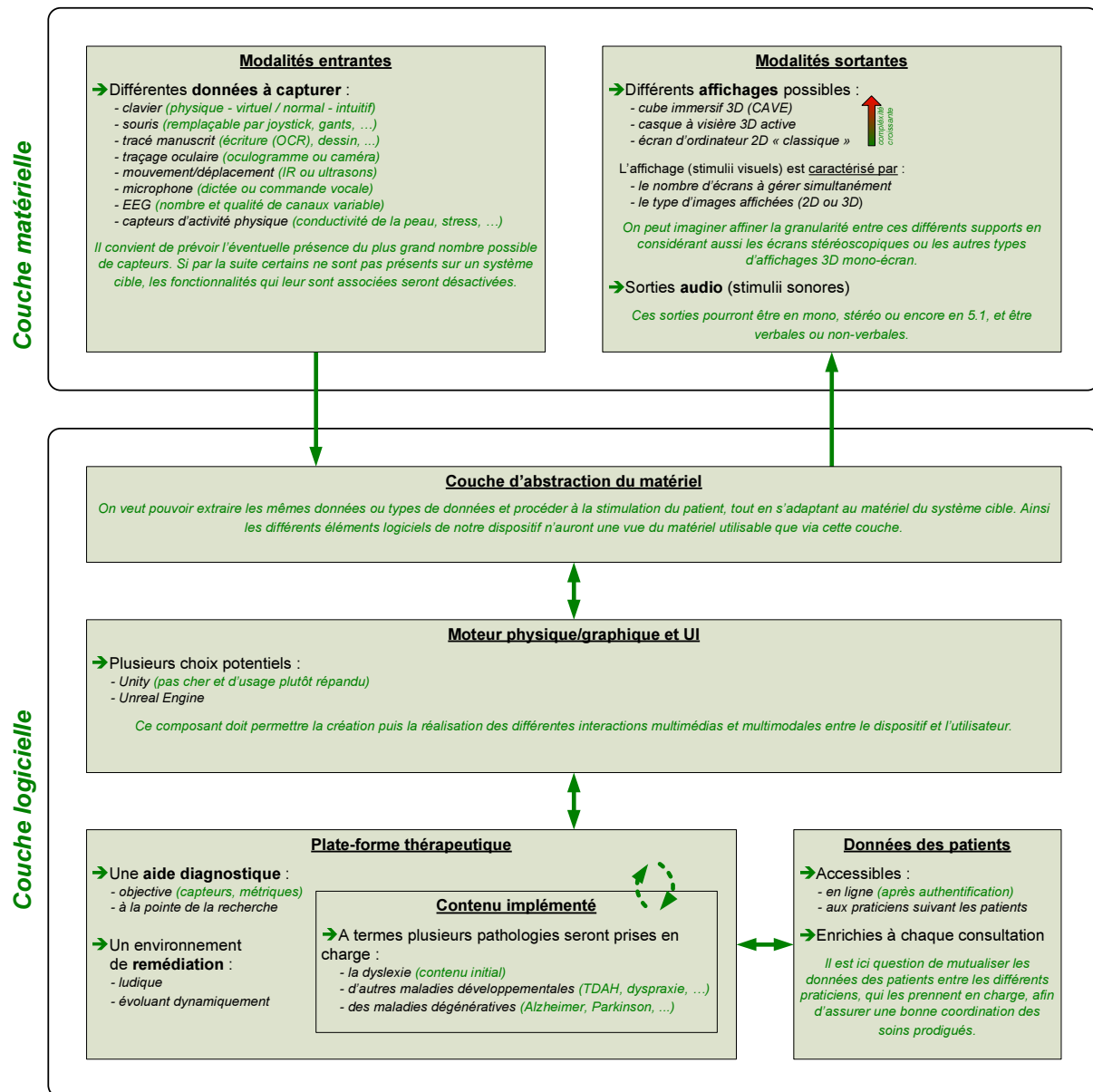


Figure 5 : Architecture matérielle et logicielle de notre projet

Sur ce schéma sont représentés les différents éléments, logiciels et matériels, impliqués dans notre projet d'application. Les différentes fonctionnalités couvertes par l'interface multimodale appartiennent à la couche logicielle.

Au niveau logiciel, la figure ci-dessous représente les différents agents impliqués dans les traitements des données manipulées par l'interface multimodale :

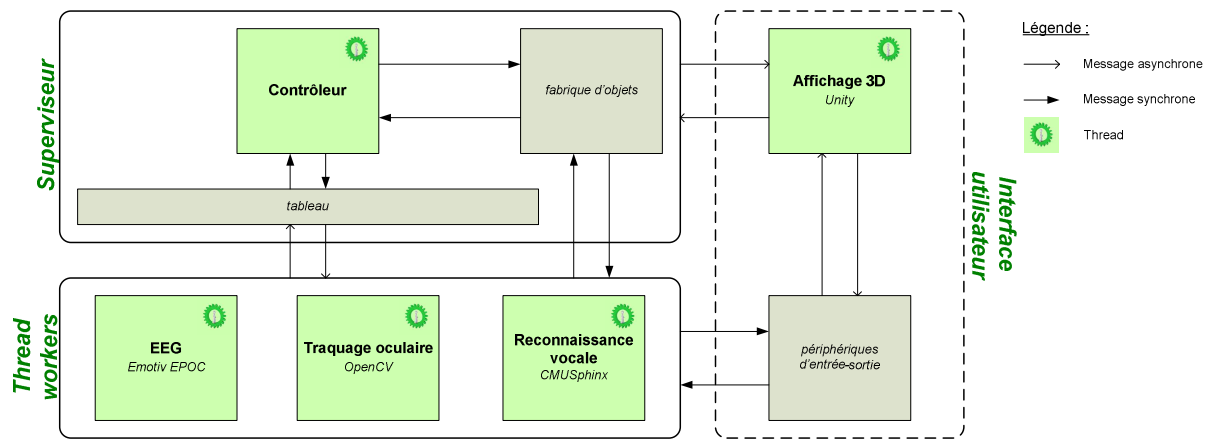


Figure 6 : Briques logicielles du modèle multi-agents

B) Premiers résultats

Bien que le casque de suivi oculaire n'ait pas encore pu être réellement testé, ci-dessous se trouvent quelques images issues de son prototypage. On signalera que les deux photographies illustrent le résultat d'une impression 3D du casque.

