



HAL
open science

Modèle d'organisation multi-agent pour l'aide au travail coopératif dans les processus d'entreprise : application aux systèmes administratifs complexesbeta

Emmanuel Adam

► To cite this version:

Emmanuel Adam. Modèle d'organisation multi-agent pour l'aide au travail coopératif dans les processus d'entreprise : application aux systèmes administratifs complexesbeta. Autre [cs.OH]. Université de Valenciennes et du Hainaut-Cambresis, 2000. Français. NNT : . tel-00007961v1

HAL Id: tel-00007961

<https://theses.hal.science/tel-00007961v1>

Submitted on 7 Jan 2005 (v1), last revised 4 May 2005 (v2)

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 DE VALENCIENNES ET DU HAINAUT-CAMBRÉSIS

N° attribué par la bibliothèque
0016

THÈSE

pour obtenir le grade de

**DOCTEUR DE
L'UNIVERSITÉ DE VALENCIENNES ET DU HAINAUT-CAMBRÉSIS**

Discipline : Informatique

présentée et soutenue publiquement

par

Emmanuel ADAM

28 Septembre 2000

Titre :

**Modèle d'organisation multi-agent
pour l'aide au travail coopératif dans les processus d'entreprise :
application aux systèmes administratifs complexes**

JURY

M. Jean-Paul BARTHES	Université Technologique de Compiègne	Rapporteur
M. Christophe KOLSKI	Université de Valenciennes	Directeur
M. René MANDIAU	Université de Valenciennes	Co-Directeur
M. Philippe MATHIEU	Université de Lille I	Examineur
M. Joël QUINQUETON	Université de Montpellier II	Rapporteur
M. Frédéric SEMET	Université de Valenciennes	Président
M. Emmanuel VERGISON	SOLVAY S.A.	Examineur

"Les liens entre individus d'une société animale, comme ceux des cellules d'un organisme, ne sont pas seulement de communication selon un code commun, ils comportent aussi une dimension communautaire, c'est-à-dire d'inclusion des êtres dans un circuit trans-subjectif."

Edgar Morin, La Méthode 2. La Vie de la Vie.

*A Lisa,
A mes parents,
A ma famille,
A mes amis.*

Avant propos

Le travail présenté dans ce mémoire a été réalisé au sein du département de la Propriété Industrielle de l'entreprise SOLVAY S.A. situé à Bruxelles, dans le cadre d'une bourse co-financée par la région Nord – Pas de Calais et SOLVAY S.A., en collaboration avec le Laboratoire d'Automatique et de Mécanique Industrielles et Humaines de l'Université de Valenciennes et du Hainaut-Cambrésis, et plus précisément dans l'équipe Raisonnement Automatique et Interaction Homme-Machine dirigée par le Professeur Christophe KOLSKI.

J'adresse un grand merci à monsieur Emmanuel VERGISON, responsable du Knowledge Management au sein de la Direction Centrale des Recherches et Technologies de SOLVAY, qui m'a guidé tout au long de cette thèse réalisée en entreprise.

Je suis très reconnaissant envers Messieurs Jean-Paul BARTHES, Professeur à l'Université de Technologie de Compiègne et Joël QUINQUETON, Professeur à l'Université Paul Valéry de Montpellier, pour m'avoir fait l'honneur d'être les rapporteurs de ce mémoire. Je suis également très reconnaissant envers Messieurs Philippe MATHIEU, Professeur à l'Université de Lille I et Frédéric SEMET, Professeur à l'Université de Valenciennes et du Hainaut-Cambrésis, pour avoir accepté d'examiner cette thèse.

Je remercie également monsieur le professeur Christophe KOLSKI, ainsi que monsieur René MANDIAU, maître de conférences, tous deux de l'Université de Valenciennes et du Hainaut-Cambrésis, pour leur patience, leurs encouragements et leurs nombreux avis pertinents qui m'ont permis de mener à terme cette thèse.

Je tiens à remercier les membres de SOLVAY N.O.H., et plus précisément du département de Propriété Industrielle avec qui j'ai eu le plaisir de travailler, de m'avoir accueilli et d'avoir participé à la réalisation de ce travail.

Je remercie également les chercheurs, ingénieurs et administratifs du Laboratoire d'Automatique et de Mécanique Industrielles et Humaines de l'Université de Valenciennes et du Hainaut-Cambrésis pour leur soutien, leurs conseils et leur amitié.

Je remercie spécialement tous ceux qui ont croisé pour quelques temps ma route au sein de SOLVAY, je pense notamment à l'équipe des stagiaires qui ont animé ces années de travail.

Enfin, je remercie bien sûr mon épouse, pour sa grande patience, son aide et ses encouragements. Et je remercie naturellement mes parents, qui m'ont donné les moyens de poursuivre mes études. Je leur dois à tous beaucoup et leur exprime ma plus profonde reconnaissance.

TABLE DES MATIERES

INTRODUCTION GENERALE.....	13
CHAPITRE I MODÈLES D'ORGANISATIONS MULTI-AGENTS POUR LES SYSTÈMES D'AIDE AU TRAVAIL COOPÉRATIF EN ENTREPRISE.....	
INTRODUCTION.....	16
I.1 CARACTÉRISTIQUES DES SYSTÈMES D'INFORMATION DISTRIBUÉS.....	18
I.1.1 L'architecture d'un réseau coopératif.....	18
I.1.2 La coopération dans un réseau d'entreprise.....	21
I.1.3 Architecture d'organisations du travail.....	23
I.1.4 Conclusion sur la structure des systèmes d'information distribués.....	27
I.2 MODÈLES D'ORGANISATIONS D'AGENTS LOGICIELS EN INTELLIGENCE ARTIFICIELLE DISTRIBUÉE.....	27
I.2.1 Définition des agents logiciels.....	28
I.2.2 Typologie des agents logiciels.....	31
I.2.3 Les modèles d'organisation.....	35
I.3 UN MODÈLE UNIFICATEUR : LES SYSTÈMES HOLONIQUES.....	38
I.3.1 Définition informelle des systèmes multi-agents holoniques.....	40
I.3.1.1 Règles relatives à l'effet Janus.....	40
I.3.1.2 Règles relatives à la structure arborescente.....	42
I.3.1.3 Règles relatives aux mécanismes des systèmes holoniques.....	43
I.3.1.4 Règles relatives aux communications.....	44
I.3.1.5 Règles relatives à l'adaptation à l'environnement.....	46
I.3.2 Une définition formelle des systèmes multi-agents holoniques.....	47
CONCLUSION.....	52
BIBLIOGRAPHIE DU CHAPITRE I.....	54
CHAPITRE II ETUDE COMPARATIVE DE METHODES DU GENIE LOGICIEL DANS UN BUT DE MODELISATION DES ORGANISATIONS MULTI-AGENTS.....	
INTRODUCTION.....	62
II.1 BESOINS EN ANALYSE ET MODELISATION D'ORGANISATIONS MULTI-AGENTS DEDIEES AUX SYSTEMES D'INFORMATIONS DISTRIBUES.....	63
II.2 CHOIX D'UN CADRE DE COMPARAISON DE METHODES.....	65
II.2.1 La dimension représentation.....	65
II.2.2 La dimension environnement d'application.....	67
II.2.3 La dimension méthodologie.....	68
II.2.4 La dimension coopération.....	69
II.2.5 La dimension technologie.....	71
II.3 UTILISATION DU CADRE DE COMPARAISON POUR L'EVALUATION DE SEPT METHODES REPRESENTATIVES.....	72
II.3.1 La méthode MERISE.....	73
II.3.2 La méthode OMT.....	75
II.3.3 Les extensions d'UML.....	77
II.3.4 La méthode SADT.....	78
II.3.5 La méthode OSSAD.....	79
II.3.6 La méthode CISAD.....	81
II.3.7 La méthode MKSM.....	82
II.3.8 Synthèse et discussion.....	84
CONCLUSION.....	86
BIBLIOGRAPHIE DU CHAPITRE II.....	88

CHAPITRE III MAMOSACO : UNE MÉTHODE ADAPTABLE DE MODÉLISATION DE SYSTÈMES ADMINISTRATIFS COMPLEXES	91
---	-----------

INTRODUCTION	92
III.1 BESOINS EN ANALYSE ET MODÉLISATION POUR UNE NOUVELLE MÉTHODE	94
III.1.1 Besoins selon les cinq dimensions.....	94
III.1.2 Besoins vis-à-vis de l'analyse.....	97
III.2 ÉLÉMENTS DE MODÉLISATION DANS LA MÉTHODE MAMOSACO	99
III.2.1 Modèle des données.....	99
III.2.2 Modèle des flux de données.....	100
III.2.3 Modèle des traitements de données.....	101
III.3 MODÈLE DYNAMIQUE ET SIMULATION	102
III.3.1 Choix des réseaux de Petri paramétrés.....	103
III.3.1.1 Définition des réseaux de Petri paramétrés.....	105
III.3.1.2 Exemple d'utilisation de réseaux de Petri paramétrés.....	107
III.3.2 Application des réseaux de Petri paramétrés à la représentation des interruptions.....	110
III.3.3 La simulation de procédures administratives complexes.....	114
III.4 LES DIFFÉRENTES PHASES DE LA MÉTHODE MAMOSACO	115
III.5 SPÉCIFICATION DE SYSTÈME ORIENTÉ HOLON POUR L'AIDE AU TRAVAIL COOPÉRATIF (SOHTCO)	118
III.5.1 Règles d'assistance.....	118
III.5.2 Structure de l'organisation.....	120
III.5.3 Modélisation et Spécification du SOHTCO.....	123
III.5.3.1 Description individuelle des agents holoniques.....	123
III.5.3.2 Fonctionnement coopératif interne du SOHTCO.....	126
III.6 MODÈLE NABLA ET PRINCIPE DE RÉUTILISABILITÉ DANS LA MÉTHODE MAMOSACO	130
CONCLUSION	133
BIBLIOGRAPHIE DU CHAPITRE III	134

CHAPITRE IV CONCEPTION D'UN ATELIER DE GÉNIE LOGICIEL SUPPORTANT LA MÉTHODE MAMOSACO	137
---	------------

INTRODUCTION	138
IV.1 PRÉSENTATION GLOBALE DE L'ATELIER MAMOSACO	139
IV.1.1 Rappel sur les ateliers de génie logiciel.....	139
IV.1.2 Architecture de l'atelier MAMOSACO.....	141
IV.2 FONCTIONNALITÉS DE MODÉLISATION POUR MAMOSACO	141
IV.2.1 Conception de l'atelier.....	143
IV.2.2 Utilisation des fonctionnalités de modélisation.....	146
IV.2.2.1 Fonctionnalités associées à la matrice Activité-Rôle.....	146
IV.2.2.2 Fonctionnalités associées au modèle des données.....	147
IV.2.2.3 Fonctionnalités associées au modèle des activités.....	149
IV.2.2.4 Fonctionnalités associées au modèle des traitements.....	150
IV.2.2.5 Fonctionnalités associées au modèle dynamique.....	151
IV.2.3 Autre fonctionnalité disponible dans l'atelier MAMOSACO.....	152
IV.2.4 Conclusion sur les fonctionnalités de modélisation.....	153
IV.3 FONCTIONNALITÉS DE SIMULATION D'ORGANISATION ADMINISTRATIVE COMPLEXE	153
IV.3.1 Conception du simulateur SIMOH.....	154
IV.3.2 Utilisation des fonctionnalités de simulation.....	156
IV.3.3 Conclusion sur les fonctionnalités de simulation.....	160
CONCLUSION	160
BIBLIOGRAPHIE DU CHAPITRE IV	161

CHAPITRE V APPLICATION DE LA MÉTHODE MAMOSACO À DES PROCESSUS ADMINISTRATIFS COMPLEXES ET PERSPECTIVES DE RECHERCHE	163
INTRODUCTION	164
V.1 PRÉSENTATION GÉNÉRALE DU CAS D'APPLICATION	165
V.2 DESCRIPTION DE L'UTILISATION DE MAMOSACO SUR UNE PROCÉDURE DE DÉPÔT DE BREVET	166
V.2.1 Analyse du fonctionnement de la procédure	167
V.2.2 Modélisation de la procédure de dépôt de brevet	168
V.2.2.1 Matrice Activités-Rôle de la procédure de dépôt de brevet	169
V.2.2.2 Modèle des données de la procédure de dépôt de brevet	170
V.2.2.3 Modèle des activités et simplifications de la procédure dépôt de brevet	171
V.2.2.4 Modèle de traitements de la procédure de dépôt de brevet	174
V.2.2.5 Modèle dynamique de la procédure de dépôt de brevet	176
V.2.3 Simulation de la procédure de dépôt de brevet prioritaire	179
V.2.4 Spécification du SOHTCO	182
V.2.5 Eléments de développement du SOHTCO	185
V.3 ANALYSE DES DONNÉES RECUEILLIES ET DISCUSSION	187
V.4 PERSPECTIVES DE RECHERCHE	191
V.4.1 Perspectives associées à MAMOSACO	191
V.4.2 Perspectives associées aux SOHTCO	192
CONCLUSION	195
BIBLIOGRAPHIE DU CHAPITRE V	197
CONCLUSION GENERALE	199
RÉFÉRENCES BIBLIOGRAPHIQUES	203
INDEX DES FIGURES ET DES TABLEAUX	215
ANNEXE A. LE MODELE V	221
ANNEXE B. EXERCICE DE MODELISATION	225
ANNEXE C. MODELISATION DE LA PROCEDURE DE DEPOT DE BREVET	241

Glossaire

G.E.D. : *Gestion Electronique des Documents.*

I.A. : *Intelligence Artificielle.*

I.A.D. : *Intelligence Artificielle Distribuée.*

MAMOSACO : *Méthode Adaptable de MOdélisation de Systèmes Administratifs COmplexes.*

OMA : *Organisation Multi-Agent.*

S.I.D. : *Système d'Information Distribué.*

SIMOH : *SIMulateur d'Organisation Humaine.*

SMA : *Système Multi-Agents.*

SOHTCO : *Système Orienté Holon pour l'aide au Travail COopératif.*

T.C.A.O. : *Travail Coopératif Assisté par Ordinateur.*

Introduction générale

L'explosion de la technologie Internet et des réseaux d'entreprise a contribué à bouleverser bon nombre d'habitudes établies dans les entreprises depuis plusieurs décennies. Dans un contexte de globalisation de l'économie et de profondes modifications des structures socio-économiques, les processus techniques et administratifs qui sous-tendent les activités de l'Entreprise font, en particulier, l'objet de révisions profondes. Les documents papier échangés de mains en mains, laissent progressivement place aux documents électroniques transmis automatiquement par des machines sans prendre en considération les facteurs humains tels que les notions de groupe (les individus sont isolés face à leurs poste de travail), les niveaux de responsabilité ou même la coopération homme-machine. Certes, des outils d'aide au travail coopératif ont déjà été proposés, avec succès pour certains, mais ils ne "s'attaquent" pas à l'ensemble d'une organisation.

L'objectif de ce mémoire est de proposer des principes d'aide à la gestion commune d'informations au sein d'une organisation de type administrative complexe. Nous nous intéressons aux organisations possédant une structure pyramidale et mettant en jeu un nombre important d'acteurs qui ont des rôles différents et des degrés de responsabilité propres. Dans un système administratif complexe, les activités consistent essentiellement en la création, la gestion et la distribution d'informations effectuées par l'ensemble des acteurs..

L'aide à la gestion d'information doit être adaptée, distribuée à chaque élément de l'organisation. Ces notions d'organisation et de système distribué se situent au cœur même de l'Intelligence Artificielle Distribuée (I.A.D.), plus précisément dans le domaine des organisations multi-agents. L'analyse et la conception d'un outil d'aide au travail coopératif, reposant sur une organisation multi-agent, font donc particulièrement l'objet de ce mémoire.

Le premier chapitre a pour objectif la recherche d'une structure d'organisation permettant la spécification d'un système d'aide au travail coopératif dans les processus de type administratif complexe. Il définit tout d'abord la structure générale des systèmes d'information distribués puis établit une comparaison entre les différents types d'organisation existant en entreprise d'une part, dans le domaine de l'Intelligence Artificielle Distribuée d'autre part. A l'issue de cette comparaison, le modèle holonique, modèle organisationnel applicable aux organisations humaines ou agents à structure pyramidale, est proposé.

Le deuxième chapitre a pour but la recherche d'une méthode permettant de modéliser les organisations de type holonique tels que les systèmes administratifs complexes. Cette recherche s'effectue par confrontation de différentes méthodes existantes sur un cadre de comparaison jouant le rôle de benchmark. De cette comparaison, il est apparu qu'aucune méthode, prise individuellement, n'est en mesure de répondre à la question. Par contre, il est possible d'intégrer plusieurs méthodes testées et de construire ainsi une nouvelle méthode plus adéquate, plus adaptée à la problématique faisant l'objet de cette thèse.

Le troisième chapitre décline la méthode d'analyse et de modélisation MAMOSACO (Méthode Adaptable de Modélisation de Systèmes Administratifs Complexes) s'articulant selon quatre points de vue (utilisant un modèle de données, d'activités, de traitements et un modèle dynamique). Cette méthode intègre également une phase de simulation. Celle-ci s'avère importante pour la compréhension des mécanismes de l'organisation étudiée ainsi que pour l'obtention de ses règles de fonctionnement. Ces règles sont alors intégrées dans un Système Orienté Holon facilitant le Travail COopératif (SOHTCO) d'un ensemble d'acteurs impliqués dans une procédure administrative complexe. La spécification générale d'un SOHTCO est également proposée dans ce chapitre.

Le quatrième chapitre décrit les différents outils logiciels mis en place pour faciliter les étapes de modélisation, de simulation et de spécification. Ces outils forment l'atelier de modélisation et de simulation de la méthode MAMOSACO.

Le cinquième chapitre expose une application de la méthode à un cas réel de système administratif complexe. Les étapes d'analyse, de modélisation et de simulation sont détaillées au travers d'une application de la méthode MAMOSACO à une procédure précise d'un département de SOLVAY S.A. Une spécification d'un SOHTCO adapté à cette procédure est également décrite. Les premiers retours d'expériences issus de l'application de la méthode MAMOSACO et de la conception de l'organisation multi-agent holonique sur un cas réel, sont présentés en fin de ce chapitre. Les réutilisations possibles des différentes parties de ce travail y sont également exposées. Enfin, ce chapitre décrit nos perspectives de recherche.

Chapitre I -

Modèles d'Organisations Multi-Agents pour les

Systèmes d'aide au travail coopératif en entreprise

INTRODUCTION	16
I.1 CARACTERISTIQUES DES SYSTEMES D'INFORMATION DISTRIBUES.....	18
I.2 MODELES D'ORGANISATIONS D'AGENTS LOGICIELS EN INTELLIGENCE ARTIFICIELLE DISTRIBUEE.....	27
I.3 UN MODELE UNIFICATEUR : LES SYSTEMES HOLONIQUES	38
CONCLUSION	52
BIBLIOGRAPHIE DU CHAPITRE I.....	54

Introduction

La notion d'organisation a toujours été dépendante de la notion de tâche ou de celle d'échange d'informations. Or, cette dernière décennie a connu l'émergence et l'utilisation généralisée de la messagerie électronique et surtout la mise en place de plus en plus systématique d'outils d'aide au travail coopératif. Ces outils sont soit de type workflow [Marshak 95], c'est-à-dire destinés à la gestion de flux de données (le terme GED pour Gestion Electronique des Documents est également utilisé), soit de type groupware, plus orienté vers le travail de groupe sur un même ensemble de documents [Schael 97]. Ces outils, issus du domaine du Travail Coopératif Assisté par Ordinateur¹ (TCAO) [Wilson 94] contribuent à la restructuration des organisations traditionnellement centralisées, au profit d'organisations distribuées dont les acteurs sont physiquement éloignés (notion d'entreprise distribuée ou virtuelle).

Cette notion d'organisation d'acteurs distribués est très étudiée dans le domaine de l'Intelligence Artificielle (IA) et plus particulièrement dans celui de l'Intelligence Artificielle Distribuée (IAD) s'intéressant particulièrement à la modélisation du comportement d'entités intelligentes distribuées dans un environnement. Les organisations multi-agent proposées dans ce domaine ne contraignent pas leurs agents à être présents dans un même lieu. Se communiquant des informations sur des sites physiquement distribués, les organisations multi-agents peuvent servir de support à la réflexion pour la conception de systèmes d'informations distribués [Yu 99].

Ce chapitre vise à proposer, suite à une réflexion sur les modèles organisationnels, une architecture générique pour la mise en place d'outils d'aide à la coopération, principalement dans les systèmes d'informations distribués.

Cette démarche est apparue nécessaire. En effet, si de nombreux logiciels d'aide au travail coopératif existent, leurs concepteurs oublient trop souvent qu'ils s'adressent à des agents ayant des rôles particuliers et les considèrent dans de nombreux cas comme des réservoirs ou fournisseurs de données. Les solutions de type groupware sont beaucoup plus évoluées, mais sont par nature locales ou focalisées sur une activité particulière (par exemple : la vidéo conférence). Il y a donc un manque d'approche globale et sociale des organisations humaines en ce qui concerne la mise en place d'une gestion large de l'information.

¹ La TCAO a pour traduction CSCW (Computer Supported Cooperative Work) [Bannon 91].

La première partie de ce chapitre présente les caractéristiques générales des systèmes d'information distribués. Dans la deuxième partie, sera examiné un ensemble de modèles d'organisations humaines, généralement utilisés de nos jours dans les entreprises, ainsi que leur prédisposition à la mise en place d'un Système d'Information Distribué (S.I.D.). La troisième partie aborde l'organisation du point de vue des systèmes multi-agent et montre la similitude de leur structure avec celle des SID. La quatrième partie propose un modèle particulier d'organisation multi-agent, le modèle sociologique holonique [Koestler 69]. Ce modèle est présenté dans cette partie sous le point de vue multi-agent et est considéré comme une forme particulière d'Organisation Multi-Agent (OMA). Plus généralement, ce modèle est applicable aux sociétés (humaines ou agents [Mandiau 99]) de type hiérarchie flexible, c'est-à-dire possédant une structure pyramidale, mais dont les membres possèdent une relative autonomie.

I.1 Caractéristiques des systèmes d'information distribués

Cette partie propose de définir la structure générale des systèmes d'information distribués qui existent dans les entreprises. Celle-ci est également appelée réseau coopératif d'entreprise [Schael 97]. Les définitions proposées sont issues d'une synthèse des différentes propositions fournies dans les domaines du groupware et du workflow.

Premièrement, la structure générale d'un réseau d'entreprise sera définie. Ensuite seront présentées les principales composantes de la coopération entre les acteurs des réseaux coopératifs d'entreprise. Finalement les architectures des organisations du travail qui ont cours aujourd'hui dans les entreprises seront étudiées en fonction de leur adéquation avec la mise en place de systèmes d'information distribués. Ceci afin de proposer, au final, une structure réellement applicable en entreprise.

I.1.1 L'architecture d'un réseau coopératif

L'architecture des réseaux d'entreprise peut être décrite sous la forme de graphes [Berge 70] dont les sommets transforment les données qui circulent le long des arcs afin de réaliser un objectif général qui peut être décomposé en sous-objectifs (par exemple : la procédure permettant le dépôt de brevet nécessite de vérifier l'absence d'un brevet similaire, avant de procéder aux démarches administratives). Les connexions entre les sommets dépendent de l'organisation humaine du système, structurée selon les rôles ou selon les fonctions des acteurs. La figure I.1 schématise cette représentation des réseaux d'entreprise, sur un exemple.

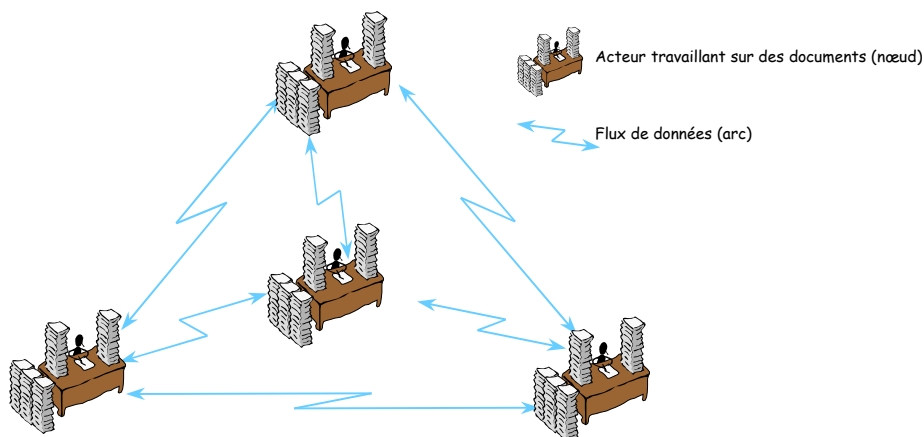


Figure I.1. Un exemple de réseau coopératif

Entre les sommets, le long des arcs circulent les *données* qui constituent la matière première des réseaux d'entreprise. Ces données peuvent être des documents uniques (rapports, brevets, ...) ou des recueils de données (destinés généralement aux bases de données). Actuellement, ces données sont de plusieurs types : textuelles, graphiques ou animées.

Les sommets correspondent aux lieux de transit où les informations sont traitées. Ils peuvent être considérés comme des *nœuds* [Schael 97] constitués d'un ou plusieurs acteurs humains (par exemple, un(e) secrétaire ou un bureau administratif), ainsi que d'acteurs logiciels ou matériels (par exemple, serveurs distribuant l'information) [Wilson 94]. Chaque acteur effectue une ou plusieurs activités, correspondant à son *rôle* dans le réseau, activités qui consistent à transformer les données pour un objectif donné (comme par exemple la copie d'un rapport pour archivage). Les sommets peuvent également être considérés comme des *activités* [WFMC 99], faisant appel aux ressources humaines et/ou matérielles (les acteurs).

Les *arcs* représentent donc les échanges de données entre les nœuds ; ils sont également appelés *relations* entre nœuds [Schael 97] ou *transitions* entre activités [WFMC 99]. Ces échanges peuvent se faire soit de façon synchrone (par exemple : un téléphone ou une vidéo conférence pour la synchronisation temporelle, une réunion pour la synchronisation spatio-temporelle), soit de façon asynchrone (par exemple : un tableau d'affichage pour une communication synchronisée uniquement dans l'espace, la messagerie électronique pour une communication synchrone ni dans le temps, ni dans l'espace).

Tous ces échanges de données se font dans le cadre d'un objectif fixé, qui peut être soit la réalisation d'une *tâche* simple (par exemple : la transmission de la liste des ouvrages disponibles, qui peut être considérée comme une tâche simple pour peu que cette liste soit à jour), soit la mise en œuvre d'une *procédure* c'est-à-dire d'un ensemble de tâches interdépendantes (par exemple : l'étude de brevetabilité d'un procédé).

Des données sont échangées entre les nœuds dans le but de réaliser une procédure. Ceci conduit à un maillage que nous ne pouvons pas encore qualifier d'organisation. Pour le définir comme tel, il faut prendre en compte la *structure* établie par la position des nœuds dans le système et par leurs relations. Cette structure organisationnelle peut être découpée selon les rôles *joués* par les agents (par exemple, une administration organisée en services), ou selon des *sous-objectifs* (par exemple, les étapes de fabrications d'un produit dans une chaîne de

montage est structurée autour de ce produit). On parle alors de *l'articulation* de l'organisation, correspondant à la connexité du graphe [Berge 70].

Ainsi, à chaque procédure correspond un réseau plus ou moins structuré, composé de nœuds entre lesquels circulent des données. Par exemple, la figure I.2a schématise l'activité d'un call-center, faiblement articulé : un responsable gère deux équipes : une équipe administrative et une équipe de télé-opérateurs. La figure I.2b représente l'activité d'un système administratif, hiérarchisé et plus fortement articulé.

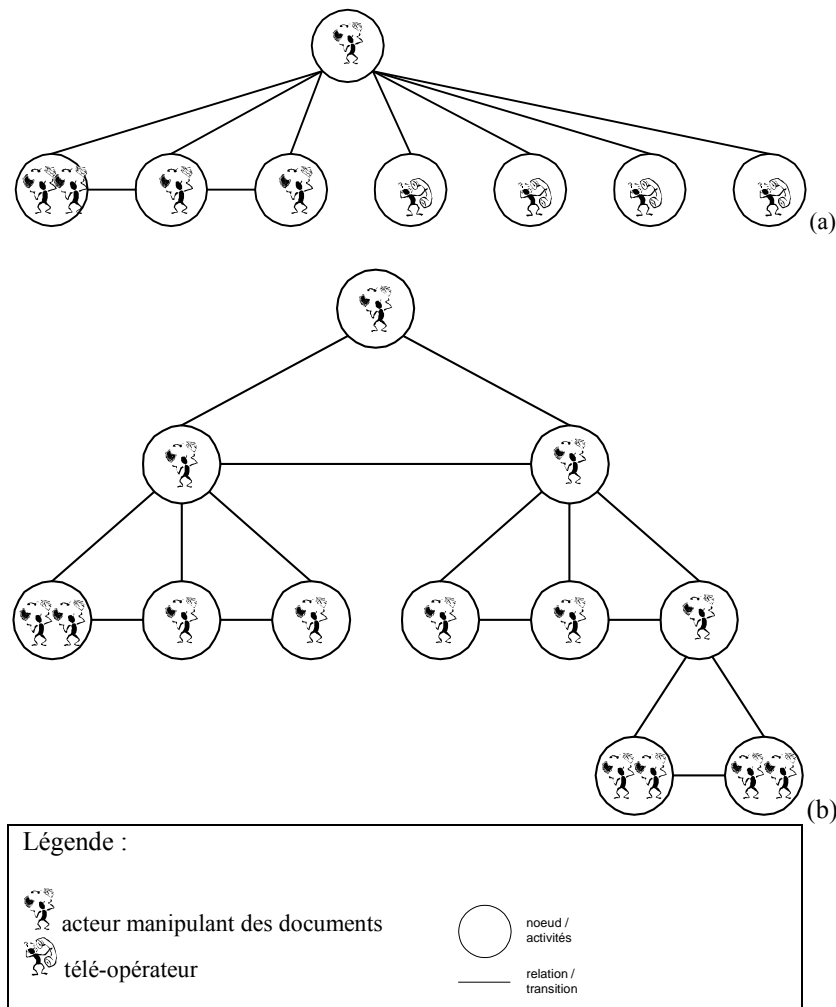


Figure I.2. Réseau représentant l'activité d'un système administratif
(a) à faible degré de structuration
(b) à fort degré de structuration

Ces notions d'articulation, de nœuds, de transitions, d'acteurs et de rôles peuvent être synthétisées dans cette définition d'un réseau d'entreprise.

Définition I-1 : un réseau d'entreprise est constitué de nœuds reliés par des transitions. Chaque nœud est constitué d'un ou plusieurs acteurs réalisant des opérations (activités) sur

les données qui y circulent. Le réseau s'articule soit sur les rôles joués par les agents, soit sur les sous-objectifs d'un but commun.

Différents réseaux peuvent se combiner. En effet, la plupart des entreprises ne possèdent pas un réseau unique mais un ensemble de réseaux plus ou moins coopératifs qui peuvent interférer, chacun étant associé à un produit ou à une procédure. Par exemple, la figure I.3 utilise plusieurs réseaux pour représenter un département gérant trois types de produits.

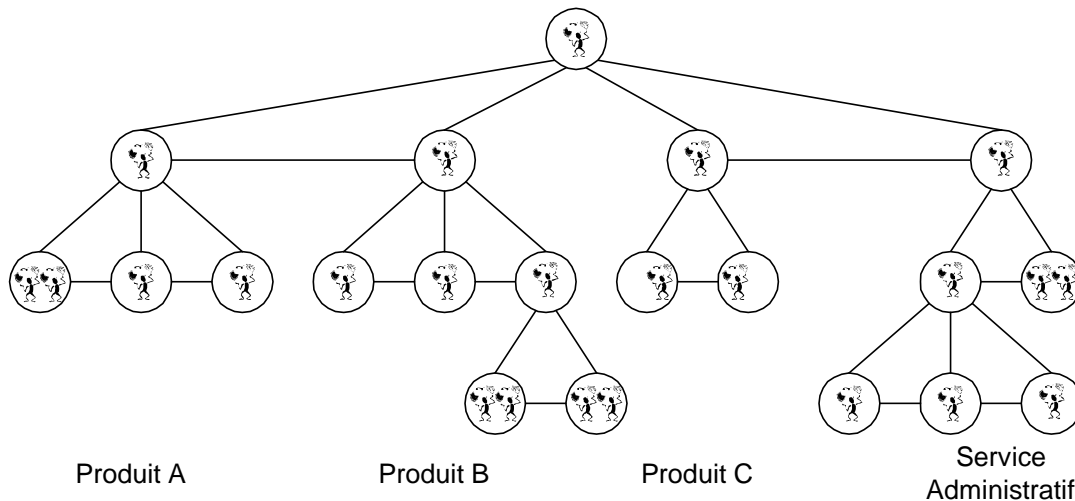


Figure I.3. Représentation d'un système à l'aide de plusieurs réseaux.

Les acteurs d'un réseau coopératif s'échangent des informations selon leurs rôles et les tâches (engagements) auxquelles ils sont astreints. Ces échanges interpersonnels ne constituent cependant qu'un des aspects de la coopération inter-acteurs en entreprise. Afin d'en obtenir une vision complète, il faut répertorier, de manière exhaustive, les types de réseaux que l'on peut rencontrer

I.1.2 La coopération dans un réseau d'entreprise

La coopération entre acteurs ou agents d'un réseau résulte de leur capacité à collaborer, co-décider et coordonner leurs activités [De Michelis 94]. Mais la codécision, mettant en jeu les mécanismes de négociation, peut être considérée comme une activité coopérative particulière qu'il est possible de décrire par des activités de communication, de coordination et de collaboration². La figure I.4 représente le mécanisme de négociation : une solution est proposée par un acteur (1) ; les autres acteurs approuvent cette solution ou proposent des modifications (2). Les modifications peuvent être refusées ou prises en compte pour la

² La collaboration implique la mise en commun d'informations entre différents acteurs, ce qui entraîne alors des modifications d'actions ou d'actes chez ces acteurs dans le cadre d'une procédure [Goodman 86].

proposition de nouvelles solutions. Et ainsi de suite jusqu'à l'obtention d'une solution partagée par tous (3).

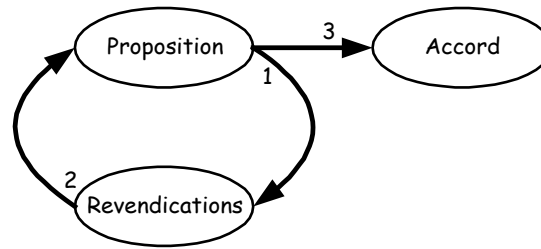


Figure I.4. Mécanisme de négociation

La coopération peut être définie de la façon suivante en s'inspirant de [Schael 97] :

Définition I-2 : *La coopération est une aptitude à la communication, la coordination et la collaboration d'un ensemble d'acteurs pour la réalisation d'un objectif commun.*

La communication est indispensable : deux acteurs ne peuvent coopérer sans échanger des informations, soit directement, soit indirectement par modification de leur environnement commun (par exemple l'ensemble des usages qui régissent le fonctionnement d'un bureau est mis au point progressivement dans le but de faciliter le travail de chaque acteur, sans communications directes préalables entre ceux-ci).

La coordination, quant à elle, est nécessaire dans tous les types d'organisations, principalement celles qui peuvent être représentées par un réseau à faible degré de structuration. Dans le cas du travail à la chaîne, par exemple, les actions de chaque opérateur sont fortement liées aux actions de l'opérateur qui le précède. La réalisation de l'objectif commun n'est possible que par une action coordonnée de l'ensemble des acteurs. De même, dans le cas des équipes sportives, la coopération prend souvent la forme de coordinations de mouvements tactiques ou techniques individuels.

La collaboration, l'activité commune d'un ensemble d'acteurs, est une troisième caractéristique de la coopération. Elle est l'expression d'une activité synchrone. Par exemple, le tri postal est une activité commune d'acteurs travaillant dans un même espace de temps pour la réalisation de l'objectif commun qui est le tri de l'ensemble du courrier.

Au niveau de l'individu, la coopération est modulée par la confiance qu'il porte envers les autres acteurs. Cette confiance peut revêtir trois formes [Zucker 86] : elle peut être fonction de caractéristiques propres à la personne telle que l'appartenance à un groupe, à une famille, à

une ethnie (il s'agit de la confiance *intuitu personae*) ; elle peut également être liée aux précédents échanges, à la réputation de l'autre acteur (il s'agit alors de la confiance relationnelle) ; elle peut être liée à une structure formelle garantissant les activités de l'autre acteur (cette confiance est nommée *institutionnelle*).

Nous disposons, à ce stade, d'un modèle général des systèmes d'informations distribués dont il est intéressant d'étudier la compatibilité avec les principales architectures d'organisation du travail. La partie suivante présente ces différentes architectures ainsi que les structures possibles de systèmes d'information distribués.

I.1.3 Architecture d'organisations du travail

Nous traitons dans cette partie de l'architecture des organisations du travail, et non pas des théories relatives à l'organisation du travail telles que le taylorisme, le fordisme, la bureaucratie ou les théories de l'école néo-classique (théories plus récentes privilégiant l'individu) [Alberto 93]. Bien que liées à l'architecture de l'organisation, ces théories mettent l'accent sur leur fonctionnement interne ne faisant pas l'objet de cette thèse.

Les entreprises sont généralement structurées autour d'une architecture fonctionnelle, d'une architecture divisionnelle ou d'une architecture matricielle [Alberto 93].

- L'architecture fonctionnelle découpe l'organisation en unités correspondant aux fonctions (administratives, commerciales, ...). La figure I.5a illustre cette architecture sur un exemple, mettant en jeu quatre unités, à savoir l'unité de réception, l'unité de production, l'unité de gestion des données commerciales et l'unité administrative. Chaque unité utilise son propre système de gestion. Du point de vue des systèmes d'information, cette architecture conduit souvent à la mise en place, dans chaque unité, de logiciels et de bases de données spécialisés avec pour conséquence la difficulté de mettre en place un système de gestion global cohérent, sauf si celui-ci a été pensé avant toute initiative locale

La figure I.5b schématise quatre systèmes de gestion de données associés chacun à une unité de l'entreprise décrite dans l'exemple. La structure de réseau associée est un arbre à quatre branches (correspondant aux quatre unités de l'entreprise). La direction est reliée à chacun des systèmes par un filtre logiciel ou humain.

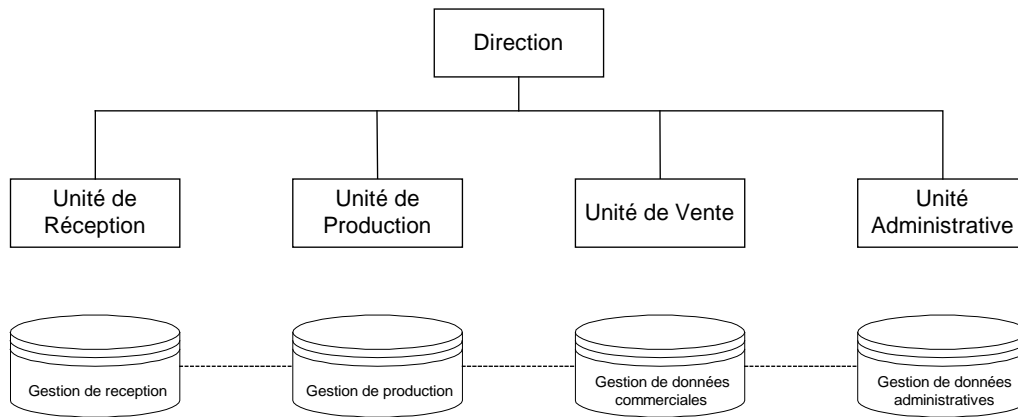


Figure I.5a. Exemple de structure fonctionnelle

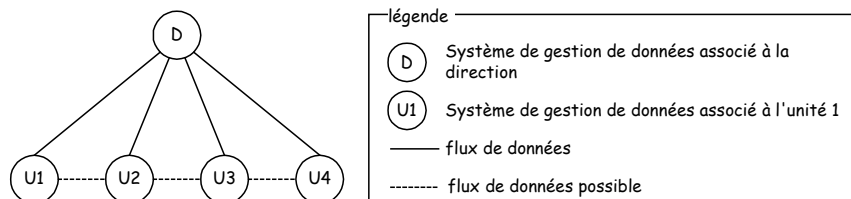


Figure I.5b. Exemple de réseau associé à une structure fonctionnelle

- L'architecture divisionnelle³, quant à elle, découpe la structure de l'entreprise en unités, en fonction des marchés qu'elle occupe. Ces unités peuvent être physiquement éloignées, et sont généralement elles-mêmes structurées selon une architecture fonctionnelle (fig. I.6a). Les systèmes d'information, au sein de chaque unité, sont alors composés de logiciels et bases de données couvrant l'ensemble du cycle de vie d'un projet ou d'un produit (par exemple, nous pouvons retrouver les quatre unités définies dans la figure précédente, l'unité de réception, l'unité de production, l'unité de gestion des données commerciales et l'unité administrative). Les liens entre ces différents composants d'une même unité sont moins problématiques que dans le cas des entreprises à structure fonctionnelle du fait de la taille réduite des unités. De plus, l'architecture du SID d'une unité orientée vers un produit ou un marché, peut être plus ou moins calquée, réutilisée sur une autre unité. Le problème peut alors provenir de la volonté de connecter transversalement un ensemble des bases de données relatives à une même activité (par exemple : les données relatives aux différents services achats).

La figure I.6a présente l'architecture divisionnelle d'une entreprise tournée sur trois marchés correspondant à trois systèmes de gestion d'information. Chacun de ces

³ L'architecture divisionnelle est également appelée de type BU (Business Unit ou Unité de Production). Cette terminologie n'est pas fixée et varie en fonction de la taille et du type d'activité.

Le système peut être découpé en sous-systèmes plus orientés vers les fonctions des acteurs. La structure de réseau associée (fig. I.6b) est une structure arborescente. La direction de chaque unité reçoit des données consolidées des niveaux inférieurs. La direction générale (DG) est elle-même reliée aux directions des différentes unités.

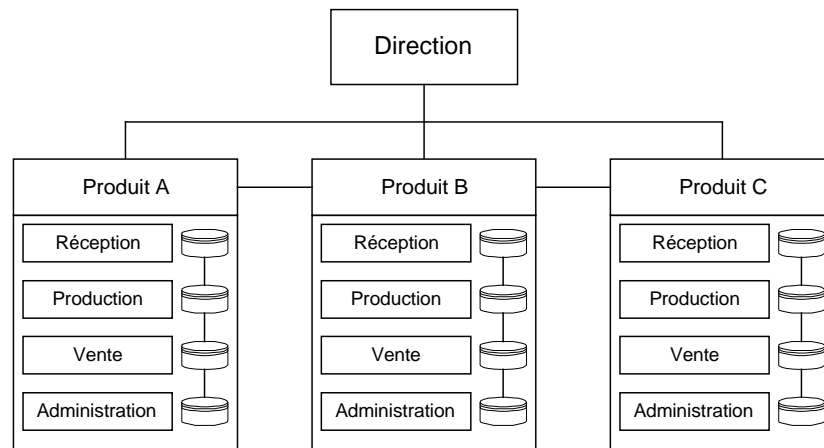


Figure I.6a. Exemple de structure divisionnelle

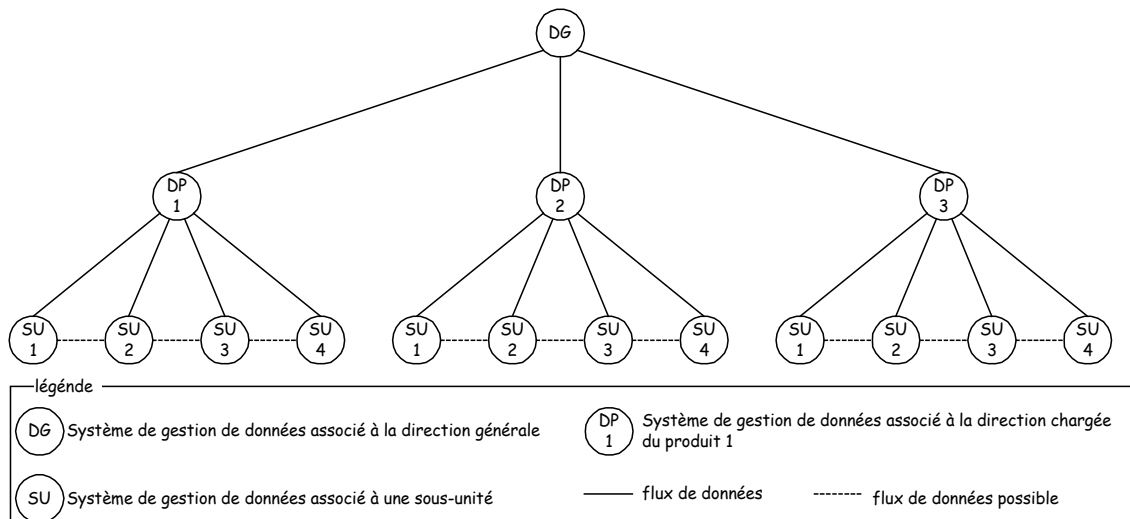


Figure I.6b. Exemple de réseau associé à une structure divisionnelle

- L'architecture matricielle est à la fois structurée en unités fonctionnelles et en produits ou projets. L'entreprise est découpée en unités fonctionnelles. Les responsables de ces unités fonctionnelles coordonnent les activités des chefs de projets, qui gèrent les activités d'équipes. Chaque acteur a donc au moins deux responsables : son chef de projet et son responsable d'unité (fig. I.7a). Cet état de fait peut être source de problèmes de négociation entre les chefs de projets, et les responsables fonctionnels. Les besoins en systèmes d'information dans les organisations à structure matricielle concernent donc essentiellement la gestion des projets. Il est également indispensable

que le système informatique intègre un module d'aide à la décision, permettant aux différents responsables de partager au mieux les ressources humaines et matérielles. La construction d'un système d'information devient vite complexe dans le cadre d'une grande entreprise. En effet, pour posséder des informations cohérentes sur l'ensemble de la matrice (c'est-à-dire l'ensemble des projets et l'ensemble des unités), il est nécessaire de disposer d'un système global couvrant l'ensemble de l'entreprise, ce qui devient souvent difficilement gérable.

La figure I.7a schématise l'architecture matricielle d'une entreprise possédant quatre unités et gérant trois projets. Ce type d'architecture nécessite de pouvoir disposer d'un système de gestion de projets global (fg I.7b) qui doit être relié aux systèmes d'information des différentes unités qui gèrent les ressources humaines et matérielles de l'entreprise, ceci afin de faciliter les négociations entre les différents responsables.

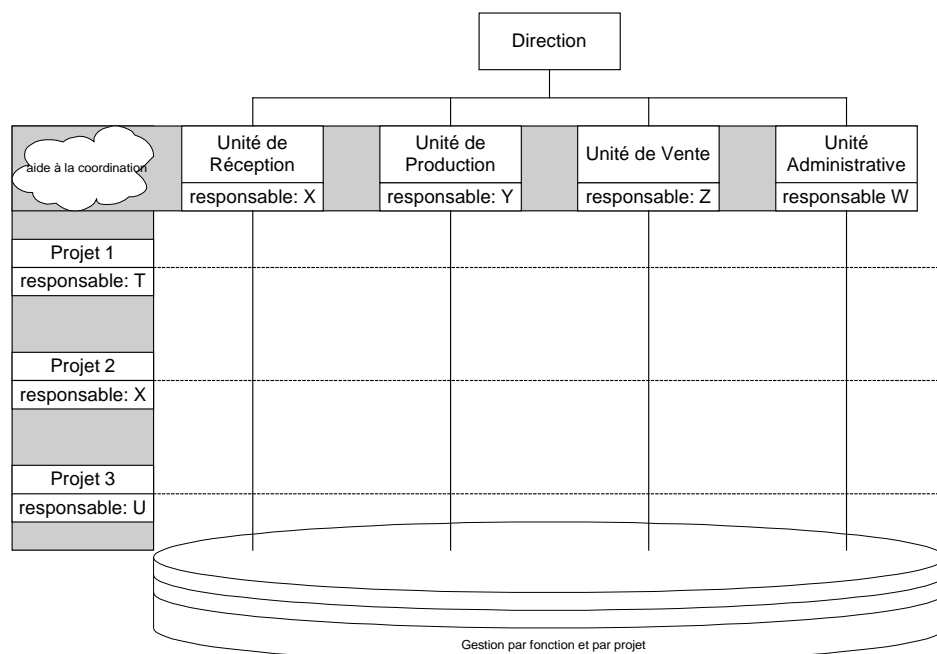


Figure I.7a. Exemple de structure matricielle

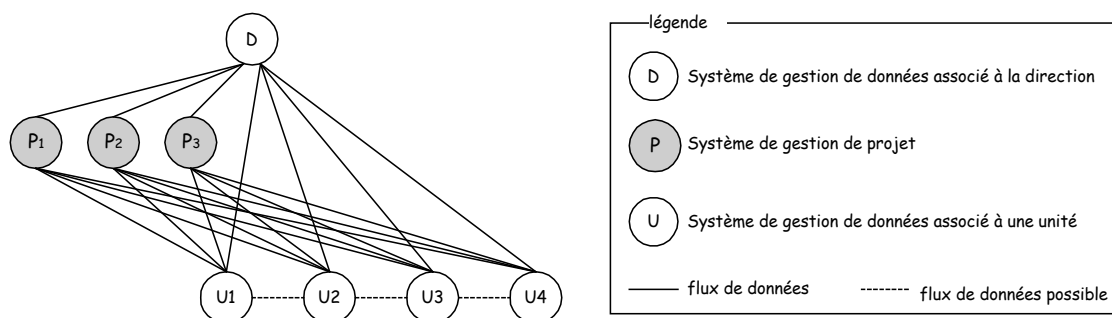


Figure I.7b. Structure de réseau associée à une structure matricielle

I.1.4 Conclusion sur la structure des systèmes d'information distribués

Des trois modèles d'organisation étudiés, il apparaît que la structure divisionnelle est la mieux adaptée pour la mise en place d'un système d'information clair, rapide et réutilisable. En effet, la structure fonctionnelle s'articule sur des systèmes centralisés, inévitablement plus lents, et la structure matricielle, qui nécessite un système de gestion des données cohérent sur l'ensemble de l'organisation, implique une structure de réseau compliquée.

De cette étude des organisations effectives dans le monde du travail, il ressort que les organisations à structures arborescentes sont plus faciles à gérer. Dans la partie suivante, nous proposons de passer en revue les modèles d'organisation d'agents provenant de l'Intelligence Artificielle Distribuée (IAD) qui, eux aussi, se caractérisent par des échanges d'informations entre entités⁴. Cette description aura pour objectif de guider le choix du type de structure que devra posséder le système d'aide à la gestion d'informations au sein d'une organisation humaine.

I.2 Modèles d'organisations d'agents logiciels en Intelligence Artificielle Distribuée

Selon D. Hofstadter [Hofstadter 85], l'Intelligence Artificielle regroupe les disciplines visant la conception d'une entité reproduisant les activités intellectuelles humaines, soit dans la forme, soit dans la finalité, et qui passerait le test de Turing⁵. L'Intelligence Artificielle Distribuée regroupe, quant à elle, les disciplines visant la conception d'un ensemble d'entités reproduisant les activités intellectuelles humaines [Chaib-Draa 92], [Huhns 87], [Mandiau 93]. Elle propose, notamment dans le domaine des organisations multi-agent, plusieurs structures à ces ensembles d'entités intelligentes. Suite à la définition de ces entités intelligentes, appelées agents, ce paragraphe en propose une classification et une liste d'organisations possibles.

⁴ De nombreux travaux ont d'ailleurs déjà proposé l'utilisation de techniques de l'I.A.D. pour répondre au problème de la gestion distribuée de l'information dans les organisation humaines [Briffaut 98], [Dignum 99], [Handl 99], [O'Brien 98], [Odgers 99], [Quintero 98].

⁵ Le test de Turing se présente sous la forme d'un jeu à trois joueurs A, B et C. Un des deux premiers est une machine. Le rôle de C est de trouver qui est la machine. Le rôle de A est d'induire C en erreur et le rôle de B est d'aider C. Une machine passe le test si le joueur C ne l'a pas détectée [Turing 50].

I.2.1 Définition des agents logiciels

Selon le FOLDOC⁶ (dictionnaire en ligne de l'informatique) recommandé par le site du laboratoire d'Intelligence Artificielle du MIT⁷, l'Intelligence Artificielle est le domaine de l'informatique concerné par les concepts et méthodes d'inférence et par les représentations symboliques des connaissances utilisées pour inférer. Elle peut être vue comme une tentative de modéliser certains aspects de la pensée humaine [Simon 95]. L'IA est également définie comme un domaine de l'informatique essayant de résoudre par ordinateur des problèmes que les humains résolvent généralement plus rapidement. La définition suivante reprend ce concept :

Définition I-3 : *L'Intelligence Artificielle a pour but de faire exécuter par l'ordinateur des tâches (surtout intellectuelles) pour lesquelles l'homme, dans un contexte donné, est meilleur que la machine. [Alliot 95]*⁸

Mais l'Intelligence Artificielle ne suffit pas à l'étude d'un travail coopératif ou du raisonnement d'un groupe de personnes (comme par exemple la recherche d'informations sur des sites répartis ou la résolution commune d'un problème). En effet, ces tâches font intervenir un ensemble d'entités intelligentes qui doivent au minimum communiquer (de manière directe ou indirecte), sinon coopérer, pour atteindre un *objectif commun*. L'Intelligence Artificielle ne traitant pas ces problèmes de groupe, une nouvelle branche est apparue : l'Intelligence Artificielle Distribuée (IAD) [Demazeau 95], [Ferber 95], [Gasser 89], [Gasser 92], [Simon 95].

Définition I-4 : *« L'Intelligence Artificielle Distribuée est l'ensemble des disciplines tendants à faire que des ensembles d'entités imitent des comportements intelligents humains. » [Le Strugeon 95]*

Les disciplines et les outils utilisés dans le domaine de l'I.A.D. peuvent être issus de l'IA. Ainsi trouve-t-on des systèmes experts distribués (tel que Mycin⁹), des ensembles d'entités spécialisées dans la reconnaissance de forme, travaillant ensemble à la

⁶ FOLDOC est l'acronyme de Free On-Line Dictionary Of Computer situé à l'adresse : <http://wombat.doc.ic.ac.uk/foldoc/index.html>

⁷ L'adresse du site du laboratoire est : <http://www.ai.mit.edu/>

⁸ Cette définition a également été donnée par E. Rich [Rich 83] : « Artificial Intelligence is the study of how to make computers do things at which, at the moment, people are better »

⁹ Gleize M-P., Polycopié du cours Systèmes à base de connaissances, Université Paul Sabatier, Toulouse, 1995.

reconnaissance d'une forme globale [Boissier 94], ou encore des ensembles de logiciels travaillant à la recherche d'informations pertinentes sur Internet (par exemple les « Info Agents » de [Aloisi 95]). Plus généralement, ces entités intelligentes utilisant des techniques de l'IA sont appelées agents. Un ensemble d'agents inter-agissant constitue un Système Multi-Agents (SMA).

Il est très difficile de donner une définition commune du terme agent [Flores-Mendez 99], de même qu'il est encore très difficile de donner actuellement une définition commune de ce qu'est un système d'Intelligence Artificielle. Les débats qui ont cours actuellement sur les listes de diffusions en sont la preuve¹⁰.

Un agent peut être défini comme une entité intelligente faisant partie d'un SMA. L'espace dans lequel évolue le système et donc l'agent est appelé le monde. L'agent est capable de percevoir ce qui l'entoure, c'est-à-dire son environnement, et de le modifier dans un but donné et avec une certaine autonomie. Généralement, l'agent n'est pas unique, mais fait partie d'un système multi-agent, il doit donc posséder des capacités de communication.

En ce qui concerne la notion d'intelligence, il s'agit de souligner le principe suivant : une entité est intelligente si elle est capable d'apprendre, c'est-à-dire d'adapter ses connaissances. Un agent peut donc être défini de la façon suivante :

Définition I-5 : *Un agent est une entité adaptative, rationnelle, autonome, capable de communication et d'action.*

Un agent possède généralement des accointances ; il s'agit d'agents avec lesquels il communique ou interagit. Chaque agent possède des connaissances sur son environnement. Ces connaissances sont aussi appelées représentations ou croyances. Un agent possède un ou plusieurs objectifs également appelés buts ou encore désirs. Selon l'importance des buts, l'agent peut être amené à planifier ses actions.

La figure I.8 présente un agent faisant partie d'un système multi-agent situé dans un environnement. L'ensemble constitué du SMA et de l'environnement forme le monde.

¹⁰ le groupe de discussion international sur l'intelligence artificielle se trouve sur le forum comp.ai .

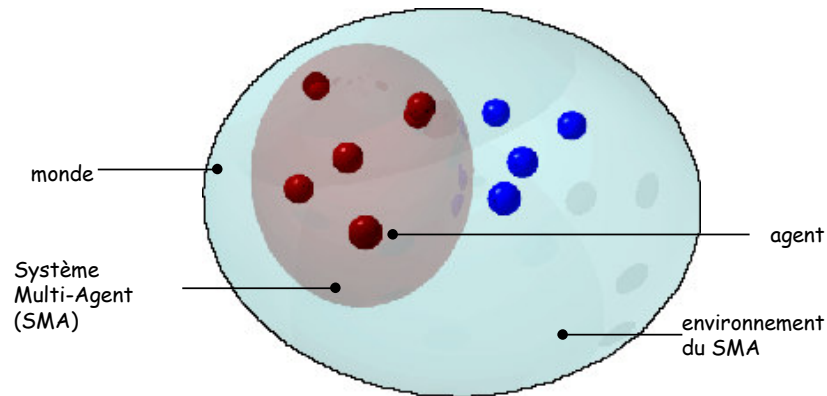


Figure I.8. *Positionnement d'un agent dans le monde*

J. Ferber propose d'associer six fonctions aux agents : la fonction représentationnelle liée à la représentation des connaissances ; la fonction organisationnelle liée à la planification d'actions, à la nature rationnelle et adaptative de l'agent ; la fonction conative liée aux besoins, désirs, à la nature autonome; la fonction interactionnelle liée aux communications; la fonction productive liée aux différentes actions que peut entreprendre l'agent et la fonction conservative liée à la protection, à la conservation de l'agent ou du système [Ferber 95].

Ces six fonctions se déclinent autour de 5 dimensions (tableau I.1) : la dimension 'personnelle' (relative à l'agent) ; la dimension 'environnementale' ; la dimension 'sociale' (relative aux autres agents, c'est-à-dire aux accointances de l'agent) ; la dimension 'relationnelle' (relative aux relations entre l'agent et le monde) et la dimension 'physique' (relative à l'implémentation des fonctions).

Tableau I.1. Grille d'analyse fonctionnelle [Ferber 95]

	Personnel	Environnemental	Social	Relationnelle	Physique
Représentationnel					
Organisationnel					
Conative					
Interactionnel					
Productif					
Conservatif					

La grille d'analyse fonctionnelle permet de classer les différents types d'agents et de systèmes multi-agents. Il sera d'ailleurs intéressant d'utiliser cette grille sur les agents que nous allons proposer pour constituer le système distribué d'aide à la gestion d'information. Selon le type d'agents, les fonctions sont plus ou moins développées, voire inexistantes. Par exemple, les agents assistants que l'on voit apparaître dans les suites bureautiques ont un comportement essentiellement de réaction à des événements [Arafa 99], [van Mulken 99]. En revanche, les systèmes experts distribués ont des capacités de réflexion plus élevées. Plusieurs types d'agents sont décrits dans la partie suivante.

I.2.2 Typologie des agents logiciels

Dans cette partie, les agents sont classés selon les caractéristiques généralement considérées comme principales en IAD, c'est-à-dire selon le degré d'autonomie, de coopération et d'adaptation. Ces trois notions sont très importantes : l'agent a un but à atteindre et il doit pouvoir prendre des initiatives pour atteindre ce but. De plus, pour que l'ensemble des agents constitue un système cohérent, chaque agent doit avoir un certain degré de coopération. Enfin, l'agent doit agir en fonction de son environnement, c'est-à-dire qu'il doit s'adapter à celui-ci. Le plus haut niveau d'autonomie permet à l'agent de planifier ses actions. Le plus haut niveau de coopération accorde à l'agent des capacités de négociation. Le plus haut degré d'adaptation permet à l'agent d'adapter, d'acquérir des connaissances. Un agent possédant ces trois caractéristiques à leur plus haut niveau est appelé "agent intelligent".

Deux types d'agents ont longtemps été utilisés dans les systèmes multi-agents [Chaib-Draa 96] : les agents réactifs dont le comportement est basé sur les stimuli réponses et les agents cognitifs (le terme délibératif est également utilisé) dont le comportement est plus « réfléchi », c'est-à-dire qu'il résulte d'un choix effectué parmi un ensemble d'actions possibles. D'autres types d'agents, qualifiés d'hybrides, utilisant donc ces deux types de comportement, sont ensuite apparus.

Les agents réactifs ne disposent ni de représentation du monde, ni de représentation de leurs actions qui, prises individuellement, consistent surtout en une réaction rapide face à un stimulus. Les agents réactifs possèdent une faible capacité de coopération qui, généralement, est limitée à la communication par messages, le plus souvent de façon indirecte (voir par exemple les travaux sur les sociétés d'araignées [Bourjot 99]). Bien que très simples dans leur conception, un ensemble d'agents réactifs peut avoir un comportement complexe ; il s'agit de l'émergence d'un comportement [Brooks 91]. Ainsi, par exemple, E. Bonabeau propose et utilise un système multi-agents reproduisant le comportement de fourmis pour trouver le chemin le plus rapide entre deux points d'un réseau dont le nombre de nœuds et de passerelles est inconnu [Bonabeau 99]).

Les agents cognitifs quant à eux, possèdent une représentation symbolique de l'environnement sur lequel ils peuvent raisonner. Ils regroupent actuellement plusieurs sous-types d'agents définis de la façon suivante :

- Les agents "intelligents" combinent les trois caractéristiques (autonomie, coopération, adaptativité) à leur plus haut niveau. C'est-à-dire qu'ils planifient leurs actions, sont capables de négocier avec d'autres agents et d'acquérir ou de modifier leurs connaissances. Devant gérer des connaissances et des actions complexes, ces agents sont inévitablement moins 'réactifs' face aux modifications brutales de leur environnement (citons entre autres les travaux de [Tranvouez 99]). Il faut signaler que le terme d'agent intelligent est souvent utilisé pour caractériser des agents dotés de la capacité d'apprentissage [Kufik 00].

- Les agents "collaborants" [Nwana 96] : ce sont des agents cognitifs non apprenants. Ils sont donc à la fois fortement autonomes et coopérants, mais leur adaptativité ne va pas jusqu'à l'adaptation et à l'acquisition des connaissances. La coopération entre de tels agents est plus élaborée que dans le cas des agents réactifs car les agents collaborants doivent être capables de négocier entre eux ou avec l'utilisateur [Rich 97] en fonction de leurs buts ou de leurs tâches. Si les agents collaborants ne possèdent pas un haut niveau d'adaptation, un système composé de tels agents peut, par contre, disposer d'une capacité d'adaptation complexe. Ce qui répond au principe sous-tendant l'IAD que le tout est supérieur à la somme des parties. Les agents collaborants sont surtout utilisés dans les domaines qui nécessitent une décentralisation comme par exemple la

maintenance de réseau, ou encore pour simuler le comportement d'organisations humaines ou animales [Lant 94].

- Les agents "interfaces" [Koda 96], [Lashkari 94], [Lieberman 97] : ce sont ceux que l'on rencontre le plus souvent lors de l'utilisation de certains logiciels de bureautique. Leur fonction consiste à capturer les actions de l'utilisateur (le plus souvent les actions sur le clavier, la souris, mais il est également possible de capturer la voix ou l'expression du visage par utilisation d'une webcam, caméra dédié aux communications internet), ceci afin de modifier leur représentation ou l'aspect de l'interface selon un schéma de règles préétablies [Laurel 97]. Ces agents possèdent en général une capacité de coopération limitée à l'échange d'informations concernant les actions de l'utilisateur. Ils sont principalement utilisés pour l'assistance à l'utilisateur dans le cas d'interfaces aux fonctionnalités nombreuses et complexes (par exemple, certaines suites bureautiques), mais également dans le domaine des systèmes tuteurs intelligents [Adam 95] afin de faciliter l'apprentissage et de diminuer la surcharge cognitive chez l'apprenant.

- Les agents "informations" [Rhodes 00] [Rus 97]: ces agents qui apparaissent de plus en plus sur le marché sont dédiés à la recherche d'information, principalement sur l'internet. Il faut cependant noter que peu de logiciels de ce type méritent réellement le qualificatif d'agent ; beaucoup ne sont que des programmes se déclenchant à date fixe, sans capacité d'adaptation, ni de coopération (par exemple, les nombreux logiciels de veille qui se déclenchent à heure fixe pour rechercher des informations sur un produit ou un concurrent). Par contre, les agents "informations" (tels que SAIRE [Odubiyi 97] ou softbot [Etzioni 94]) possèdent une grande autonomie ; ils agissent seuls, soit en fonction d'un calendrier, soit en fonction d'un manque d'information, soit en fonction d'une nouvelle disponibilité d'information. Ils sont capables d'adapter leurs fonctionnements en fonction du besoin de l'utilisateur ou de la quantité ou pertinence de l'information (par exemple, si un nouveau site propose des informations plus pertinentes, ce site sera alors privilégié pour les recherches futures). Toutefois, ces agents agissent le plus souvent de manière isolée, ce qui peut entraîner un problème de redondance d'informations : les agents ne sachant pas ce que leurs accointances ont trouvé, peuvent télécharger les mêmes informations. Il est à noter que l'on trouve également la notion d'agent mobile, il s'agit en fait d'une caractéristique supplémentaire

permettant à un agent, généralement à un agent information, de parcourir le réseau à la recherche d'informations [White 97].

La figure I.9 résume ces différents types d'agents cognitifs ainsi que leurs degrés d'autonomie, d'adaptativité et de coopération. Les agents collaborants sont autonomes et coopérants, leurs capacités de négociation impliquent une faculté d'adaptation, de modification de leur comportement. Les agents "interfaces" réagissent aux sollicitations de l'utilisateur, ils sont donc peu autonomes. Ils s'adaptent au comportement de l'utilisateur (par exemple, ils peuvent changer leur représentation) et communiquent leurs déductions sur son état (débutant, occasionnel, expert) à d'autres agents "interfaces". Les agents "informations" peuvent agir de façon très autonome pour la recherche d'une information au travers de différents sites, mais souvent indépendamment des autres agents. Ils sont capables d'adapter leur stratégie de recherche en fonction des informations trouvées ou non trouvées (par exemple, pour modifier la requête ou chercher de nouvelles sources d'information).

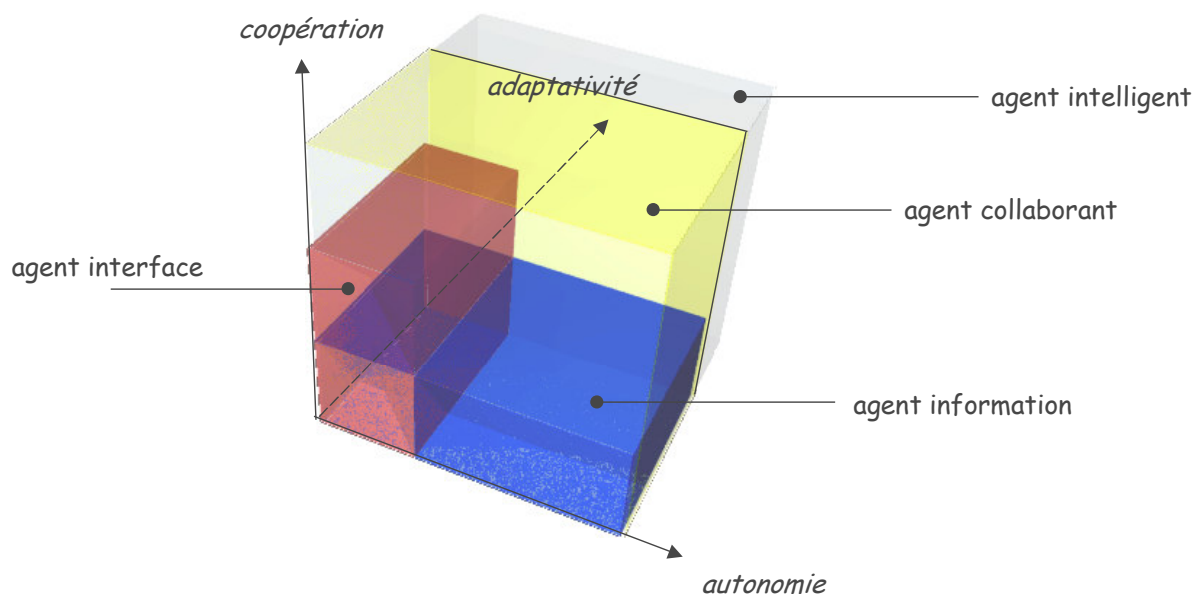


Figure I.9. Degrés d'autonomie, de coopération et d'adaptativité des principaux agents cognitifs.

Le choix du ou des types d'agents à utiliser dépend en fait du système multi-agent le plus pertinent pour le problème à résoudre. Ainsi, par exemple, certains SMA n'utiliseront qu'un seul type d'agents regroupés par objectif, d'autres plusieurs types correspondant à des rôles précis nécessaires à la résolution. En effet, si les SMA sont tous composés d'agents communiquant pour réaliser un objectif commun, il existe plusieurs façons de les organiser.

I.2.3 Les modèles d'organisation

Certains systèmes multi-agents pouvant être vus comme un ensemble de nœuds, contenant un ou plusieurs agents pouvant avoir des objectifs communs ou propres, reliés par les flux de données, l'architecture de ces SMA devient alors similaire à celle des réseaux coopératifs évoqués précédemment.

Il existe trois façons de structurer un SMA [Le Strugeon 95]:

1 - selon une *centralisation/décentralisation des mécanismes de contrôle*, ce qui entraîne une modification du degré de coopération et d'autonomie des agents (une architecture centralisée affecte l'autonomie et la coopération des agents au contraire d'une architecture décentralisée). Avec ce type de découpage, le rôle des agents importe peu. Dans le cas d'une centralisation extrême, un agent contrôle l'ensemble du système composé d'autres agents peu autonomes et peu coopérants. Dans le cas d'une décentralisation poussée, chaque agent est "maître de soi", c'est-à-dire qu'il se contrôle et prend des décisions, cependant, pour être en présence d'un SMA cohérent, il est nécessaire que les agents coopèrent fortement ensemble¹¹.

2 - selon les *rôles* joués par les agents (comme le permet la plateforme MADKIT¹² [Ferber 97]). Les mécanismes de contrôle sont alors distribués aux agents en fonction de leurs rôles. Ce type de découpage permet d'obtenir des systèmes de type "marché" constitués d'agents coordinateurs lançant des appels d'offres auxquels répondent des agents exécutants [Bensaid 97]. On peut également trouver des découpages selon le type de connaissances à gérer (par exemple sociales, personnelles et environnementales), ce qui donne généralement lieu à une organisation de type hiérarchie [Ito 98], [Laird 97].

3 - selon les *objectifs* visés (dans le cas d'un système à objectifs multiples par exemple). Dans ce cas des sous-systèmes eux-mêmes structurés selon les deux modes précédents (selon les rôles ou selon le contrôle) peuvent coexister [Alvares 98].

¹¹ Chaib-draa B., Agents et systèmes multiagents. (IFT 64881A). Notes de cours. Département d'informatique. Faculté Des Sciences et de génie. Université Laval. Québec. Novembre 1999.

¹² MADKIT est l'acronyme de Multi-Agent Development Kit.

Globalement, il est possible de classer les différentes architectures de SMA sur base de la notion d'OMA¹³ (Organisation Multi-Agent) introduite par [Le Strugeon 95]. Celle-ci distingue :

- les OMA à structure hiérarchique, telles que les hiérarchies rigides qui ne poursuivent qu'un objectif opérationnel et où le contrôle est centralisé sur un agent qui communique des ordres (fig. I.10) [Ito 98], [Odubiyi 97].

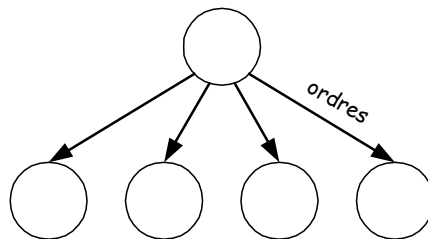


Figure I.10. OMA à structure hiérarchique

- les OMA de type hétéroarchitecturales telles que les organisations :

- de type marché, composé d'agents coordinateurs et d'agents exécutants. Une telle OMA possède une structure un peu plus décentralisée autour de plusieurs objectifs opérationnels (fig. I.11), citons par exemple, l'architecture MAGIQUE [Bensaid 97],

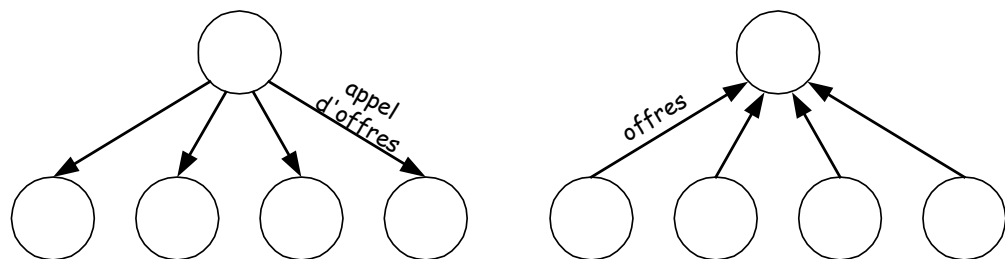


Figure I.11. OMA à structure de marché poursuivant deux objectifs

- de type communauté où le contrôle est fortement distribué et dont les membres possèdent les mêmes capacités, et poursuivent des objectifs multiples (fig. I.12), tels que le système proposé par [Minar 98] par exemple (citons également le modèle d'agents ASTRO qui permet de telles architectures [Ocelllo 98]),

¹³ Un système multi agent peut être qualifié d'organisation multi agent (OMA) lorsque les agents qui le composent possèdent une connaissance des autres et qu'ils suivent des règles fixées par le système global tout en gardant leurs capacités d'autonomie [Mandiau 99].

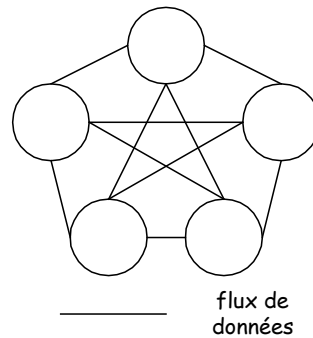


Figure I.12. OMA à structure de communauté composée de 5 agents de même niveau

- de type société où le contrôle est décentralisé autour d'unités, non nécessairement du même type, qui poursuivent chacune un objectif opérationnel (tel que le propose l'architecture ASIC [Boissier 94]). Ces unités doivent s'ajuster mutuellement par des principes de négociation (fig. I.13).

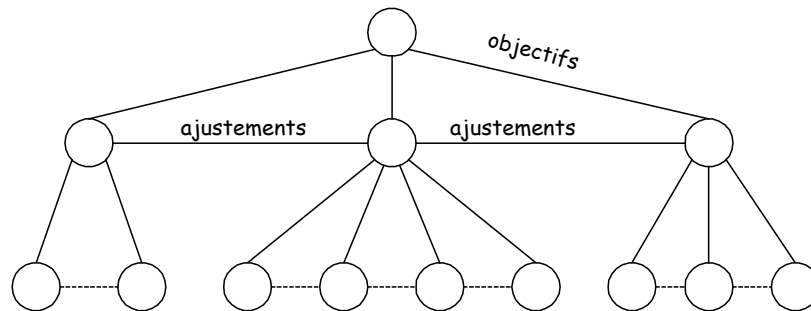


Figure I.13. OMA à structure de société composée de 3 entités

Jusqu'à ces dernières années, les organisations multi-agent proposées étaient constituées d'agents de même nature. Il est vite apparu nécessaire d'utiliser les différents types d'agents pour bénéficier des avantages des différents types d'agents. Par exemple, la plate-forme Magique [Bensaid 99], est basée sur une architecture de marché où des agents lancent des offres auxquelles répondent des agents de service. De même, la plate-forme GEAMAS [Soulié 98] est basée sur une architecture hiérarchique à trois niveaux. GEAMAS permet de construire des applications de simulation de systèmes complexes. L'organisation agent de GEAMAS est composée de trois niveaux : un niveau société (macro-agent) pouvant être détaillé en un niveau secondaire composé d'agents cognitifs (medium-agents), ce niveau étant détaillé en agents réactifs (micro-agents), chaque agent ayant la même architecture. On trouve donc ici une intéressante composition de trois types d'agents.

Les solutions proposées par les systèmes multi-agents pour résoudre le problème de la distribution de l'information sont nombreuses, mais ne correspondent pas réellement aux types d'organisations des entreprises. Si une ressemblance peut être remarquée entre la

structure divisionnelle d'une entreprise et les organisations multi-agents de type société, cette adéquation est surtout structurelle. On constate bien dans les deux cas un découpage de l'organisation par unités poursuivant des objectifs, mais ce n'est pas suffisant pour guider vers la spécification d'un système d'information distribué facilitant la coopération entre les acteurs de l'organisation humaine.

En fait, pour que l'organisation multi-agents soit proche de l'organisation humaine, elle doit être structurée selon les rôles joués par les agents, l'idéal serait de faire correspondre ces rôles à ceux joués par les agents humains, de manière à optimiser l'aide apportée au traitement de l'information.

Le modèle organisationnel holonique, décrit dans la partie suivante, apporte une réponse à ce problème. Ce modèle organisationnel présente l'avantage de proposer un modèle général et récursif issu d'une réflexion sur les organisations sociales et les organismes vivants. Nous montrerons que les règles qui y sont décrites peuvent également être utilisées pour la spécification de systèmes multi-agents.

I.3 Un modèle unificateur : les systèmes holoniques

Il y a environ 30 ans, l'écrivain et philosophe hongrois Arthur Koestler [Koestler 69] qui étudiait les organismes vivants et les organisations sociales a mis en évidence le fait que dans la vie réelle, une entité, pouvant être un tout ou une partie, ne peut être considérée comme un absolu. Il a alors introduit le concept de Holon, concaténation de « holos », signifiant le tout et du suffixe « on » suggérant une partie comme dans « proton ».

Un Holon est défini par Koestler comme une partie d'un tout ou d'une organisation plus large, répondant strictement à trois conditions : être *stable*, avoir une capacité d'*autonomie* et être capable de *coopérer*.

- La stabilité signifie qu'un holon est capable de faire face et de réagir lorsqu'il est sujet à des sollicitations fortes ou à des perturbations importantes.
- L'autonomie suggère qu'un holon soit capable de s'autogérer lorsqu'il est soumis à des sollicitations, afin de réaliser ses objectifs personnels.

- La capacité de coopérer signifie que les holons sont en mesure de coexister avec d'autres holons ou d'autres couches de holons, et sont capables de travailler sur des objectifs et dans des projets communs.

Nous pouvons constater dès à présent la correspondance assez forte avec la définition d'un agent (cf. I.2.1). Dans les systèmes holoniques, l'adaptativité a surtout pour but la stabilité du holon.

L'originalité du concept holonique, par rapport aux approches systémiques traditionnelles, réside dans le fait que grâce à cette triple caractérisation des composants d'un système, Koestler a pu mettre en évidence une condition nécessaire de viabilité et de pérennité des systèmes sociaux. De plus, Koestler ne se limite pas à ces trois règles de base, mais propose un ensemble de 65 règles décrivant les notions de dualité coopération-autonomie, de communication et d'architecture.

Les idées de Koestler ont fait l'objet de plusieurs adaptations, notamment dans un projet international baptisé « Intelligent Manufacturing Systems » [van Brussel 96] [Wyns 99], afin de constituer l'un des paradigmes sur lequel pourrait se construire l'usine du futur. Ces idées sont aussi appliquées dans le domaine de la robotique [Arai 97], ainsi que dans le domaine de la psychologie cognitive [Young 95]. On doit également citer les travaux de l'équipe de Siekmann [Gerber 99] sur la description formelle d'agents holoniques et ses propositions d'application dans les domaines de la planification des transports, des systèmes de productions flexibles et des Robocup¹⁴.

Certes, le modèle holonique ne s'applique pas à toutes les organisations ; il est cependant particulièrement adapté aux organisations ayant une structure de hiérarchie flexible (les éléments devant posséder une relative autonomie), où les notions de rôle, de responsabilité et de coopération sont importantes. De par sa définition récursive (un holon peut être décomposé en holons), le modèle holonique est efficace pour la spécification d'une organisation multi-agents bien adaptée aux organisations humaines, plus particulièrement aux organisations humaines d'entreprise de nature hiérarchique, voire même holonique.

¹⁴ Une robocup est un match de football entre robots ou entre équipes de robots. Les dimensions du terrain sont bien sûr réduites pour des raisons pratiques. Ces matches permettent de tester les robots tant en ce qui concerne leurs performances techniques et qu'en ce qui concerne leurs performances cognitives et collectives [Kitano 98].

En annexe de son livre [Koestler 69], Arthur Koestler fournit une série de règles définissant les systèmes holoniques qu'il appelle systèmes hiérarchiques ouverts (SHO) ou encore holarchie.

En premier lieu, l'interprétation des règles que nous avons jugées particulièrement pertinentes en ce qui concerne les systèmes multi-agents est présentée. Puis, une définition formelle, fortement inspirée des définitions de [Gerber 99], est présentée afin de souligner la récursivité du modèle holonique multi-agent.

I.3.1 Définition informelle des systèmes multi-agents holoniques

Les règles proposées par A. Koestler dans son ouvrage peuvent être regroupées en cinq ensembles. Le premier porte sur l'équilibre nécessaire entre les propriétés d'autonomie et de coopération des holons (ce que Koestler qualifie "d'effet Janus"). Le second a trait à la structure arborescente d'un système holonique. Le troisième traite des mécanismes d'action et de contrôle au sein des systèmes holoniques. Le quatrième est relatif au protocole de communication entre holons. Enfin, le cinquième ensemble de règles concerne l'adaptation des systèmes holoniques à leur environnement.

Les règles présentées ci-après proviennent des 65 règles que A. Koestler a structurées en dix groupes et que nous avons filtrées pour en extraire les règles qui se rapprochent le plus de notre problématique. La numérotation initiale des règles se trouve suite à chacune d'elle, en italique.

I.3.1.1 Règles relatives à l'effet Janus¹⁵

Le premier ensemble de règles concerne "L'effet Janus", c'est-à-dire la dualité entre l'autonomie d'un agent holonique et son intégration dans une holarchie. Cette notion est importante ; elle souligne la nécessité d'avoir un équilibre entre les capacités d'autonomie et de coopération des agents holoniques, plus un agent est autonome, moins il est tributaire des autres agents. Inversement, un agent coopérant fortement avec d'autres agents est limité dans ses actions.

¹⁵ Janus, fils de Mars, est une divinité romaine à deux visages, il est le gardien des Portes, des Passages et des Commencements. Ses deux visages symbolisent les deux faces d'une porte. Il est possible de considérer qu'un holon possède deux visages, l'un tourné vers le bas du système, le holon se considérant comme un tout composé de parties, l'autre visage tourné vers le haut, le holon se considérant comme une partie du système.

Règle 1.1 : L'organisme doit être considéré comme une hiérarchie à plusieurs niveaux de sous-ensembles semi-autonomes, se ramifiant en sous-ensembles d'ordre inférieur, et ainsi de suite. On appellera holons les sous-ensembles de n'importe quel niveau de la hiérarchie. *(correspondant à la règle 1.2 de [Koestler 69])*

Règle 1.2 : Dans le domaine de la vie il n'existe ni parties ni totalités au sens absolu. Le concept de holon a pour but de concilier la conception holiste et la conception atomiste. *(correspondant à la règle 1.3 de [Koestler 69])*

Règle 1.3 : Les holons biologiques sont des systèmes ouverts autorégulateurs qui ont à la fois les propriétés autonomes des totalités et les propriétés de dépendance des parties. Cette dichotomie apparaît à chaque niveau de chaque type d'organisation hiérarchique; on l'appellera effet Janus, ou principe de Janus. *(correspondant à la règle 1.4 de [Koestler 69])*

Règle 1.4 : Chaque holon a une double tendance à conserver et affirmer son individualité en tant que totalité quasi autonome, et à fonctionner comme partie intégrée d'une totalité plus vaste (existante ou en cours d'évolution). Cette polarité de la tendance à l'affirmation du moi et de la tendance à l'intégration est inhérente au concept d'ordre hiérarchique; c'est une caractéristique universelle de la vie. Les tendances à l'affirmation du moi sont l'expression dynamique de la totalité du holon, les tendances à l'intégration sont celles de sa partialité. *(correspondant à la règle 4.1 de [Koestler 69])*

Dans ces règles apparaît également le fait que l'organisation ne peut être réduite à un super-agent ou à une somme d'agents. Ce qui rejoint l'idée que le tout est plus grand que la somme des parties, concept que l'on retrouve dans le domaine des organisations émergentes.

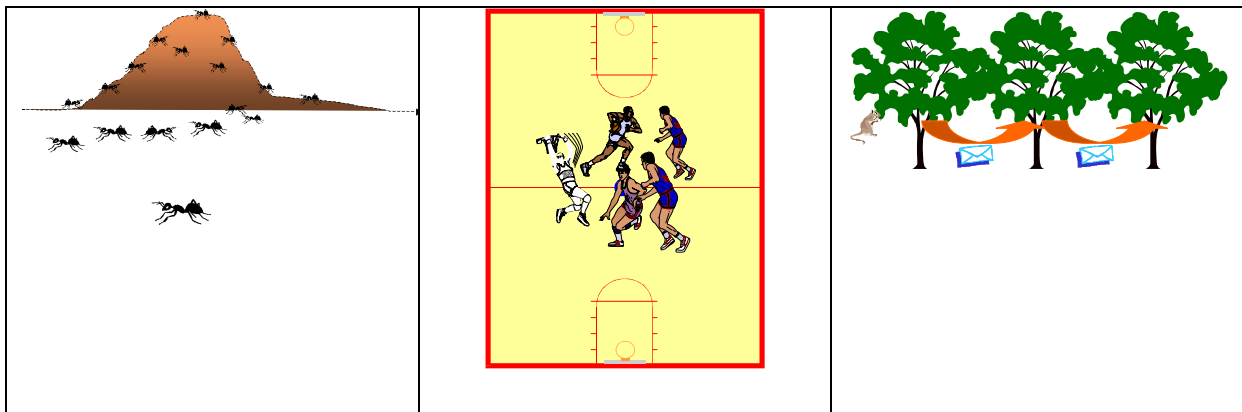


Figure I.14. Exemples de systèmes holoniques

La figure I.14 illustre trois exemples de systèmes holoniques. Si chaque élément est autonome et peut se suffire à lui-même, son but est plus facilement atteint s'il coopère. Dans le premier cas, la fourmi, si elle peut se nourrir seule, ne peut survivre longtemps sans l'aide que peuvent lui apporter ses consœurs. De même, des études ont récemment montré qu'un groupe d'arbres résiste mieux aux attaques de l'environnement qu'un ensemble d'arbres isolés. Le premier arbre attaqué sécrète une substance pour se protéger, qui est détectée par contact

par ses voisins sécrétant alors cette même substance. Enfin, le sport d'équipe est également un exemple qui plaide en faveur de l'importance du collectif vis-à-vis de l'individualisme.

I.3.1.2 Règles relatives la structure arborescente

Le deuxième ensemble de règles est relatif à la nature arborescente de l'organisation holonique.

Règle 2.1 : Les hiérarchies sont dissécables en embranchements, qui les constituent, et dont les holons représentent les nœuds; les ramifications figurent les circuits de communication et de contrôle. *(correspondant à la règle 2.1 de [Koestler 69])*

Règle 2.2 : Le nombre des niveaux que comporte une hiérarchie mesure la "profondeur" de cette hiérarchie, le nombre des holons à n'importe quel niveau donné en exprime "l'envergure". *(correspondant à la règle 2.2 de [Koestler 69])*

Règle 2.3 : On peut considérer les hiérarchies comme des structures "verticalement" arborescentes dont les branches se croisent avec celles d'autres hiérarchies à une multiplicité de niveaux et forment des réseaux "horizontaux" : arborisation et réticulation sont des principes complémentaires de l'architecture des organismes et des sociétés. *(correspondant à la règle 6.1 de [Koestler 69])*

Règle 2.4 : Alors que dans la théorie S-R (stimuli-réponse) les circonstances du milieu déterminent le comportement, dans la théorie SHO (systèmes hiérarchiques ouverts) elles ne font que guider, corriger et stabiliser des systèmes de comportement préexistants. *(correspondant à la règle 6.8 de [Koestler 69])*

Ces règles indiquent qu'une organisation holonique peut être constituée de plusieurs hiérarchies imbriquées (cf. figure I.15), les nœuds étant constitués d'agents holoniques. Il est en effet possible pour un agent d'avoir plusieurs responsables (par exemple dans le cas de l'enseignant-chercheur ayant trois fonctions : d'enseignement, de recherche et d'administration). Cet agent se trouve alors à l'intersection de plusieurs couches. On définit en effet une couche d'agent comme étant l'ensemble des agents supervisés par le même responsable.

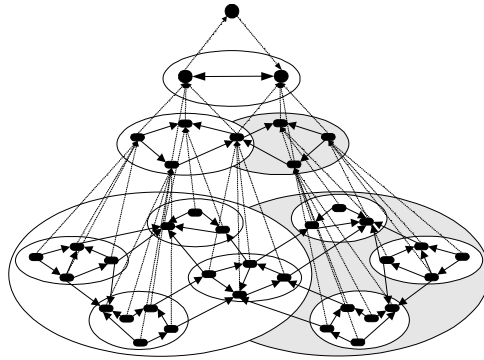


Figure I.15. Hiérarchies imbriquées.

I.3.1.3 Règles relatives aux mécanismes des systèmes holoniques

Le troisième ensemble concerne l'établissement et le respect des règles et des stratégies.

Règle 3.1 : Les holons fonctionnels obéissent à des règles fixes et manifestent des stratégies plus ou moins souples. *(correspondant à la règle 3.1 de [Koestler 69])*

Règle 3.2 : Les règles, appelées canons du système, déterminent les propriétés invariables du système, sa configuration structurelle et (ou) son schème fonctionnel. *(correspondant à la règle 3.2 de [Koestler 69])*

Règle 3.3 : Alors que le canon définit les mouvements possibles dans l'activité du holon, la sélection stratégique de tel ou tel mouvement parmi les choix possibles est guidée par les contingences du milieu. *(correspondant à la règle 3.3 de [Koestler 69])*

Règle 3.4 : Fonctionnellement, le comportement des organismes obéit à des "règles du jeu" qui en expliquent la cohérence, la stabilité et la forme spécifique. *(correspondant à la règle 3.8 de [Koestler 69])*

Règle 3.5 : A mesure que l'on s'élève dans la hiérarchie, les holons manifestent des types d'activité de plus en plus complexes, de plus en plus souples, de moins en moins prévisibles ; à mesure que l'on descend on trouve des activités de plus en plus mécaniques, stéréotypées et prévisibles. *(correspondant à la règle 8.1 de [Koestler 69])*

Règle 3.6 : Toutes les techniques, innées ou acquises, tendent, avec la pratique, à devenir des routines automatiques. On peut décrire ce processus comme une transformation continue d'activités "mentales" en activités "mécaniques". *(correspondant à la règle 8.2 de [Koestler 69])*

Règle 3.7 : Toutes choses égales d'ailleurs, un milieu monotone facilite la mécanisation. *(correspondant à la règle 8.3 de [Koestler 69])*

Règle 3.8 : Inversement, des circonstances nouvelles ou inattendues exigent que les décisions soient renvoyées à des échelons supérieurs, et que les commandes passent des activités "mécaniques" aux activités "attentives". *(correspondant à la règle 8.4 de [Koestler 69])*

Règle 3.9 : Chaque passage des commandes à l'échelon supérieur se reflète par une conscience plus vive et plus précise de l'activité en cours; et puisque la diversité des choix augmente avec la complexité des échelons, chaque passage à l'échelon supérieur s'accompagne de l'expérience subjective de la liberté de décision. (correspondant à la règle 8.5 de [Koestler 69])

Les agents holoniques doivent donc obéir à des règles générales fixées par l'organisation. Mais la manière de répondre à ces règles, c'est-à-dire la stratégie d'un agent est choisie en fonction de son environnement. Et cette stratégie est plus ou moins souple selon la position de l'agent dans l'organisation. En effet, l'agent holonique a un comportement de plus en plus réactif vers le bas de l'organisation et de plus en plus cognitif vers le sommet de la structure.

La figure I.16 présente un exemple de stratégies autour d'un objectif précis : le transport de marchandises de Bruxelles à Biarritz. Chaque routier doit respecter cet objectif tout en étant libre de ses choix quant à la meilleure route à suivre. En cas d'incident non gérable par le routier (panne, manifestation durable par exemple), le responsable du service doit proposer une stratégie adaptée.



Figure I.16. Différentes stratégies répondant à la règle "aller de Bruxelles à Biarritz"

I.3.1.4 Règles relatives aux communications

Le quatrième ensemble de règles fait référence aux communications, d'une part entre les éléments de la holarchie, et d'autre part entre la holarchie et son environnement.

Règle 4.1 : Les hiérarchies d'émission (output) opèrent généralement selon le principe du déclenchement, un signal relativement simple, implicite ou codé, déclenchant des mécanismes complexes préétablis. (correspondant à la règle 5.1 de [Koestler 69])

Règle 4.2 : Dans l'exécution de techniques acquises, techniques verbales incluses, un commandement implicite généralisé est explicité en descendant des échelons successifs, dont l'action, une fois déclenchée, active les unités subordonnées dans l'ordre stratégique approprié, conformément aux indications de rétroaction (feed-back). *(correspondant à la règle 5.5 de [Koestler 69])*

Règle 4.3 : Les hiérarchies d'admission (input) opèrent d'après le principe inverse: au lieu de déclencheurs, elles ont des dispositifs de filtrage (filtres, "résonateurs", classificateurs) qui débarrassent l'information de sa gangue de bruit, en extraient et résumant le contenu pertinent, selon les critères de pertinence de la hiérarchie en question. Les "filtres" opèrent à tous les échelons que le courant d'information doit traverser en montant de la périphérie au centre, dans les hiérarchies sociales comme dans le système nerveux. *(correspondant à la règle 5.8 de [Koestler 69])*

Règle 4.4 : Les hiérarchies d'émission analysent, concrétisent, particularisent. Les hiérarchies d'admission synthétisent, abstraient, généralisent. *(correspondant à la règle 5.11 de [Koestler 69])*

Règle 4.5 : Les échelons supérieurs d'une hiérarchie ne sont pas, normalement, en communication directe avec les échelons inférieurs, et réciproquement; les signaux sont transmis par des "voies de régulation" et à la montée comme à la descente ne franchissent qu'un échelon à la fois. *(correspondant à la règle 7.1 de [Koestler 69])*

Dans une organisation holonique, les messages ne sont possibles qu'entre un agent et son responsable ou entre agents d'une même couche. Les messages deviennent de plus en plus détaillés ou explicités au fur et à mesure qu'ils descendent dans l'organisation et sont filtrés, généralisés au cours de leur montée. La base jouant le rôle d'interface réactive entre l'organisation et l'environnement.

Les figures I.17 et I.18 représentent des exemples de filtrage de l'information. Dans la figure I.17, le corps humain est vu comme un système holonique. L'être humain, par l'intermédiaire du cerveau décide de saisir la tasse ; cette information est répartie au niveau du bras droit, de la main, des différents doigts et des capteurs. Si la température de la tasse est située au-dessus d'un certain seuil, l'information est traduite, rassemblée, filtrée pour parvenir au cerveau sous une forme plus simple : très chaud.

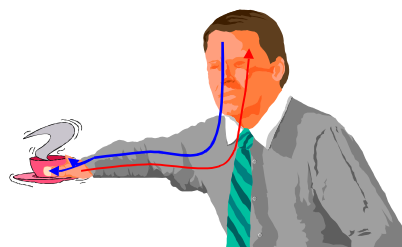


Figure I.17. Filtrage de l'information dans un système holonique

Dans la figure I.18, illustrant une bataille¹⁶, les armées sont considérées comme des systèmes holoniques. Les ordres donnés aux régiments sont décomposés au niveau des sections puis au niveau des équipes pour enfin être interprétés de manière réactive au niveau des soldats. Le retour d'information suit également la holarchie pour être synthétisée en arrivant au poste de commandement.



Figure I.18. Autre exemple de filtrage de l'information dans un système holonique

I.3.1.5 Règles relatives à l'adaptation à l'environnement

Le cinquième ensemble de règles concerne l'adaptation des organisations holoniques à son environnement :

Règle 5.1 : Les organismes vivent d'échanges avec leur milieu. Dans des conditions normales, les tensions provoquées dans les holons impliqués dans l'échange sont transitoires, et l'équilibre est restauré quand l'échange s'accomplit. *(correspondant à la règle 9.3 de [Koestler 69])*

Règle 5.2 : Les règles de conduite d'un holon social ne sont pas réductibles aux règles de conduite de ses membres. *(correspondant à la règle 9.7 de [Koestler 69])*

Règle 5.3 : Le potentiel régénérateur des organismes et des sociétés se manifeste dans des fluctuations qui vont du plus haut niveau d'intégration jusqu'à des niveaux plus anciens et primitifs, pour remonter vers des structures nouvelles. Il semble que de tels processus jouent un grand rôle dans l'évolution biologique et dans l'évolution intellectuelle : les mythes universaux de la mort et de la résurrection en sont les symboles. *(correspondant à la règle 10.2 de [Koestler 69])*

Cette adaptation concerne surtout les notions d'autonomie et de coopération. En fonction de l'environnement, les parties s'individualisent ou coopèrent fortement afin de réduire le déséquilibre causé par ses fluctuations de l'environnement. En fait, dans un système

¹⁶ La bataille de Valmy, Les musées nationaux ©.

holonique, il n'y a pas d'ordre émergeant du chaos, ni d'explosion de l'ordre en chaos. Il s'agit en fait de fluctuations modérées.

La figure I.19 schématise l'ensemble des concepts holoniques. L'organisation est structurée en holarchie par opposition aux hiérarchies, c'est-à-dire avec communication d'objectifs plutôt que d'ordres. Les communications s'effectuent soit selon la holarchie (elles sont alors filtrées lors des passages entre différents niveaux), soit entre agents d'une même couche. Un autre point important est qu'un agent holonique donné possède plus de capacité cognitive que l'agent de niveau inférieur, et est plus réactif que l'agent de niveau supérieur.

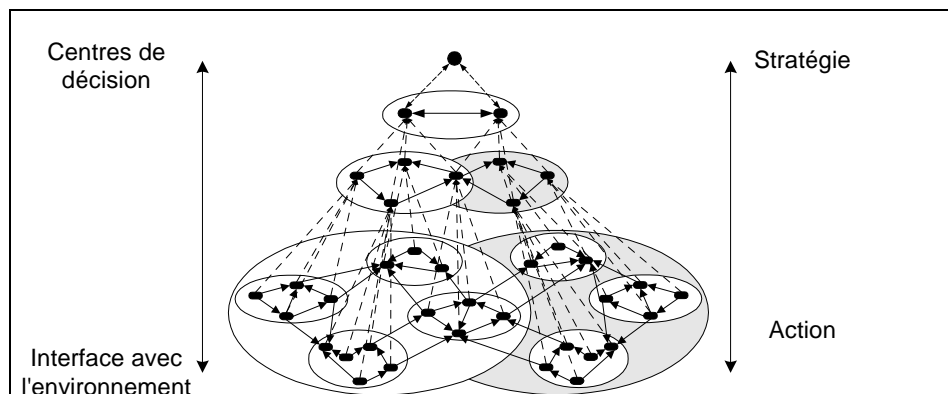


Figure I.19. Organisation holonique sous forme de multi-hiérarchie

Chaque agent holonique peut donc être défini comme un agent cognitif dont les connaissances et les actions varient selon sa position.

Le principe proposant la découpe d'un système holonique en holons, pouvant être eux-mêmes considérés comme des systèmes holoniques, est très intéressant pour la spécification du système global. Il suffit en effet de définir le comportement général du système pour en déduire le comportement général de ses parties.

Cette notion de récursivité sera détaillée dans la partie suivante, qui présente une approche formelle des systèmes multi-agents holoniques, démontrant en particulier qu'un holon peut être à la fois considéré tant comme un agent unique que comme un ensemble d'agents.

I.3.2 Une définition formelle des systèmes multi-agents holoniques

Les définitions formelles proposées dans cette partie permettent en particulier de montrer la récursivité du modèle holonique. Ces définitions ont été proposées par les chercheurs du

DFKI (Centre de Recherche Allemand pour l'Intelligence Artificielle) sur les agents holoniques [Gerber 99].

La première définition concerne la formalisation du monde holonique dans lequel se trouve un système multi-agents. Rappelons que le monde est constitué d'un ensemble d'agents et d'un environnement (cf. figure I.8). Le principe de perception est le suivant : chaque agent holonique reçoit sa perception du monde de la part de celui-ci et peut ensuite choisir une action sur son environnement adaptée à cette perception et à son état. Chaque agent possède donc un ensemble d'états possibles, ainsi qu'un ensemble de perceptions de son environnement et un ensemble d'actions sur cet environnement. Il possède également une fonction de prise de décision associant à son état courant et à sa perception actuelle, une action et un nouvel état.

Définition I-6 :

Un environnement multi-agent est un tuple défini par : $EMA = \{ \mathcal{A}, \mathcal{E}, \Pi, \Delta \}$ où :

$\mathcal{A} = \{ \alpha_1, \dots, \alpha_n \}$ est l'ensemble des agents avec

α_i le tuple défini par $\{ S_i, P_i, A_i, \phi_i \}$ où

- S_i est l'ensemble des états possibles de l'agent α_i ,
- P_i est l'ensemble des perceptions,
- A_i est l'ensemble des actions (sur l'environnement),
- $\phi_i: S_i \times P_i \rightarrow S_i \times A_i$ est sa fonction de prise de décision.

\mathcal{E} est l'ensemble des états d'environnement,

$\Pi: \mathcal{E} \rightarrow (P_1 \times \dots \times P_n)$ est une fonction de perception,

$\Delta: \mathcal{E} \times (A_1 \times \dots \times A_n) \rightarrow \mathcal{E}$ est une fonction de modification de l'environnement.

Pour simplifier le raisonnement et de manière à rester proche de la programmation, le temps est défini par une suite d'événements réguliers. A chaque intervalle de temps :

- chaque agent α_i reçoit, quelque soit l'état de l'environnement $e \in \mathcal{E}$, quelque soit son état $s_i \in S_i$, sa perception locale $p_i = \Pi(e)_i$. A partir de cette perception, l'agent calcule son nouvel état $s'_i = \phi_i(s_i, p_i)_1$ et sa prochaine action $a_i = \phi_i(s_i, p_i)_2$ ¹⁷.
- L'état du monde est modifié, et $e' = \Delta(e, a_1, \dots, a_n)$ est le nouvel état de l'environnement.

Soit $\bar{\Delta} : \mathcal{L} \times S_1 \times \dots \times S_n \rightarrow \mathcal{L} \times S_1 \times \dots \times S_n$ la fonction de transition des états, appliquée à chaque intervalle de temps, définie par $\bar{\Delta}(e, s_1, \dots, s_n) = (e', s'_1, \dots, s'_n)$. Les fonctions ϕ_i de chaque agent ainsi que la fonction Δ de l'EMA font donc partie de $\bar{\Delta}$.

Deux environnements multi-agent holoniques E et E' sont isomorphes s'il existe une bijection faisant correspondre à chaque instant, les états du monde (l'état de l'environnement et les états des agents) des deux EMA.

Définition I-7 :

Deux environnements multi-agents sont isomorphes s'il existe une bijection

$$\Psi : \mathcal{L} \times S_1 \times \dots \times S_n \rightarrow \mathcal{L}' \times S'_1 \times \dots \times S'_m$$

telle que $\forall (e, s_1, \dots, s_n) \in \mathcal{L} \times S_1 \times \dots \times S_n, \quad \bar{\Delta}'(\Psi(e, s_1, \dots, s_n)) = \Psi(\bar{\Delta}(e, s_1, \dots, s_n))$

Ainsi, il peut exister une bijection permettant de relier un environnement multi-agent ne contenant aucun agent (les transitions d'états sont alors toutes définies dans la fonction d'environnement Δ) à un environnement multi-agent contenant tous les agents sauf un (figure I.20), où toutes les transitions sont définies par la fonction ϕ de l'agent unique.

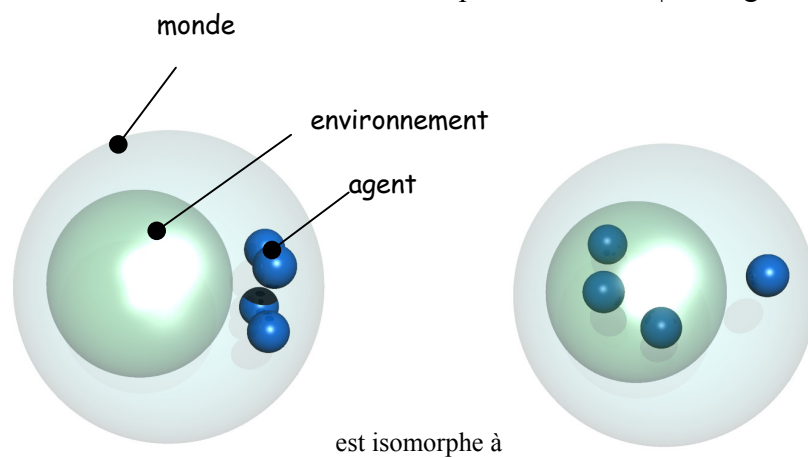


Figure I.20. Equivalence entre deux environnements multi-agent

Le lemme suivant illustre la figure I.20 : chaque agent d'un monde multi-agent peut se percevoir comme l'unique agent du monde, les autres agents faisant partie intégrante de son environnement.

¹⁷ la notation est la suivante : $\Pi(e)_i$ est la $i^{\text{ème}}$ composante de $\Pi(e)$ dans $(P_1 \times \dots \times P_n)$, et $\phi_i(s_i, p_i)_1$ est la première composante de $\phi_i(s_i, p_i)$ dans $S_i \times A_i$.

Lemme 1 :

Si $E = (\{\alpha_1, \dots, \alpha_n\}, \mathcal{L}, \Pi, \Delta)$ est un environnement multi-agent alors
 pour tout $i \leq n$, il existe un environnement multi-agent isomorphe $(\{\alpha_i\}, \mathcal{L}', \Pi', \Delta')$

Le rassemblement des trois agents à droite de la figure I.21 est appelé rassemblement holonique.

Définition I-8 :

Considérons les deux environnements multi-agents isomorphes

$(\{\alpha_1, \dots, \alpha_n\}, \mathcal{L}, \Pi, \Delta)$ et $(\{\alpha_1, \dots, \alpha_{i-1}, \alpha_{i,1}, \dots, \alpha_{i,m}, \alpha_{i+1}, \dots, \alpha_n\}, \mathcal{L}, \Pi', \Delta')$
 α_i est appelé le rassemblement holonique de $(\alpha_{i,1}, \dots, \alpha_{i,m})$.

De façon générale, il est toujours possible d'unir, de rassembler, des agents holoniques en une collection d'agents holoniques pouvant être considérée comme un tout, c'est-à-dire comme un seul agent. Ceci est exprimé par le lemme 2.

Lemme 2 :

Pour tout environnement multi-agent $(\{\alpha_1, \dots, \alpha_n\}, E, \Pi, \Delta)$ et pour tout $k \leq n$, on peut construire un environnement multi-agent $(\{\alpha', \alpha_{k+1}, \dots, \alpha_n\}, E, \Pi', \Delta')$ où α' est un rassemblement holonique de $\alpha_1, \dots, \alpha_k$.

[Gerber 99] note \otimes ¹⁸ cette fonction de rassemblement. Ainsi les deux collections d'agents H_1 et H_2 sont considérées comme des sous-holons dont le rassemblement $(H_1 \otimes H_2)$ devient également un holon.

L'équilibre nécessaire entre les propriétés d'autonomie et de coopération des holons est nécessaire pour la loi de rassemblement. En effet, si une partie est trop autonome, elle ne peut être liée à d'autres parties (la fonction ϕ_i pourrait alors se contredire) et n'est donc pas un holon. De même une partie trop coopérante d'un ensemble ne peut en être extraite et n'est donc pas non plus un holon.

¹⁸ Si I représente le holon unité, (H, I, \otimes) est un monoïde abélien (ou commutatif), c'est-à-dire que :

- la loi interne \otimes est associative : $\forall (a, b, c) \in H^3, (a \otimes b) \otimes c = a \otimes (b \otimes c)$.
- tout élément est régulier : $a \otimes c = b \otimes c \Rightarrow a = b$
- la loi interne \otimes est commutative : $\forall (a, b) \in H^2, (a \otimes b) = (b \otimes a)$.

Par contre (H, I, \otimes) ne peut être un groupe : il n'existe en effet pas d'élément symétrique, c'est-à-dire de holon négatif y tel que pour tout $x \neq I, x \otimes y = I$.

La structure arborescente du système, ainsi que les mécanismes d'action et de contrôle, ne sont pas représentés dans ces définitions. Ils sont en fait inclus dans la fonction de perception Π de l'EMA distribuant aux agents une nouvelle perception du monde, dans les fonctions de prise de décision ϕ_i et dans les actions a_i des agents.

D'après les définitions formelles, un agent ne peut communiquer qu'avec son environnement. Pour qu'il communique avec d'autres agents, il suffit donc que ceux-ci se trouvent dans l'environnement de l'agent en question.

L'adaptation d'un holon à l'environnement, ou à sa perception de l'environnement, est traduite par la fonction de prise de décision ϕ_i de chaque agent.

Ces définitions permettent donc de considérer l'architecture holonique comme étant une architecture récursive, chaque holon ayant les mêmes propriétés que l'ensemble, faisant partie d'un tout, et pouvant être considéré comme un tout composé de parties. Ceci est très intéressant pour la mise en place d'organisations multi-agents, notamment pour la spécification d'une organisation multi-agent ayant pour but la gestion d'information d'une organisation humaine. Nous décrivons dans le chapitre III les propriétés de l'ensemble et donc de chaque agent de l'organisation multi-agent holonique.

Conclusion

L'objectif de ce mémoire est de proposer une modélisation des organisations humaines dans le but de spécifier des systèmes d'information coopératifs proposant une réactivité (au sens du temps de réponse) adéquate, ainsi qu'une cohérence des informations traitées. Ce premier chapitre s'est intéressé à la recherche d'un modèle d'organisation répondant à ces besoins.

Tout d'abord, une structure générale des systèmes d'information distribués a été définie sous la forme de réseaux coopératifs d'entreprise constitués de nœuds reliés par des transitions. Puis, il a été montré que, parmi les structures d'entreprise existantes, celles disposant d'une structure pyramidale (telle que les architectures fonctionnelles ou divisionnelles), sont plus à même de posséder des SID performants. Les autres architectures (telles que les architectures de type marché ou matriciel) entraîneraient la conception de SID centralisés et donc peu réactifs.

Les systèmes multi-agents constituent une solution particulièrement bien adaptée à la mise en place d'un SID réactif et cohérent. Mais pour parvenir à ces deux caractéristiques, le système multi-agents doit être à la fois réactif et cognitif. De tels systèmes existent, mais leurs structures ne correspondent pas aux structures des organisations humaines auxquelles le SID doit être appliqué.

Le problème de trouver l'architecture la plus pertinente pour la mise en place de systèmes d'aide au travail coopératif en entreprise semble donc trouver une réponse grâce aux systèmes holoniques. En effet, cette approche issue d'une réflexion sociologique sur les organisations humaines et les organismes vivants, répond à la définition des réseaux coopératifs d'entreprise présentés au début du chapitre, à savoir un ensemble de nœuds manipulant des informations et reliés entre eux par des flux d'information. Ce modèle permet aussi de traduire les organisations d'entreprise de type divisionnelles ou fonctionnelles, mais non matricielles. Ce type d'organisation étant par définition trop variable. En effet, le modèle holonique, de par son origine, s'applique principalement aux organisations à hiérarchie flexible, c'est-à-dire dont les membres possèdent une certaine autonomie d'actions.

Les systèmes holoniques se prêtent donc très bien à la modélisation des organisations administratives complexes, fonctionnant sur base de flux de données. En effet, ces organisations sont typiquement hiérarchisées, et les nœuds (centres d'actions et de décisions)

y possèdent une relative autonomie et s'échangent des informations le plus souvent au sein d'une même unité et suivant une hiérarchie. Ce modèle nous sert de base à l'analyse et la modélisation de systèmes coopératifs pour la mise en place d'un système d'assistance répondant lui aussi aux principes holoniques. En effet, l'approche multi-agent est particulièrement adaptée à l'assistance pour des opérateurs répartis dans une organisation. Selon l'architecture choisie, le système multi-agent peut disposer de réactions rapides envers l'utilisateur et d'une gestion de la cohérence globale du système d'aide. Des différentes organisations multi-agents présentées, les organisations hiérarchiques sont les plus intéressantes, étant donné qu'elles « collent » plus facilement au système naturel. Ce dernier étant en fait structuré de manière holonique, il est pertinent de construire le système d'assistance selon ce modèle. C'est ce que nous avons proposé en fin de ce chapitre tout en indiquant l'avantage de la conception récursive de l'architecture holonique pour la rapidité de spécification d'un système multi-agent holonique. L'architecture générale du système d'aide au travail coopératif est maintenant posée. Il faut maintenant obtenir les règles d'assistance et donc analyser et modéliser le fonctionnement de l'organisation humaine.

Le chapitre suivant a pour objet la recherche d'une méthode pertinente pour la modélisation de systèmes à caractères holoniques, c'est-à-dire à structure hiérarchique flexible, dont les acteurs possèdent des niveaux de responsabilité dans des procédures coopératives de gestion de données.

Bibliographie du chapitre I

- [**Adam 95**] Adam E., Modèle d'interface réactive dans les systèmes tutoriels de découverte guidée. Mémoire de D.E.A., Université Paul Sabatier, Toulouse, Juin 1995.
- [**Alberto 93**] Alberto T., Combemale P., *Comprendre l'entreprise. Théorie, gestion relations sociales*. Paris : Nathan, 1993.
- [**Alliot 95**] Alliot J.M., Schiex T., *Intelligence Artificielle et Informatique Théorique*. Toulouse : Edition Cépaduès, 1995.
- [**Aloisi 95**] D'Aloisi D., Giannini V., The Info Agent : an interface for supported users in intelligent retrieval. *Proceedings of the Network Information Retrieval - Italia Conference*, (21-23 November 1995 ; Milan, Italia), NIR-IT'95.
- [**Alvares 98**] Alvares L.O., Menezes P.B., Demazeau Y., Problem decomposition: an essential step for multi-agent systems. *Proceedings of the 10th International Conference on Systems Research, Informatics and Cybernetics* (August 1998 ; Baden-Baden, Allemagne), ICSRIC'98.
- [**Arafa 99**] Arafa Y., Charlton P., Fehin P., Mamdani A., Personal Service Assistants with personality. *Human Computer Interaction – Communication, Cooperation and Application Design*, Ed. by Bullinger H.J. & Ziegler J., London : LEA, pp. 147- 151, 1999.
- [**Arai 97**] Arai T., Aiyama Y., Sasaki Y., Holonic Storage: an assembly and storage cell by manipulation using environment. *proceedings of the 29th CIRP International Seminar on Manufacturing System - New Manufacturing Era*, (11-13 May 1997 ; Osaka, Japan), pp.221-226.
- [**Bannon 91**] Bannon L.J., Schmidt K., CSCW: Four characters in Search of a Context. *Studies in CSCW, Theory, Practice and Design*, Ed. by Bowers J.M. & Benford S.D., Nottingham, 1991.
- [**Bensaid 97**] Bensaid N., Mathieu Ph., A Hybrid and Hierarchical Multi-Agent Architecture Model. *Proceedings of the Practical Application of Intelligent Agents and Multi-Agent Technology*, (21-23 April 1997 ; London, U.K.), PAAM'97.
- [**Bensaid 99**] Bensaid N., Contribution à la réalisation d'une modèle d'architecture multi-agent hiérarchique. Thèse de doctorat, Université des Sciences et Technologies de Lille, France, Mai 1999.
- [**Berge 70**] Berge C., *Grappe et hyper-graphes*. Paris : Dunod, 1970.
- [**Boissier 94**] Boissier O., Demazeau Y., An Architecture for Social and Individual Control and its Application to Computer Vision. *Proceedings of the Modelling Autonomous Agents in a Multi-Agent World conference*, (2-4 August 1994 ; Odense, Denmark), MAAMAW.

-
- [Bonabeau 99]** Bonabeau E., Dorigo E., Theraulaz G., L'intelligence en essaim. Ingénierie des Systèmes Multi-Agent (Actes des Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents), (Novembre 1999 ; Saint Gilles), JFIADSMA'99, Ed. par Marie-Pierre Gleizes, Paris : Hermes, 1999.
- [Bourjot 99]** Bourjot C., Chevrier V., Bernard A., Krafft B., Coordination par le biais de l'environnement, une approche biologique. *Ingénierie des Systèmes Multi-Agent (Actes des Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents)*, (Novembre 1999 ; Saint Gilles), JFIADSMA'99, Ed. par Marie-Pierre Gleizes, Paris : Hermes, 1999.
- [Briffaut 98]** Briffaut X., Guichard N., Pierre D., MERCURE : une plate-forme multi-agents cognitifs communiquant par KQML, appliquée à la communication d'entreprise. *Proceedings of the complex systems, intelligent systems & interfaces conference*, (25-27 Mai 1998 ; Nîmes), *NÎMES'98*.
- [Brooks 91]** Brooks R. A., Intelligence Without Reason. *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, (Août 1991 ; Sydney, Australie), IJCAI'91, Ed. by Myopoulos J.& Reiter R., pp. 569-595.
- [Chaib-Draa 92]** Chaib-Draa B., Moulin B., Mandiau R., Millot P., Trends in Distributed Artificial Intelligence. *AI Review*, 1992, n°4, pp.35-66.
- [Chaib-draa 96]** Chaib-draa B., Interaction between agents in routine, familiar and unfamiliar situations. *International Journal of Intelligent and Cooperative Information Systems*, 1996, 1, 5, pp. 7-20.
- [De Michelis 94]** De Michelis G., From the analysis of cooperation within work-processes to the design of CSCW Systems. *Proceedings of the 15th Interdisciplinary Workshop on Informatics and Psychology: Interdisciplinary approaches to system analysis and design*, (24-26 May 1994, Schaerding, Austria).
- [Demazeau 95]** Demazeau Y., From Interactions To Collective Behaviour in Agent-Based Systems. *Proceedings of the first European Conference on Cognitive Science*, Saint-Malo, France, Avril 1995.
- [Dignum 99]** Dignum F., Are Information Agent just an extension of information systems or a new paradigme ? *Proceedings of Autonomous Agents Conference*, (1-5 May 1999 ; Seattle, USA), Agents'99.
- [Etzioni 94]** Etzioni O., Weld D., An Softbot-based interface to the Internet. *Communication of the ACM*, 1994, 37, 7, pp. 72-76.
- [Ferber 95]** Ferber J., Les systèmes multi-agents, vers une intelligence collective. IIA, Paris : InterEditions, 1995
- [Ferber 97]** Ferber J., Gutknecht O., Aladin: a meta-model for the analysis and design of organization in multi-agent systems. Rapport de recherche LIRMM 97189, Laboratoire d'Informatique, de Mécanique et de Micro-électronique de Montpellier, Décembre 1997.

-
- [Flores-Mendez 99]** Flores-Mendez R.A., Towards the Standardization of Multi-Agent System Architectures: An Overview. In ACM Crossroads. *Special Issue on Intelligent Agents*, Association for Computer Machinery, 5.4, pp.18-24, 1999.
- [Gasser 89]** Gasser L., Research Notes in Artificial Intelligence. *Distributed Artificial Intelligence, volume I.* Ed. by Huns. Londres : Pitman, 1989.
- [Gasser 92]** Gasser L., An overview of DAI. *Distributed Artificial Intelligence: Theory and Praxis.* Ed. by Avouris & Gasser, pp. 9-30. Dordrecht, Netherlands : Kluwer Academic Publishers, 1992.
- [Gerber 99]** Gerber C., Siekmann J., Vierke G., Holonic Multi-Agent Systems. *Rapport de Recherche RR-99-03*, DFKI (Deutsche Forschungszentrum für Künstliche Intelligenz) GmbH, Mars 1999.
- [Goodman 86]** Goodman GO, Abel M.J., Collaboration research in SCL. *Proceedings of the Conference on Computer-Supported Cooperative Work*, (3-5 Décembre 1986 ; Austin, USA), CSCW'86, Ed. by Peterson D.
- [Handl 99]** Handl D., Hoffmann H-J., Workflow agents in the document-centred communication in MALL2000 systems. *Proceedings of Autonomous Agents Conference*, (1-5 Mai 1999 ; Seattle, USA), Agents'99.
- [Hofstadter 85]** Hofstadter D., *Gödel, Escher, Bach, Les brins d'une guirlande éternelle.* Paris : InterEdition, 1985.
- [Ito 98]** Ito N., Nakagawa K., Hotta T., Du X., Ishii N., EAMMO : an environmental agent model for multiple objects. *Information and software technology*, 40, 1998, pp. 397-404.
- [Kitano 98]** Kitano H., Asada M., Kuniyoshi Y., Noda I., Osawa E., Matsubara H., RoboCup: A Challenge Problem for AI and Robotics. *RoboCup-97: Robot Soccer World Cup I*, Ed. Springer-Verlag, pp.1-19, 1998.
- [Koda 96]** Koda T., Agents with Faces: A Study on the Effects of Personification of Software Agents. M.S. Thesis, Massachusetts Institute of Technology, 1996.
- [Koestler 69]** Koestler A., *The Ghost in the Machine.* London : Arkana Books, 1969.
- [Kuflik 00]** Kuflik T., Shoal P., User Profile Generation for Intelligent Information Agents - Research in Progress. *Proceedings of the Conference on Advanced Information Systems Engineering*, (5-6 juin 2000 ; Stockholm, Suède), CaiSE'00.
- [Laird 97]** Laird J. E., Pearson D. J., Huffman S. B., Knowledge-directed adaptation in multi-level agents. *Journal of Intelligent Information systems.* 9, 3, pp. 261-275, 1997.
- [Lant 94]** Lant T. K., Computer Simulations of Organizations as Experiential Learning Systems: Implications for Organization Theory. *In Computational Organization Theory.* Ed. by Carley K. M. & Prietula M.J., London : LEA publishers, pp. 195-215, 1994.

-
- [Lashkari 94]** Lashkari Y., Metral M., Maes P., Collaborative Interface Agents. *Proceedings of the Twelfth National Conference on Artificial Intelligence*. (1-4 August 1994 ; Seattle, USA), AAAI '94, AAAI Press.
- [Laurel 97]** Laurel B., Interface Agents : Metaphors with Character. *Software Agent*, Ed. by J. M. Bradshaw, AAAI Press, 1997.
- [Le Strugeon 95]** Le Strugeon E., Une méthodologie d'auto-adaptation d'un système multi-agents cognitifs. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, France, 17 janvier 1995.
- [Lieberman 97]** Lieberman H., Autonomous Interface Agents. *Proceedings of the Conference on Human Factors in Computing Systems*, (22-27 Mars 1997 ; Atlanta, USA), CHI'97, Ed. by Pemberton S., New-York, USA : ACM Press, pp. 67-74.
- [Mandiau 93]** Mandiau R., Contribution à la modélisation des univers multi-agents : génération d'un plan partagé. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, 22 février 1993.
- [Mandiau 99]** Mandiau R., Le Strugeon E., Agimont G., Study of the influence of organizational structure on the efficiency of a multi-agent system. *Networking and Information Systems Journal*, 2, 1999, pp.153-179.
- [Marshak 95]** Marshak R. T., Workflow : Applying Automation to Group Processes. *Groupware, Technology and Applications*, Ed. by Coleman D. & Khanna R., Prentice Hall, pp. 71-95, 1995.
- [Minar 98]** Minar N., Kramer K., Maes P., Cooperating Mobile Agents for Dynamic Network Routing. In *Software Agents for Future Communications Systems*, Ed. by Hayzelden A., Bigham J., Londres : Springer-Verlag, 1998.
- [Nwana 96]** Nwana Hyacinth S., Software Agent : an overview. *Knowledge Engineering Review*, 11, 3, 1996, pp. 205-244.
- [O'Brien 98]** O'Brien P.D., Wiegand M.E., Agent based process management : applying intelligent agents to workflow. *Knowledge Engineering Review*, 13, 2, 1998.
- [Ocelllo 98]** Ocelllo M., Demazeau Y., Baeijs C., Designing Organized Agents for Cooperation with real Time Constraints. *Proceedings of the first international workshop on collective robotics*, (4-5 July 1998 ; Paris), CRW'98, ed. by Drogoul A., Tambe M. & Fukuda T., New York, USA : Springer Verlag., pp. 25-37.
- [Odgers 99]** Odgers B.R., Thompson S.G., Shepherdson J.W., Cui Z., Judge D.W., O'Brien P. D., Technologies for Intelligent Workflows: Experiences and Lessons. *Proceedings of Agent-Based Systems in the Business Context*, (18-22 juillet 1999 ; Orlando, Californie), AAAI 1999, AAAI Press, pp. 63-67.
- [Odubiyi 97]** Odubiyi J.B., Kocur D.J., Weinstein S. M., Wakim N., Srivastava S., Gokey C., Graham J., SAIRE - A scalable agent-based information retrieval engine. *Proceedings of the Autonomous Agent conference*, (5-8 Février 1997 ;), Agents'97, Ed. by Müller J., New York : ACM Press, p. 292-299.

- [Quintero 98]** Quintero Carillo Ramos A. C., Ucros M., Multi-Agent System Coordination Model applied to the administrative Process of Systems Auditory in an Organization. *Proceedings of the 4th International Conference on information Systems Analysis and Synthesis*, (12-16 July 1998 ; Orlando, USA), ISAS'98.
- [Rhodes 00]** Rhodes J., Just-In-Time Information Retrieval. Ph.D. Thesis, MIT Media Lab, May 2000.
- [Rich 97]** Rich C., Candace L., Sidner (Lotus Development Corporation) *Proceedings of the ninth ACM Symposium on User Interface Software and Technology*, (November 1997 ; Seattle, USA), pp. 21-30.
- [Rich 83]** Rich E., *Artificial Intelligence*. New York : Mac Graw Hill, 1983.
- [Rus 97]** Rus D., Gray R., Kotz D., Transportable Information Agent. *Journal of Intelligent Information systems*. 9, 3, pp. 215-238, 1997.
- [Schael 97]** Schael T., Théorie et pratique du workflow. Des processus métiers renouvelés. New-York : Springer Edition, 1997.
- [Simon 95]** Simon H. A., Explaining the ineffable : AI on the Topics of Intuition, Insight and Inspiration. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, (20-25 Auhust 1995 ; Montréal, Canada), IJCAI'95, Ed. by Dean T., Morgan Kaufman Publishers, pp. 939-948.
- [Soulié 98]** Soulié JC., Marsenac P., Calderoni S., Courdier R., GEAMAS V2.0 : an Object Oriented Platform for Complex systems Simulations. *Proceedings of Technology of Object-Oriented Languages and Systems conférence*, (28 Août 1998 ; Santa-Barbara, USA), TOOLS'98 , IEEE Computer Society.
- [Tranvouez 99]** Tranvouez E., Espinasse B., Protocoles de coopération pour le réordonnement d'atelier. Ingénierie des Systèmes Multi-Agent (Actes des Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents), (Novembre 1999 ; Saint Gilles), JFIADSMA'99, Ed. par Marie-Pierre Gleizes, Paris : Hermes, 1999.
- [Turing 50]** Turing A. M., Computing Machinery and Intelligence. *Journal of the Mind Association*. LIX, 236, pp. 433-460, Oxford , UK :Oxford University Press, 1950.
- [Van Brussel 96]** Van Brussel H., Bongaerts L., Wyns J., Valckenaers P., Van Ginderachter T., A Conceptual Framework for Holonic Manufacturing Systems: Identification of Manufacturing Holons. *Journal of Manufacturing Systems*, 18, 1, 1999, pp. 35-52.
- [Van Mulken 99]** Van Mulken S., André E., Müller J., An empirical Study on the Trustworthiness of Life-Like Interface Agents. *Human Computer Interaction – Communication, Cooperation and Application Design, Vol 2*. Ed. by Bullinger H-J. & Ziegler J., Londres : LEA., pp. 152-156, 1999.
- [WFMC 99]** Terminology and Glossary. *Workflow Management Coalition*. Document numéro WFMC-TC-1011, 3.0, February 1999.

-
- [White 97] White J. E., Mobile Agents. In *Software Agent*. Ed. by J. M. Bradshaw, AAAI Press, Menlo Park, USA, 1997.
- [Wilson 94] Wilson P., Introducing CSCW, What it is and why we need it. *CSCW : The multimedia and networking paradigm*, UNICOM, ed. by S.A.R. Scrivener, Aldershot, England: Avebury Technical, pp. 1-18, 1994.
- [Wyns 99] Wyns J., Reference architecture for holonic manufacturing systems - the key to support evolution and reconfiguration. Ph.D. Thesis, K.U. Leuven (Université Catholique de Louvain), Belgique 1999.
- [Young 95] Young M. J., Human Error and the holon cognitive architecture. *Proceedings of the 6th Symposium on analysis, design and evaluation of Man-Machine systems*, Massachusetts Institute of Technology, Cambridge, USA : MIT-Press, 1995.
- [Yu 99] Yu L., A Conceptual Framework For Agent Oriented and Role Based Workflow Modeling. *Proceedings of the Conference on Advanced Information Systems Engineering*, (14-15 june 1999; Heidelberg, Germany), CaiSE'99.

Chapitre II -

Etude comparative de méthodes du génie logiciel dans un but de modélisation des Organisations Multi-Agents

INTRODUCTION	62
II.1 BESOINS EN ANALYSE ET MODELISATION D'ORGANISATIONS MULTI-AGENTS DEDIEES AUX SYSTEMES D'INFORMATIONS DISTRIBUES	63
II.2 CHOIX D'UN CADRE DE COMPARAISON DE METHODES	65
II.3 UTILISATION DU CADRE DE COMPARAISON POUR L'EVALUATION DE SEPT METHODES REPRESENTATIVES.....	72
CONCLUSION	86
BIBLIOGRAPHIE DU CHAPITRE II	88

Introduction

Afin d'obtenir les informations nécessaires au fonctionnement d'un Système d'Information Distribué (S.I.D.) coopératif au sein d'une organisation humaine (c'est-à-dire en particulier les règles de fonctionnement de cette organisation ainsi que les données manipulées), l'organisation doit être analysée et modélisée à l'aide d'une méthode adéquate. Notre objectif est que cette méthode soit adaptée aux organisations humaines à structure pyramidale et dont les acteurs possèdent une relative autonomie.

Ce chapitre a pour objectif de nous orienter vers le choix d'une telle méthode, ou, le cas échéant, de faciliter la construction d'une nouvelle méthode plus pertinente à partir de méthodes existantes.

Les méthodes existantes en Génie Logiciel sont souvent destinées à un domaine particulier et/ou à une organisation particulière (par exemple, la construction de base de données, la programmation objet). Dans le cas de besoins particuliers, il devient nécessaire de les adapter.

Pour pouvoir choisir et sélectionner de façon rigoureuse la ou les méthodes adéquates les plus pertinentes, il faut pouvoir disposer d'un ensemble de critères de sélection, c'est-à-dire d'un cadre comparatif (également appelé benchmark) qui permette de les confronter.

La première partie de ce chapitre présente, selon un point de vue du Génie Logiciel, les besoins en analyse et modélisation d'organisations humaines de type holonique nécessaires à la spécification de systèmes d'informations distribués.

La seconde partie propose un cadre de comparaison (benchmark), composé de 75 critères, destiné à faciliter le choix ou la construction d'une méthode d'analyse et de modélisation d'organisation humaine de type holonique, dans le but de mettre en place un système d'information distribué.

La troisième partie présente une application du cadre de comparaisons sur six méthodes d'origines et d'objectifs différents, généralement utilisées pour l'implémentation de systèmes informatiques : MERISE, OMT (ainsi que les extensions d'UML), SADT, OSSAD, CISAD et MKSM. Une synthèse de cette confrontation de méthodes est ensuite proposée en regard de la méthode idéale qui devrait permettre d'analyser et de modéliser les organisations humaines de type holonique dans un but de spécification de SID coopératifs.

II.1 Besoins en analyse et modélisation d'organisations multi-agents dédiées aux systèmes d'informations distribués

Chaque année, des dizaines de logiciels interactifs sont développés dans les entreprises. Dans le meilleur des cas, l'équipe de développement se base sur une méthode rigoureuse de développement issue du Génie Logiciel. Mais bien souvent, elle se base sur une méthode « maison », issue d'une adaptation d'une ancienne méthode (telle que SADT par exemple) [Iivari 98]. Il est alors fréquent de constater que les systèmes interactifs développés posent de nombreux problèmes d'utilisabilité, ne répondent pas toujours aux besoins des utilisateurs, et sont souvent mal adaptés à l'organisation du travail [Helander 97] [Kolski 97] [Lim 94] [Nielsen 93].

Ceci est dû à une inadéquation entre les méthodes utilisées et l'objectif visé. Cette nécessité d'adapter les méthodes provient du fait qu'il n'existe pas de méthode unique de modélisation et de spécification de systèmes informatiques [Jackson 95]. Les développeurs qui ressentent un manque lors de l'application de « leur » méthode à une nouvelle situation tentent alors de l'améliorer selon leurs propres critères. Il en résulte alors un foisonnement de méthodes personnelles manquant souvent de cohérence sur certaines phases du cycle de développement.

L'intégration de méthodes est une solution à la recherche d'une méthode, rigoureuse et cohérente, adaptée à un problème. « L'intégration de méthodes est la combinaison de deux ou plusieurs méthodes dans le but de former une nouvelle méthode plus efficace que chacune des méthodes séparées » [Paide 99]. Il existe plusieurs cas d'intégration possibles (dans le but d'obtenir une notation plus adaptée, pour avoir un cycle de développement plus complet, etc.). C'est pourquoi, il importe d'objectiver ces choix et de proposer un cadre cohérent de comparaison [Adam 00a].

Dans notre cas, ce cadre doit permettre la sélection ou la création d'une méthode d'analyse et de modélisation d'organisation holonique. Pour rappel, la méthode doit permettre l'analyse et la modélisation d'organisations à structure pyramidale, dont les membres s'échangent des informations, possèdent une relative autonomie, sont responsables et/ou travaillent sous la responsabilité d'autres membres.

Elle doit s'appliquer de façon pertinente aux organisations holoniques, et perturber au minimum les membres de ces organisations. La méthodologie, la mise en œuvre de la méthode, est par conséquent très importante et doit faire l'objet d'une étude critique détaillée.

L'environnement d'application de la méthode et l'angle sous lequel elle considère une organisation (par exemple comme une boîte noire ou un système ouvert, figure II.1), doivent être analysés en détail.



Figure II.1. Exemple de perception d'organisation par une méthode
 (a) selon une boîte noire
 (b) selon un système ouvert

Ces figures schématisent deux façons de percevoir les organisations selon le point de vue des méthodes. Certaines ne s'intéressent qu'aux entrées-sorties et considèrent l'organisation comme une boîte noire ; pour d'autres, l'organisation est un système ouvert et les flux de données internes sont considérés.

Chaque méthode utilise généralement ses propres formalismes adaptés au problème auquel elle s'attaque (par exemple la représentation des objets pour les méthodes OMT ou UML, ou encore la représentation des actions à entreprendre selon les situations pour la méthode MKSM). Les formalismes sont en quelque sorte la clé de voûte des méthodes. Sans formalismes adaptés aux problèmes (permettant de représenter toute l'étendue des problèmes) et aux utilisateurs des méthodes (adoptant une représentation aisément compréhensible), elles risquent de ne pas être appliquées.

Dans les problèmes qui nous intéressent, la coopération et les relations de responsabilité sont essentielles. Il importe donc d'analyser sinon de représenter ou du moins d'analyser les communications, les coordinations d'actions et les collaborations entre les acteurs des organisations étudiées par les méthodes.

Le cadre de comparaison vise la sélection d'une méthode devant permettre la mise en place d'un système d'information distribué. C'est pourquoi il est intéressant de noter quels types de logiciels et surtout quels types d'interfaces homme-machine peuvent être réalisés à partir des méthodes confrontées.

Il importe donc d'analyser la méthode autour des notions de méthodologie, d'environnement, de représentation, de prise en compte de la coopération et de technologie. La partie suivante décrit ces *dimensions* de comparaison, ainsi que les critères retenus pour apprécier l'intérêt des diverses méthodes selon ces cinq dimensions.

II.2 Choix d'un cadre de comparaison de méthodes

Dans cette partie sont résumés les critères, composant les dimensions, utilisés pour la confrontation des méthodes. Les développements qui vont suivre sont fortement inspirés de D. Pascot et C. Bernardas, ils ont été adaptés et complétés pour répondre à notre problématique : l'analyse et la conception de systèmes interactifs dédiés aux systèmes administratifs complexes, où la coopération joue un rôle important. Ces auteurs, dans leur article [Pascot 93], avaient proposé un cadre de référence, sorte de méta-modèle, permettant de comparer des méthodes de conception de systèmes d'information informatisés.

Il est à noter qu'il existe encore peu de comparaisons de méthode, la plupart d'entre elles portent sur des sujets précis (par exemple [Chang 97] pour la confrontation de méthodes dans le but de conception d'interfaces homme-machine) ou sur des méthodes de même type (par exemple [Henderson-Sellers 99] pour la comparaison de Open et de UML).

Le cadre de référence proposé dans ce chapitre est, rappelons le, composé de 5 dimensions : *représentation*, *environnement*, *méthodologie*, *technologie* et *coopération*, qui sont successivement détaillées.

II.2.1 La dimension représentation

L'utilisation d'une méthode étant est fortement dépendante de ses formalismes et de sa façon de représenter le problème auquel elle est dédiée.

Cette représentation peut être *directe* dans le cas où la représentation des données issues de l'analyse est neutre (par exemple : les actigrammes de la méthode SADT transcrivent directement les activités identifiées lors de l'analyse). Mais elle peut être également *interprétée*. Dans ce cas, les données de l'analyse sont interprétées par les auteurs lors de la modélisation (par exemple : transformation sous forme objet).

Généralement, les méthodes construisent leurs représentations autour de trois axes : la représentation des *données*, la représentation des flux de données (appelée modèle *d'activités*)

et la représentation du *traitement* des données. Certaines méthodes utilisent également une représentation de la dynamique fournissant plus de détails et permettant de passer à une phase de simulation du système étudié. Ce modèle est particulièrement important pour la représentation de la dynamique inhérente aux systèmes coopératifs (tels que le parallélisme, les interruptions, les boucles). L'ordre dans lequel sont appliqués ces modèles est variable selon les méthodes. Il est essentiel que ces différents modèles forment un ensemble cohérent. Malheureusement, peu de méthodes proposent un recoupement entre leurs modèles.

Outre ces différents modèles, les méthodes peuvent également décomposer le système auquel elles s'appliquent selon des *niveaux d'abstraction* (séparation en niveaux conceptuel, organisationnel et physique), selon un axe *généralisation - spécialisation* (lorsqu'une organisation est étudiée, on peut partir du comportement des personnes interrogées, puis abstraire la personne en l'associant à un rôle, puis à un groupe de travail, dans le cas d'une approche ascendante), selon un découpage *type - occurrence* (appelé aussi classe-instanciation dans les principales méthodes orientées objet) ou selon une orientation *stratégie - tactique* (dans le cas d'un département, il s'agit d'étudier ses stratégies à long terme, le pourquoi de l'activité, ainsi que les tactiques, le comment).

Le nombre de formalismes utilisés est variable selon les méthodes. Certaines réutilisent les mêmes formalismes pour les différentes parties du système, d'autres suggèrent d'utiliser de nouveaux formalismes lors de la modélisation de nouveaux concepts (par exemple, pour la modélisation du niveau conceptuel et du niveau organisationnel).

La dimension représentation est donc associée à quatre critères :

La position de l'analyse (façon dont les auteurs de la méthode "attaquent" le système, en terme de modélisation) : pouvant être soit *interprétée*, soit *directe*.

Le principe de construction : les sous critères *données*, *traitements*, *activité dynamique* indique l'ordre dans lequel ils sont traités.

Le découpage du système : indiquant si le système étudié est décomposé selon les *niveaux d'abstraction*, une *généralisation - spécialisation*, un découpage *type-occurrence* ou selon une orientation *stratégie - tactique*.

Les formalismes (schémas, concepts et règles) : ce critère énumère les formalismes utilisés pour la représentation des *données*, des *activités*, des *traitements* et de la *dynamique* utilisés par la méthode.

II.2.2 La dimension environnement d'application

La représentation d'un système coopératif ou d'une organisation humaine est dépendante de chaque méthode. Ainsi, certaines d'entre elles considèrent les organisations comme une *boîte noire* (c'est-à-dire ne considérant que les entrées et sorties, sans se soucier de ce qui se passe à l'intérieur), ou traduisent les organisations *hiérarchiques*, *communautaires* (où chaque acteur a le même niveau de responsabilité), *holoniques*, de type *système cybernétique*¹ (l'organisation est connue par ses échanges et ses actions sur les données) ou encore de type *système ouvert finalisé* (l'organisation est considérée comme un système d'échange et les stratégies du système sont étudiées ; il existe aussi à ce sujet la dénomination de *système cybernétique ouvert*).

L'environnement du système à étudier influe sur la réussite de la méthode. Ainsi, certaines méthodes sont sensibles au *degré de structuration* (dans certaines entreprises, les fonctions de certaines personnes ne sont pas cernées clairement), à la *stabilité* du système (une entreprise en pleine restructuration pourra être considérée selon ce critère comme instable) et/ou à la *certitude* des informations recueillies lors de l'analyse (une analyse menée dans une entreprise où les tâches ne sont pas précisément réparties conduit à des incertitudes concernant les données recueillies).

La manière de retranscrire une organisation ou un système est également perçue au niveau des caractéristiques des données manipulées par les méthodes. En effet, une méthode considérant le système comme une boîte noire n'utilisera que des informations données sur la *quantité* d'informations entrantes et sortantes, tandis qu'une méthode considérant le système comme un système ouvert s'intéressera à la *qualité* des données (sa structure). Au niveau d'analyse le plus élevé, les méthodes utilisent des informations quant à la *pertinence* des données manipulées, c'est-à-dire quant à leurs adéquations avec les activités modélisées.

Ainsi, la dimension environnement est associée à trois critères :

¹ Selon le dictionnaire Larousse, la cybernétique est l'étude des processus de commande et de communication chez les êtres vivants, dans les machines et les systèmes sociologiques et économiques.

Images d'organisation : ce critère est composé des sous critères *hiérarchie, communauté, système holonique, boîte noire, système cybernétique* (l'organisation est connue par ses échanges et ses actions sur les données), *système cybernétique ouvert* (les stratégies de l'organisation sont étudiées).

Nature de l'environnement : ce critère est composé des sous critères *degré de structuration, stabilité et certitude*.

Caractéristiques des données : ce critère est composé des sous critères *qualité, quantité, pertinence*.

II.2.3 La dimension méthodologie

La mise en œuvre d'une méthode, sa méthodologie, décrit la façon dont ses différentes phases s'enchaînent. Cette méthodologie doit être adaptée au domaine auquel la méthode est destinée. Ainsi, pour modéliser une organisation humaine, il est préférable, pour des raisons de disponibilité des acteurs concernés, de procéder par phases successives (comme dans le modèle en cascade par exemple), plutôt que par essai-erreur (cf. le modèle en spirale). En effet, le modèle en spirale nécessiterait de consulter, à chaque nouveau prototype, l'ensemble des acteurs de l'organisation, ce qui, dans le cadre d'une organisation composée de plusieurs dizaines de personnes serait très coûteux en temps.

Les principales phases² suivies par les méthodes sont l'*analyse* de l'existant, la *modélisation*, la *spécification*, la *conception* (englobant la conception architecturale et la conception détaillée au sens du génie logiciel) et la *validation technique*. Mais il est important de considérer d'autres phases importantes au sens des facteurs humains telles que les *évaluations ergonomiques a priori* (avant la réalisation logicielle) de la partie interactive, et les *évaluations ergonomiques a posteriori* (après la réalisation logicielle) [Balbo 94] [Grislin 96].

Les principaux modèles utilisés par les méthodes pour enchaîner ces phases sont le modèle en cascade *cascade*, en *spirale* (au sens de Boehm [Boehm 88]), en *V*, le modèle *évolutif* (dit aussi *incrémental*), et le modèle ∇ (prononcer *nabla*) [Kolski 97] [Kolski 98]

² Nous omettons délibérément la phase de réalisation qui concerne plus l'utilisation de langages ou de logiciels spécialisés et ne faisant pas l'objet de ce mémoire. Cet aspect est traité globalement dans la dimension technologie.

proposé récemment par C. Kolski (cf. annexe A), et dédié à la conception et à l'évaluation de systèmes interactifs.

Dans le cas de systèmes complexes à étudier, les méthodes procèdent par couches successives, soit par une approche *descendante* (ou top-down : l'organisation est d'abord étudiée dans son ensemble, puis la méthode traite les cas particuliers), *ascendante* (ou bottom-up : la méthode 'part' des différents postes ou tâches, puis remonte progressivement vers des ensembles tels que les groupes de travail ou les procédures) ou une approche *évolutive* (middle-out : cette approche consiste à partir des groupes de travail pour ensuite descendre au niveau des postes ou remonter au niveau de l'organisation).

Enfin, dans le cadre d'une modélisation d'organisation humaine dans le but de spécifier un système d'aide au travail coopératif, il paraît essentiel d'inclure le futur utilisateur³ dans ces phases (ce qui n'est pourtant pas le cas dans la plupart des méthodes étudiées).

La dimension méthodologie est donc associée à cinq critères :

Cycle de développement : composé des sous critères *cascade*, *spirale*, *en V*, *évolutif* et *∇ (nabla)*.

Phases concernées : *Analyse*, *Modélisation*, *Spécification*, *Conception*, *Validation technique*, *Evaluation ergonomique a priori* et *Evaluation ergonomique a posteriori*.

Approche de développement : pouvant être *descendante*, *ascendante* ou *évolutive*.

Degré d'implication de l'utilisateur : ce degré va de *pas d'implication* à *implication essentielle*.

Moment d'implication de l'utilisateur : celui-ci se situe au *début*, *au milieu et/ou en fin de cycle*.

II.2.4 La dimension coopération

La coopération au sein d'une organisation a été définie dans le chapitre 1 comme la communication, la coordination d'actions et la collaboration entre les acteurs de l'organisation.

³ Le terme utilisateur est vu au sens générique du terme [Barthet 88][Kolski 97]. Il concerne dans notre cas les acteurs humains composant une organisation.

Si les flux de données peuvent être représentés par les modèles d'activités, de traitements et/ou de dynamique, les informations sur le *mode de communication* y apparaissent rarement. Afin de mieux représenter la communication entre acteurs, les méthodes doivent indiquer si les communications suivent un chemin *direct* (par envoi de message) ou *indirect* (par tableau noir), et si elles s'effectuent de manière *synchrone* (par exemple par téléphone) ou asynchrone (par exemple : la messagerie). Ces informations peuvent se trouver dans un *modèle* de communication, le plus souvent basé sur des théories de la linguistique, permettant de formaliser les communications.

La *négociation* entre acteurs met en jeu des mécanismes de communication et de collaboration. Elle est une partie essentielle de la coopération et devient courante dans de nombreuses entreprises qui se tournent de plus en plus vers un management par projets. Dans le cadre d'une recherche de méthode applicable aux organisations humaines complexes où entrent en jeu des mécanismes de coopération, il convient donc de rechercher les méthodes prenant en compte cette négociation.

Les acteurs de ces organisations complexes sont liés entre eux par des relations de *hiérarchie* et/ou de *responsabilité*. Ces relations influent sur leurs coopérations et doivent être représentées.

A l'opposé de la coopération se trouve *l'autonomie*. Ces deux notions sont indispensables et font partie des trois caractéristiques nécessaires (avec la stabilité) à l'obtention d'un système holonique (cf. I.3). Dans cette dimension sera noté également le fait qu'une méthode puisse prendre en compte les problèmes de *confiance* que pose le travail coopératif [Thimbleby 94].

La dimension coopération est donc associée à quatre critères :

Communication : composé du sous-critère *mode de communication* (celui-ci peut prendre les valeurs : *direct, indirect, synchrone, asynchrone*) ; ainsi que des sous-critères *modèle* et *négociation*.

Relations : regroupant les sous-critères *hiérarchie* et *responsabilité*,

Coordination : indiquant si la méthode permet de représenter les coordinations d'actions,

Individualité : associé à *l'autonomie* et à la *confiance*.

II.2.5 La dimension technologie

La finalité de la méthode recherchée est la spécification d'un système d'information distribué utilisant les technologies issues du domaine du multi-agent. Cette spécification doit ensuite mener à une réalisation.

Les méthodes devront donc être également comparées sous l'angle technologique, c'est-à-dire que les types de programmation auxquels mènent les méthodes seront étudiés : à savoir, si cette programmation doit être *structurée*, orientée *base de données* (que ces bases soient orientées objets ou non), orientée *objet*, orientée *agent* (ou acteur, au sens de l'Intelligence Artificielle Distribuée).

De même, une méthode prédisposée à la conception d'interfaces évoluées, telles que les interfaces *adaptables* (ou flexibles, au sens de Williges [Williges 87]), *adaptatives* (au sens de Schneider-Hufschmidt et al. [Schneider-Hufschmidt 93], une interface de ce type vise à s'adapter aux besoins des utilisateurs), ou *assistantes* (au sens de Boy [Boy 91], l'interface raisonne en parallèle à l'utilisateur et se comporte comme un assistant humain), sera préférée par rapport à une méthode proposant la conception d'une interface *classique*.

De manière plus générale, les modes de traitement (*batch*, *interactif*, *client-serveur*, *synchrone*, *asynchrone*, *distribué*) qu'il est possible de réaliser à partir des méthodes seront également notés.

La dimension technologie est alors associée à trois critères :

Programmation : *structurée*, orientée *base de données*, orientée *objet*, orientée *agent*.

Interface homme-machine : *classique*, *adaptable*, *adaptative*, *assistante*,

Mode de traitement : *batch*, *interactif*, *client-serveur*, *synchrone*, *asynchrone* et/ou *distribué*.

C'est donc par ces cinq dimensions (méthodologie, organisation, représentation, coopération et technologie) que vont être comparées les méthodes. La confrontation d'une méthode à l'ensemble de ces critères permet d'obtenir une fiche d'identité de cette méthode. L'ensemble de ces fiches facilite ainsi la comparaison de méthodes, soit pour la sélection de la plus pertinente, soit pour la conception d'une nouvelle méthode reprenant les parties essentielles des méthodes testées.

La partie suivante montre l'utilisation du cadre de comparaison sur un ensemble de méthodes représentatives de leurs domaines d'application.

II.3 Utilisation du cadre de comparaison pour l'évaluation de sept méthodes représentatives

Il est important d'étudier et de comparer les différentes classes de méthodes disponibles et de voir dans quelle mesure celles-ci sont à même de modéliser une organisation humaine en vue d'une spécification d'un système d'information distribué.

Dans ce but, sept méthodes, représentatives de cinq domaines d'application différents, ont été testées. Ces domaines concernent particulièrement la mise en place de systèmes de bases de données, la conception objet, la représentation structurée, la modélisation d'organisation et la gestion des connaissances. Ces méthodes sont :

- MERISE [Tardieu 85] [Tardieu 91] : représentative des méthodes systémiques (au sens de Le Moigne [Le Moigne 77]), et dont la finalité est l'analyse et la conception de systèmes d'information (reposant sur une base de données).
- OMT (Object Modeling Technique ; [Rumbaugh 91]) : certainement une des plus représentative des méthodes d'analyse et de conception orientées objets utilisée dans les entreprises avant l'apparition de UML (cf. ci-dessous). Bien entendu, d'autres méthodes bien connues auraient pu également être considérées, telles les méthodes proposées par Booch [Booch 94], Coad et Yourdon [Coad 91] ou Jacobson [Jacobson 93].
- UML (Unified Modeling Language; [Rumbaugh 98]) : nouvelle méthode de modélisation et de conception unifiant (pour la représentation) les modèles présents dans les principales méthodes orientées objets [Perrin 97] [Muller 97]. Une comparaison sommaire avec OMT permettra de constater si UML est mieux adaptée à la modélisation d'organisations humaines qu'OMT.
- SADT (Structured Analysis and Design Technique ; [IGL 89]) : inspirée des concepts proposés par Ross [Ross 77], et représentative des méthodes dites cartésiennes ou structurées, telle SA [DeMarco 79], orientées vers les phases d'analyse et de spécification.
- OSSAD (Office Support System Analysis and Design; [Dumas 90]) : méthode d'analyse et de spécification de systèmes d'information centrée sur l'organisation du travail.

- CISAD (Cooperative Information System Analysis and Design ; [Nurcan 96]) : méthode basée sur OSSAD, visant l'analyse et la conception d'applications coopératives (particulièrement de type Workflow) ; celle-ci accorde donc beaucoup d'importance aux notions de coopération et de communication entre acteurs.
- MKSM (Methodology for Knowledge Systems Management ; [Ermine 95]) : méthode de gestion de la connaissance dans l'entreprise. Nous avons ajouté cette méthode non classique (et plus orientée sur une approche de l'Intelligence Artificielle) dans la mesure où celle-ci peut également être source d'idées pour l'analyse et la conception de systèmes interactifs.

Ces méthodes ont donc chacune été analysées à l'aide du cadre de comparaison. Ces analyses sont successivement décrites. Les modèles proposés par ces méthodes sont détaillés en annexe B.

II.3.1 La méthode MERISE

Le tableau II.1 montre les différents aspects pris en compte par la méthode MERISE et permet de mettre en évidence les enseignements ci-dessous.

Dans MERISE, la représentation a une très grande importance. D'ailleurs, les formalismes y sont essentiels, même si la méthode est quelquefois injustement résumée par certains développeurs à ses formalismes principaux : le Modèle Conceptuel des Données, et à un degré moindre le Modèle Organisationnel des Traitements. Le découpage en niveaux d'abstraction est très marqué. En général, on retrouve, dans les différentes versions de MERISE, les niveaux conceptuel, organisationnel et physique. Le système est d'abord abordé par les flux de communication, puis les données et les traitements sont analysés en parallèle pour ensuite être confrontés. En principe, les moyens fournis pourraient apporter un plus pour la collaboration entre concepteurs et utilisateurs (cf. à ce sujet les travaux de Barthelet [Barthelet 88] [Barthelet 95], étendus par Tarby [Tarby 93]).

MERISE considère l'organisation par ses échanges et ses actions (par le MCD et le MOT). Cette méthode est destinée aux systèmes structurés, stables et certains, ce qui n'est pas le cas dans différents domaines d'application (ou entreprises). De plus, sa finalité orientée base de données permet d'avoir une information qualitative et quantitative sur les données traitées dans l'organisation.

Tableau II.1. Analyse de la méthode MERISE selon les cinq dimensions

Environnement		Méthodologie			
Images d'organisation	Boîte noire	Cycle de Développement	Cascade	✓	
	Hiérarchie		Spirale		
	Communauté		V		
	Système cybernétique		Evolutif		
	Système cybernétique ouvert		∇		
Nature de l'environnement	Système holonique	Phases Concernées (hors réalisation)	Analyse	✓	
	Structuré		Modélisation	✓	
	Semi-structuré		Spécification	✓	
	Non structuré		Conception	✓	
	Stable		Validation Technique	✓	
	Instable		Evaluation Ergonomique a priori		
Certain	Evaluation Ergonomique a posteriori				
Caractéristiques des données	Incertain	Approche	Descendante	✓	
	Qualitative		Ascendante		
	Quantitative		Evolutive		
Technologie	Pertinence	Degré d'implication de l'utilisateur	Pas		
	Mode de Traitement		Peu		
			Batch	Moyen	✓
			Interactif	Beaucoup	
			Client-serveur	Essentiel	
Type d'Interface Homme Machine	Synchrone	Moment d'implication de l'utilisateur	Début	✓	
	Asynchrone		Milieu		
	Distribué		Fin		
Programmation	Classique	Représentation			
	Adaptable	Position de l'analyse	Interprétée		
	Adaptative		Directe	✓	
Coopération	Assistante	Principe de construction (ordre d'apparition des formalismes)	Données	①	
	Communication		Activités	①	
			de données	Traitements	②
			Modes de Communication	Dynamique	
Relations	Modèle du Langage	Principe d'assemblage	Niveaux d'abstraction	✓	
	Négociation		Généralisation / Spécification	✓	
	Coordination		Type / Occurrence	✓	
			Stratégique / Tactique		
Individualité	Confiance	Formalisme	Données	M.C.D., M.O.D., M.L.D.	
			Autonomie	Activités	M.C.C.
	Individuelle		Traitements	M.C.T., M.O.T., M.P.T.	
				Dynamique	

M.C.C. : Modèle Conceptuel des Communications ; M.C.D. : Modèle Conceptuel des Données ;

M.O.T. : Modèle Organisationnel des Traitements ; M.O.D. : Modèle Organisationnel des Données ;

M.L.D. : Modèle Logique des Données ; M.P.T. : Modèle Physique des Traitements.

Le cycle de vie total du projet, supporte le développement en cascade et possède une approche descendante. L'utilisateur est moyennement impliqué dans le projet et sa contribution se situe surtout au début, au niveau des modèles conceptuels et organisationnels. Soulignons à nouveau les travaux de Barthet [Barthet 95] visant à mieux intégrer la prise en compte de l'utilisateur dans la méthode MERISE, hélas trop peu connus des utilisateurs de cette méthode.

Au niveau technologique, MERISE est destinée à la conception de bases de données qui peuvent supporter un mode de traitement batch, interactif, synchrone et client-serveur (suivant la version de MERISE). Elle peut aboutir à une programmation structurée, mais surtout à la

conception d'une base de données éventuellement orientée objet (grâce aux extensions actuelles de MERISE; cf. par exemple [Morejon 94]).

En ce qui concerne la coopération, MERISE permet de représenter la communication de données et la coordination (par son modèle de traitement), mais ne va pas plus loin dans ce domaine.

II.3.2 La méthode OMT

Le tableau II.2 montre les différents aspects pris en compte par la méthode OMT.

Pour OMT, les données ont bien sûr une très grande importance et le formalisme des données est très complet (il permet en particulier de représenter l'héritage, l'agrégation et les contraintes). Au niveau des formalismes, OMT est intéressante car elle n'en utilise que trois (le modèle Objet, le Diagramme d'Etat adapté et le Diagramme des Flots de Données) pour le cycle de vie du projet qui se déroule en quatre phases : l'analyse, la conception du système, la conception des données et l'implémentation. Cependant, on ne retrouve pas de modèle suggérant un rapprochement entre concepteurs et utilisateurs (comme par exemple le modèle Organisationnel des Traitements de la méthode MERISE), pour la prise en compte des besoins des utilisateurs.

Le système étudié (le terme 'organisation' n'apparaît pas vraiment dans OMT, ce qui constitue à notre sens une limitation) est connu par ses échanges et ses actions, mais par sa conception objet, OMT peut s'appliquer à des systèmes semi-structurés et instables dans une mesure prévisible. Par principe, l'approche objet permet d'obtenir toutes les informations nécessaires sur les données.

Tableau II.2. Analyse de la méthode OMT selon les cinq dimensions

Environnement		Méthodologie				
Images d'organisation	Boîte noire		Cycle de Développement	Cascade		
	Hierarchie			Spirale	✓	
	Communauté			V		
	Système cybernétique	✓		Evolutif		
	Système cybernétique ouvert			∇		
Nature de l'environnement	Système holonique		Phases Concernées (hors réalisation)	Analyse	✓	
	Structuré	✓		Modélisation	✓	
	Semi-structuré	✓		Spécification	✓	
	Non structuré			Conception	✓	
	Stable	✓		Validation Technique	✓	
	Instable	✓		Evaluation Ergonomique a priori		
Certain	✓	Evaluation Ergonomique a posteriori				
Caractéristiques des données	Incertain		Approche	Descendante	✓	
	Qualitative	✓		Ascendante		
	Quantitative	✓		Evolutive		
Technologie	Pertinence	✓	Degré d'implication de l'utilisateur	Pas		
	Mode de traitement	Batch		✓	Peu	✓
		Interactif		✓	Moyen	
		Client-serveur		✓	Beaucoup	
		Synchrone		✓	Essentiel	
Asynchrone		✓	Moment d'implication de l'utilisateur	Début	✓	
Distribué	✓	Milieu				
Type d'Interface Homme Machine	Classique	✓	Représentation			
	Adaptable	✓	Position de l'analyse	Interprétée	✓	
	Adaptative			Directe		
Programmation	Assistante		Principe de construction (ordre d'apparition des formalismes)	Données	①	
	Structurée	✓		Activités	③	
	Base de données			Traitements		
	Objet	✓		Dynamique	②	
Coopération	Agents ou acteurs		Principe d'assemblage	Niveaux d'abstraction	✓	
	Communication	de données		✓	Généralisation / Spécification	✓
		Modes de Communication			Type / Occurrence	✓
		Modèle du Langage			Stratégique / Tactique	
Négociation			Formalisme	Données	M.O.	
Relations	Hierarchie	Activités		D.F.D.		
Coordination	Responsabilité		Traitements			
		✓	Dynamique	Diag. Evts, D.E. évolué		
Individualité						
	Confiance					
	Autonomie	✓				

M.O. : Modèle Objet ; D.F.D. : Diagramme de Flux de Données ;

Diag. Evs : Diagramme d'événements ; D.E. évolué : Diagramme d'état évolué

OMT couvre presque entièrement le cycle de vie du projet (mises à part les évaluations ergonomiques a priori et a posteriori). Le cycle de développement utilisé est le cycle en spirale (privilegié pour le prototypage), mais celui-ci est uniquement centré sur la modélisation de l'application, et son raffinement successif. Son approche est également descendante, l'utilisateur n'est impliqué qu'au tout début du projet, dans la phase d'analyse. On trouve dans la plupart des méthodes orientées objets de telles limitations : l'utilisateur y est quasi inexistant.

En ce qui concerne la technologie, OMT est une méthode orientée objet. Elle permet de concevoir des programmes qui peuvent être interactifs, basés sur une architecture client-serveur, synchrones et/ou distribués. La programmation peut bien sûr être orientée objet, mais

aussi être structurée comme l'expliquent les auteurs de la méthode. Les concepts objets permettent en principe au concepteur de rendre l'interface adaptable (c'est-à-dire configurable par les utilisateurs, cf. par exemple Visual Basic ou Visual C++).

Au niveau de la coopération, OMT ne permet que la représentation des flux de données d'une organisation. Mais si l'on choisit de modéliser les acteurs du système alors la représentation de leur éventuelle autonomie devient possible.

II.3.3 Les extensions d'UML

UML est définie comme un langage d'unification des différentes méthodes objet. Ce n'est pas encore une méthode, c'est pourquoi cette analyse ne s'intéresse qu'à la dimension représentation (tableau II.3). On pourra éventuellement se reporter à la méthode OMT pour les autres aspects méthodologiques.

Tableau II.3. Analyse de la méthode UML selon la dimension représentation (pour les autres dimensions, cf. tableau II.2)

Dimension Représentation		
Position de l'analyse	Interprétée	✓
	Directe	
Principe de construction (ordre d'apparition des formalismes)	Données	① ③
	Activités	② ④
	Traitements	⑥
	Dynamique	⑤
Principe d'assemblage	Niveaux d'abstraction	✓
	Généralisation / Spécification	✓
	Type / Occurrence	✓
	Stratégique / Tactique	
Formalisme	Données	DC, DO, Dcol
	Activités	CU, Dseq.
	Traitements	DA
	Dynamique	D.E. évolué

DC : Diagramme de Classe ; **DO** : Diagramme d'Objet ; **Dcol** : Diagramme de Collaboration ; **CU** : Cas d'Utilisation ; **DSeq** : Diagramme de Séquence ; **DA** : Diagramme d'Activités ; **DE** : Diagramme d'Etat.

UML provient d'une fusion des principales modélisations objets. En cela, elle reprend et améliore leurs formalismes. Pour ce qui est des données, UML simplifie l'écriture du modèle de données d'OMT. Mais UML utilise deux fois plus de modèles qu'OMT, ce qui peut être déroutant, même si ces modèles sont basés sur des formalismes semblables (comme les diagrammes de classe et les diagrammes d'objets par exemple). Les diagrammes utilisés sont : le diagramme de classe, le diagramme d'objet, le diagramme de collaboration, les cas d'utilisation, les diagrammes de séquences, le diagramme d'activité et les diagrammes d'état. L'emploi des cas d'utilisation de Jacobson incite à intégrer ou tout au moins à prendre en compte l'utilisateur, ce qui constitue un progrès très important par rapport à OMT en particulier.

II.3.4 La méthode SADT

Le tableau II.4 montre les différents aspects pris en compte par la méthode SADT.

Tableau II.4. Analyse de la méthode SADT selon les cinq dimensions

Environnement		Méthodologie			
Images d'organisation	Boîte noire	Cycle de Développement	Cascade	✓	
	Hierarchie		Spirale		
	Communauté		∇		
	Système cybernétique		Evolutif		
	Système cybernétique ouvert		∇		
Nature de l'environnement	Système holonique	Phases Concernées (hors réalisation)	Analyse	✓	
	Structuré		Modélisation	✓	
	Semi-structuré		Spécification	✓	
	Non structuré		Conception		
	Stable		Validation Technique		
	Instable		Evaluation Ergonomique a priori		
Certain	Evaluation Ergonomique a posteriori				
Caractéristiques des données	Incertain	Approche	Descendante	✓	
	Qualitative		Ascendante		
	Quantitative		Evolutionne		
Technologie	Pertinence	Degré d'implication de l'utilisateur	Pas		
	Mode de Traitement		Batch	Peu	
			Interactif	Moyen	✓
			Client-serveur	Beaucoup	
	Type d'Interface Homme Machine		Synchrone	Essentiel	
			Asynchrone	Moment d'implication de l'utilisateur	Début
Distribué		Milieu	✓		
	Fin	-			
Programmation	Classique	Représentation			
	Adaptable		Position de l'analyse	Interprétée	
	Adaptative		Directe	✓	
Coopération	Assistante	Principe de construction (ordre d'apparition des formalismes)	Données	⊙	
	Structurée		Activités	⊙	
	Base de données		Traitements		
	Objet		Dynamique		
Communication	Agents ou acteurs	Principe d'assemblage	Niveaux d'abstraction		
	de données		Généralisation / Spécification	✓	
	Modes de Communication		Type / Occurrence		
	Modèle du Langage		Stratégique / Tactique		
Relations	Négociation	Formalisme	Données	Datagramme	
	Hierarchie		Activités	Actigramme	
	Responsabilité		Traitements		
Coordination			Dynamique		
Individualité					
	Confiance				
	Autonomie				

SADT permet l'analyse et la modélisation de systèmes à l'aide de deux formalismes : les datagrammes et les actigrammes. Cette méthode est très utilisée pour décrire de façon structurée des problèmes et leurs solutions. Elle comprend plusieurs concepts dont celui de dualité activités-données. Ce qui entraîne un test de correspondance entre les deux modèles par une analyse de liens Activités/Données et de liens Données/Activités. Elle procède par une démarche descendante (Top-Down) de généralisation/spécification. Notons que les formalismes de la méthode SADT ont été exploités par Abed et Angue [Abed 94] et Benaissa [Benaissa 93] pour la modélisation statique des tâches et/ou des activités humaines, et

complémentée par les réseaux de Petri pour la composante dynamique de ces tâches et activités.

La méthode s'applique à l'analyse fonctionnelle de systèmes et donc aux systèmes cybernétiques. De par son analyse descendante, hiérarchique et structurée, elle ne peut s'appliquer qu'à des environnements stables, certains et structurés. Seul l'aspect qualitatif des données est pris en compte. Un aspect important de la méthode est qu'elle suggère une analyse prenant en compte le point de vue des intervenants humains.

SADT facilite surtout l'analyse et la modélisation du système (même si ses auteurs suggèrent d'aller jusqu'à sa validation) et suit le cycle en cascade. Son approche est bien sûr descendante, et l'utilisateur est surtout impliqué lors de cette phase d'analyse.

En principe, SADT n'est pas destinée à la conception d'applications, au sens de la conception architecturale et de la conception détaillée (même si le 'D' de cette méthode signifie "Design"). A partir d'une analyse SADT, il est possible de concevoir des applications à l'aide d'une programmation structurée.

En ce qui concerne la coopération, SADT ne permet d'indiquer que les flux de données possibles entre les acteurs du système, même si certains formalismes basés sur le formalisme IDEF-0 (on parle de SADT-like) pourrait représenter plus d'informations (tel que les niveaux de hiérarchie). Nous reviendrons sur ce point dans le chapitre suivant.

II.3.5 La méthode OSSAD

Le tableau II.5 montre les différents aspects pris en compte par la méthode OSSAD.

OSSAD se compose de trois modèles : le modèle Abstrait qui comprend 2 graphes (A1 : relations entre fonctions ou activités, A2 : matrice Activité/Rôle), le modèle descriptif qui comprend 5 graphes (D1 : relations entre rôles, D2 : relations entre tâches, D3 : diagramme d'une tâche (1 rôle), D4 : diagramme d'une procédure (plusieurs rôles), D5 : description détaillée d'une opération). Cette méthode ne possède pas véritablement de modèle de données mais propose un concept de fiches descriptives permettant de recenser les ressources, activités, tâches et acteurs du système étudié, en y incluant les relations d'appartenance ou de subordination. En principe, les méthodes de représentation disponibles dans cette méthode sont centrées sur l'analyse du travail humain.

Tableau II.5. Analyse de la méthode OSSAD selon les cinq dimensions

Environnement		Méthodologie		
Images d'organisation	Boîte noire	Cycle de Développement	Cascade	
	Hiérarchie		Spirale	
	Communauté		V	
	Système cybernétique		Evolutif	
	Système cybernétique ouvert		∇	
Nature de l'environnement	Système holonique	Phases Concernées (hors réalisation)	Analyse	
	Structuré		Modélisation	
	Semi-structuré		Spécification	
	Non structuré		Conception	
	Stable		Validation Technique	
	Instable		Evaluation Ergonomique a priori	
Caractéristiques des données	Certain	Approche	Evaluation Ergonomique a posteriori	
	Incertain		Descendante	
	Qualitative		Ascendante	
Technologie	Quantitative	Degré d'implication de l'utilisateur	Evolutive	
	Pertinence		Pas	
	Mode de traitement		Batch	Peu
			Interactif	Moyen
			Client-serveur	Beaucoup
Synchrone		Essentiel		
Type d'Interface Homme Machine	Asynchrone	Moment d'implication de l'utilisateur	Début	
	Distribué		Milieu	
	Classique		Fin	
	Adaptable		Représentation	
Programmation	Adaptative	Principe de construction (ordre d'apparition des formalismes)	Interprétée	
	Assistante		Directe	
	Structurée		Données	
	Base de données		Activités	
Coopération	Objet	Principe d'assemblage	Traitements	
	Agents ou acteurs		Dynamique	
	de données		Niveaux d'abstraction	
	Modes de Communication		Généralisation / Spécification	
Relations	Modèle du Langage	Formalisme	Type / Occurrence	
	Négociation		Stratégique / Tactique	
	Hiérarchie		Données	
Coordination	Responsabilité	Fiches OSSAD	Activités	
	Confiance		Diag. A1, A2, D1, D2	
Individualité	Autonomie	Traitements	Diag. D3, D4, D5	
		Dynamique		

Fiches : Les fiches permettent de décrire les ressources , les rôles, les acteurs, les unités, les tâches, les opérations, les procédures et les outils.

A1 : relations (de données) entre fonctions et sous-fonctions ; A2 : Matrice Activité / Rôle ; D1 : Relations (de données) entre Rôles ;

D2 : Relations (de données) entre rôles et tâches ; D3 : Diagramme d'une tâche (liée à 1 rôle), basé sur les RdP ;

D4 : Diagramme d'une procédure incluant plusieurs rôles (M.O.T. "like" ajoutant la notion de coopération) ;

D5 : Détail d'une opération ou d'une tâche (basé sur les actigrammes)

Le symbole – indique un critère pris en compte mais non traité par la méthode

OSSAD, destinée aux systèmes d'information, voit l'organisation uniquement comme un système cybernétique ouvert. Son modèle abstrait (représentant ce qui doit être fait et pourquoi) nécessite un système stable et certain. Son modèle descriptif (représentant qui fait quoi et comment) n'utilise pas une représentation formelle. La méthode peut donc s'appliquer à un environnement semi-structuré. Au travers de ses diagrammes, elle fournit des informations relatives à la pertinence et la qualité des données.

OSSAD a pour but la réorganisation et sa simulation. Elle s'intéresse à l'analyse et à la spécification et laisse le choix à d'autres méthodes (MERISE, SADT par exemple) pour ce qui est de l'implémentation. Elle utilise le cycle de vie en V et suit une démarche descendante. L'avis de l'utilisateur est essentiel pour l'analyse du système homme-machine.

OSSAD n'est pas destinée à la conception d'applications, au sens de la conception architecturale et de la conception détaillée (même si son nom le laisse supposer, au même titre que SADT). A partir d'une analyse OSSAD, il est possible de concevoir des applications à l'aide d'une programmation structurée. Par sa modélisation des rôles et ressources par fiches OSSAD, la méthode peut aboutir en plus à une programmation orientée objet.

OSSAD (ainsi que CISAD présentée dans la partie suivante), est très intéressante pour la modélisation de la coopération. OSSAD permet, comme toutes les méthodes, de représenter les flux de données d'une organisation, mais elle permet aussi de représenter la coordination (ou synchronisation) au sein de l'organisation, à l'aide du modèle des traitements. Pour ce qui est de la représentation de l'autonomie des acteurs, elle est également possible par l'utilisation des fiches acteurs.

II.3.6 La méthode CISAD

CISAD est une extension de OSSAD. Sa contribution porte surtout sur l'aspect coopération en proposant l'utilisation de modèles de communication, ainsi que des modèles permettant de représenter les degrés de responsabilités. C'est pourquoi seules les dimensions Représentation et Coopération sont abordées ici et figurent dans le tableau II.6. On pourra éventuellement se reporter à la méthode OSSAD pour les autres aspects méthodologiques.

CISAD est une méthode d'analyse et de conception d'applications de type workflow basée sur la méthode OSSAD. Elle en reprend les modèles auxquels elle ajoute des composantes permettant de représenter la communication entre le personnel coopérant d'un système. Ces composantes sont : la matrice de Grudin [Grudin 94] pour la modélisation des communications ; le modèle d'argumentation de la méthode IBIS qui représente le raisonnement par une représentation "Question - Options - Critères" ; un modèle de conversation (d'après le modèle orienté Coordination-Action).

Le modèle de rôle D1 est adapté afin de représenter la responsabilité des rôles dans les activités. De même que OSSAD, cette méthode ne dispose pas réellement d'un modèle de

données, mais décrit en plus les liens de communications et de responsabilités. CISAD propose donc encore plus de moyens que OSSAD, pour faciliter la prise en compte des facteurs humains.

Tableau II.6. Analyse de la méthode CISAD selon les dimensions Représentation et Coopération (pour les autres dimensions, cf. tableau II.5)

Dimension Représentation			Dimension Coopération		
Position de l'analyse	Interprétée		Communi- Cation	De données	✓
	Directe	✓		Modes de communication	Synchrone Asynchrone
Principe de Construction (ordre d'apparition des formalismes)	Données	③		Modèle du langage	Mc Ma
	Activités	①		Négociation	✓
	Traitements	②		Relations	Hiérarchie
	Dynamique		Responsabilité		✓
Principe D'assemblage	Niveaux d'abstraction	✓	Coordination	✓	
	Généralisation / Spécification	✓	Individualité	Confiance	
	Type / Occurrence			Autonomie	✓
	Stratégique / Tactique				
Formalisme	Données	Fiches OSSAD			
	Activités	Diag. A1, A2, D1', D2, Ma, Mc			
	Traitements	Diag. D3, D4, D5			
	Dynamique				

MdAr : Modèle d'argumentation pour représenter un raisonnement sous la forme d'une question entraînant la proposition d'options sur base de critères ; *MdCv* : Modèle de conversation sur la base du modèle Action [Medina-Mora 92] ;

D1' : Modèle de rôle de OSSAD modifié, intégrant la chronologie et la représentation du niveau de responsabilité.

CISAD reprend les points forts de OSSAD et les étend, notamment en prenant en compte la notion de responsabilité. CISAD offre la possibilité de définir le type de communications inter-acteurs et de les détailler à l'aide des modèles de conversation et d'argumentation ; cette modélisation permet de représenter les phases de négociation entre acteurs. Elles forment de plus en plus les structures de communication dans les organisations actuelles, qui ont, rappelons-le, de plus en plus une structure par projet plutôt que pyramidale.

II.3.7 La méthode MKSM

MKSM découpe le système selon trois axes : la syntaxe, la pragmatique et la sémantique. Chacun de ces axes est partagé en trois points de vue : la structure, la fonction et l'évolution. Cette méthode commence d'abord par identifier tous les processus possibles et leurs cas d'apparition par le modèle du domaine, puis le modèle d'activité décrit les flux de données par actigrammes. Les données sont ensuite représentées par le modèle des concepts, et enfin les tâches sont représentées par le modèle des tâches. La méthode ne vise pas la modélisation d'organisation humaine. Ses formalismes sont surtout utiles pour le recensement et la description des activités individuelles des acteurs, mais font difficilement apparaître les liens entre ces acteurs.

Le tableau II.7 montre les différents aspects pris en compte par la méthode MKSM.

Tableau II.7. Analyse de la méthode MKSM selon les cinq dimensions

Environnement		Méthodologie			
Images d'organisation	Boîte noire	Cycle de Développement	Cascade		
	Hierarchie		Spirale		
	Communauté		V	✓	
	Système cybernétique		Evolutif		
	Système cybernétique ouvert		Nabla		
	Système holonique	Phases	Analyse	✓	
Nature de l'environnement	Structuré	Concernées (hors réalisation)	Modélisation	✓	
	Semi-structuré		Spécification	OMT	
	Non structuré		Conception		
	Stable		Validation Technique		
	Instable		Evaluation Ergonomique a priori		
	Certain		Evaluation Ergonomique a posteriori		
	Incertain	Approche	Descendante	✓	
Caractéristiques des données	Qualitative		Ascendante		
	Quantitative		Evolutive		
	Pertinence	✓	Degré d'implication de l'utilisateur	Pas	
Technologie				Peu	
Mode de Traitement	Batch		Moyen	✓	
	Interactif	-	Beaucoup		
	Client-serveur	-	Essentiel		
	Synchrone	-	Moment d'implication de l'utilisateur	Début	✓
	Asynchrone	-		Milieu	-
	Distribué	-		Fin	-
Type d'Interface Homme Machine	Classique	-	Représentation		
	Adaptable	-	Position de l'analyse	Interprétée	✓
	Adaptative	-		Directe	
Programmation	Assistante		Principe de construction (ordre d'apparition des formalismes)	Données	③
	Structurée	✓		Activités	②
	Base de données			Traitements	① ④
	Objet			Dynamique	
Coopération	Agents ou acteurs		Principe d'assemblage	Niveaux d'abstraction	✓
	Communication	de données		Généralisation / Spécification	✓
		Modes de Communication		Type / Occurrence	
		Modèle du Langage		Stratégique / Tactique	
Relations		Négociation	Formalisme	Données	MdD, MdC
		Hierarchie		Activités	MdA
		Responsabilité		Traitements	MdT
Coordination				Dynamique	
Individualité					
		Confiance			
		Autonomie			

MdD : Modèle du Domaine, sur base de modèles SCFC (source-cible-flux-champs), permettant de recenser les processus par systèmes et par situation en fonction d'événements déclencheurs,

MdA : Modèle d'Activité, basé sur les actigrammes ; **MdC** : Modèle des Concepts, pour la représentation arborescente des ressources,

MdT : Modèle des Tâches, déclinant pour un système source les tâches en activités.

Le symbole – indique un critère pris en compte mais non traité par la méthode

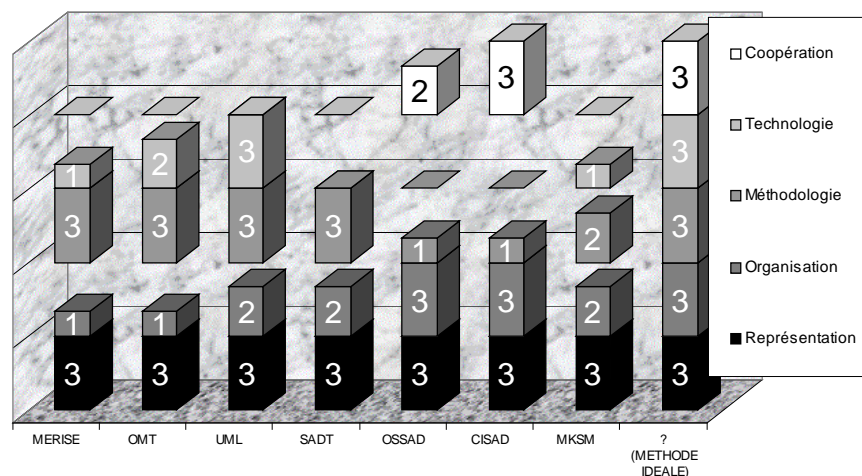
MKSM, présentée comme une méthode de gestion des connaissances, s'applique aux systèmes cybernétiques ouverts. Cette méthode permet de modéliser le patrimoine de connaissances par raffinements successifs et ne nécessite pas un environnement structuré. Par contre, son analyse de processus et d'activités nécessite un environnement stable et certain. Son modèle de processus et son modèle de concepts permettent d'avoir également des informations sur la pertinence et la qualité des données.

MKSM ne permet que l'analyse et la modélisation du système. La spécification et la conception sont réalisées en s'appuyant sur la méthode OMT. MKSM utilise le cycle de développement en V et une approche descendante. L'utilisateur s'avère moyennement impliqué durant le projet, dans la mesure où c'est la connaissance interne à l'entreprise qui est privilégiée.

Cette méthode reposant sur OMT pour ce qui est de la spécification d'application, les critères technologiques reprennent donc ceux de OMT. Le principal avantage de MKSM sur les autres méthodes confrontées est la modélisation des processus par le modèle SCFC (Source-Cible-Flux-Champ). Cette modélisation des acteurs évoluant dans une situation donnée pourrait donner lieu à une programmation de type agent.

II.3.8 Synthèse et discussion

La figure II.2 représente une synthèse des confrontations selon les cinq dimensions. Pour chaque méthode, l'importance que la méthode accorde à chaque dimension est symbolisée par une colonne. La méthode idéale devrait former une colonne pleine, c'est-à-dire couvrir toutes les dimensions. Mais, si les méthodes testées sont tout a fait pertinentes pour quelques dimensions, elles ne répondent jamais à l'ensemble de celles-ci. OMT, par exemple, possède des formalismes bien décrits permettant de représenter clairement un problème, mais ne peut pas s'appliquer à différents types d'organisation, et ne prend pas en compte la coopération dans un système multi-utilisateur. De même, si OSSAD s'intéresse à l'aspect coopération et organisation, elle ne prend pas en compte l'aspect technologique.



Légende : 1 = préoccupation peu importante, 2 = préoccupation importante, 3 = préoccupation très importante.

Figure II.2. Confrontation des 7 méthodes à l'aide de diagrammes

Ainsi, aucune méthode ne répond à la problématique de spécification et de conception de systèmes d'information distribués dans les organisations de type holonique, mais la figure II.2 montre qu'il peut être possible d'obtenir une méthode plus pertinente en couplant certaines méthodes.

Pour définir cette nouvelle méthode, il est utile à ce sujet de reprendre certains points relatifs aux différentes dimensions.

Pour ce qui est de la dimension représentation, le but étant d'obtenir une méthode d'analyse et de modélisation d'organisation humaine dans un but de spécification et de conception de systèmes d'information distribués, il serait préférable d'utiliser une partie de la méthode UML pour ce qui est de la représentation des données. Pour la représentation des activités, CISAD ne propose pas un formalisme assez clair. L'utilisation de la méthode SADT, et surtout de ses actigrammes est à envisager. Les actigrammes de SADT ne permettant pas de représenter les interruptions de travail, ni les tâches parallèles ; les réseaux de Pétri peuvent apporter leur contribution, comme ils le font dans OSSAD. Pour ce qui est des traitements, le MOT de MERISE est pertinent, mais OSSAD ajoute à ce formalisme la notion de rôle et d'équipe de travail. L'approche doit être orientée par les problèmes, afin de réduire le nombre de cas à envisager. L'activité doit être étudiée en premier lieu, puis de cette analyse d'activité est déduite la représentation des données et des traitements. Ceci de façon à avoir un modèle le plus proche possible de la réalité car déduit de trois points de vue : l'activité, les données et les traitements. Une représentation, nécessaire, de la dynamique doit ensuite être effectuée afin d'obtenir tous les détails du système.

Concernant la dimension organisation, la méthode visée est destinée aux organisations humaines de type holonique dont les flux de données et les stratégies sont connues, ce qui correspond aux systèmes ouverts finalisés. La méthode doit être capable de représenter les systèmes structurés ou semi-structurés. Comme sa finalité est la spécification d'une solution logicielle interactive diminuant la charge de travail et le risque d'erreur dans le groupe, l'environnement ne peut pas être instable ni surtout incertain en ce qui concerne les activités et rôles des acteurs du système. Il est bien sûr nécessaire d'avoir toutes les informations concernant les données qui circulent dans le système, c'est-à-dire les informations sur la quantité, la qualité et la pertinence des données.

Concernant la dimension méthodologie, le cycle nabla (cf. [Kolski 97] [Kolski 98b]) dédié à la conception de systèmes interactifs, intègre de façon importante le futur utilisateur dans le cycle de vie du projet, procède à une approche descendante et suggère l'utilisation d'un modèle réel et un modèle de référence (le modèle visé). L'implication de l'utilisateur est bien sûr essentielle dans les cas qui nous intéressent.

Pour la prise en compte de la dimension coopération, la méthode doit intégrer les aspects essentiels de communication et de relations inter-personnelles telles que la hiérarchie, la responsabilité et prendre en compte certains problèmes de confiance dans l'organisation. Seules les méthodes OSSAD et CISAD se démarquent de par leurs objectifs d'analyse et de modélisation d'organisations humaines.

Concernant la dimension technologie, le mode de traitement sera bien sûr interactif, distribué car les utilisateurs sont répartis géographiquement dans des bureaux, synchrone ou asynchrone en ce qui concerne la transmission de données. L'interface sera adaptable, et devra être adaptative, sinon assistante. L'adaptativité est nécessaire dans un système où les personnes ne jouent pas le même rôle. La programmation doit être basée sur une architecture agent ou au moins être orientée objet ce que proposent les méthodes OMT et UML.

Conclusion

Choisir une méthode d'analyse et de modélisation d'organisation humaine de type holonique dans un but de spécification et de conception de systèmes d'informations distribués implique de considérer les méthodes sur cinq dimensions : la dimension *méthodologie* ; la dimension *environnement* (indiquant à quelles organisations peuvent être appliquées les méthodes) ; la dimension *représentation* (notant l'adéquation des formalismes proposés avec le problème) ; la dimension *technologie* et la dimension *coopération* (indiquant la possibilité pour une méthode de représenter différents aspects de la coopération entre les acteurs de l'organisation étudiée).

Une comparaison de méthodes caractéristiques de leurs domaines d'application a montré que, bien qu'elles répondent (parfaitement ou non) aux besoins pour lesquelles elles ont été définies, ces méthodes ne permettent pas de traiter entièrement notre problème. En revanche, certaines méthodes s'avèrent pertinentes vis-à-vis de certains points (telles que les méthodes MERISE, OSSAD et SADT pour la représentation, la méthode OMT pour la technologie, et les méthodes OSSAD et CISAD pour les aspects organisation et coopération).

L'intégration de certaines de ces méthodes en une nouvelle méthode paraît donc envisageable et devrait permettre de se rapprocher de « la méthode idéale », pour la problématique qui nous préoccupe.

L'objectif du chapitre suivant est donc de proposer une nouvelle méthode adaptée, construite par l'intégration des méthodes étudiées.

Bibliographie du chapitre II

- [Abed 94] Abed M., Angue J.C., A new method for conception, realisation and evaluation of man-machine interfaces. *Proceedings IEEE Systems, Man and Cybernetics conference*, (2-5 Octobre 1994 ; San Antonio, USA).
- [Adam 00a] Adam E., Kolski C., Etude comparative de méthodes du Génie Logiciel en vue du développement de processus administratifs complexes. *Revue Génie Logiciel*, 49, 2000.
- [Balbo 94] Balbo S., Evaluation ergonomique des interfaces utilisateur : un pas vers l'automatisation. Thèse de doctorat, Université de Grenoble I, Septembre 1994
- [Barthet 88] Barthet M.F., *Logiciels Interactifs et ergonomie. Modèle et méthodes de conception*. Paris : Dunod Informatique, 1988.
- [Barthet 95] Barthet M.F., The DIANE Method and its connection with MERISE Method. *Proceedings IEA World Conference "Ergonomic design, interfaces, products, Information"*, Rio de Janeiro, Brazil, pp. 106-110, 16-20 October 1995.
- [Benaissa 93] Benaissa M., Une démarche de conception, réalisation et évaluation d'IHM : application au projet ferroviaire ASTREE. Thèse de doctorat. Université de Valenciennes et du Hainaut-Cambrésis, Décembre 1993.
- [Boehm 88] Boehm B.W., A spiral model of Software Development and enhancement. *Computer*, May 1988.
- [Booch 94] Booch G., *Conception orientée objets et applications. 2ème édition*. USA : Addison-Wesley, 1994.
- [Boy 91] Boy G., *Intelligent Assistant Systems*. New York: Academic Press, 1991.
- [Chang 97] Chang E. J., Dillon T. S., Les méthodes du génie logiciel pour la conception des interfaces utilisateurs, un panorama critique d'approche. *Génie Logiciel*, n°43, 1997, pp. 2-23.
- [Coad 91] Coad P., Yourdon E., *Object-oriented design*. Prentice Hall, 1991.
- [DeMarco 79] DeMarco T., *Structured analysis and system specification*. USA : Yourdon Press, 1979.
- [Dumas 90] Dumas P., Charbonnel G., *La méthode OSSAD, pour maîtriser les technologies de l'information. Tome 1 : principes*. Paris : Les éditions d'organisation, 1990.
- [Ermine 95] Ermine J-L., MKSM, méthode de gestion des connaissances. Rapport de recherche CEA DIST/SMTI, France, 1995.
- [Grislin 96] Grislin M. E., Kolski C., Evaluation des interfaces homme-machine lors du développement de système interactif. *Technique et Science Informatiques (TSI)*, 15, 3, 1996, pp. 265-296.

-
- [Grudin 94]** Grudin J., Eight Challenges for Developers. *Communications of the ACM*, 37, 1, 1994, pp. 93-105.
- [Helander 97]** *Handbook of Human-Computer Interaction*. Ed. by Helander M., Landauer T.K., Prabhu P., Netherlands : Elsevier Science Publishers, 1997.
- [Henderson-Sellers 99]** Henderson-Sellers B., Firesmith D.G., Comparing OPEN and UML: the two third generation OO development approaches. *Information and software technology*, 41, 1999, pp. 139-156.
- [IGL 89]** I.G.L. Technology, *SADT, un langage pour communiquer*. Paris : Eyrolles, 1989.
- [Iivari 98]** Iivari J., Maansaari J., The usage of systems development methods: are we stuck to old practices ? *Information and software technology*, 40, 1998, pp. 501-510.
- [Jackson 95]** Jackson M., *Software Requirements & Specifications : A Lexicon of Practice, Principles and Prejudices*. USA : Addison-Wesley, ACM Press Books, 1995.
- [Jacobson 93]** Jacobson I., Christenson M., Jonsson P., Overgaard G., *Le génie logiciel orienté objet, une approche fondée sur les cas d'utilisation*. USA : Addison-Wesley, ACM Press Books, 1993.
- [Kolski 97]** Kolski C., *Interfaces homme-Machine, application aux systèmes industriels complexes (2^{ème} édition revue et étendue)*. Paris : Editions Hermes, 1997.
- [Kolski 98]** Kolski C., A "call for answers" around the proposition of an HCI-enriched model. *ACM SIGSOFT Software Engineering Notes*, 23, 3, 1998, pp. 93-96.
- [Kolski 98b]** Kolski C., Le Strugeon E., A review of "intelligent" human-machine interfaces in the light of the ARCH Model. *International Journal of Human-Computer Interaction*, 10, 3, 1998, pp. 193-231.
- [Le Moigne 77]** Le Moigne J.L., *La théorie du système général - théorie de la modélisation*. Vendôme : Presses Universitaires de France, 1977.
- [Lim 94]** Lim K.Y., Long J.B., The MUSE method for usability engineering. *Cambridge series on Human-Computer Interaction*, Cambridge, USA : Cambridge University Press, 1994.
- [Medina-Mora 92]** Medina-Mora R., Winograd T., Flores R., Fores F., The action Workflow Approach to Workflow Management Technology. *Proceedings of the Computer Supported Cooperative Work conference (CSCW'92)*, (31 October - 4 November 1992 ; Toronto, Canada), USA : ACM Press, pp. 281-288, 1992.
- [Morejon 94]** Morejon J., *MERISE, vers une modélisation orientée objet*. Paris : Les Editions d'Organisation, 1994.
- [Muller 97]** Muller P.A., *Modélisation objet avec UML*. Paris : Eyrolles, 1997.
- [Nielsen 93]** Nielsen J., *Usability engineering*. Academic Press, 1993.

-
- [**Nurcan 96**] Nurcan S., Analyse et conception de systèmes d'information coopératifs. *Technique et Science Informatiques*, 15, 9, 1996, pp. 1287-1315.
- [**Paide 99**] Paide R. F., When are methods complementary. *Information and software technology*, 41, 1999, pp. 157-162.
- [**Pascot 93**] Pascot D., Bernadas C., L'Essence des Méthodes : Etude Comparative de Six Méthodes de Conception de Systèmes d'Information Informatisés. *Actes du congrès INFORSID'93 «Systèmes d'information, systèmes à base de connaissances»*, Lille, 11-14 Mai 1993.
- [**Perrin 97**] Perrin P., La guerre des standards des méthodes orientées objet n'aura pas lieu. *Ingénierie des Systèmes d'Information*, 5, 5, 1997, pp. 533-552.
- [**Ross 77**] Ross D.T., Structured analysis (SA) : a language for communicating ideas. *IEEE Transactions on Software Engineering*, SE-3, 1, Janvier 1977.
- [**Rumbaugh 91**] Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorenzen W., *Object-oriented modeling and design*. Prentice Hall, 1991.
- [**Rumbaugh 98**] Rumbaugh J., Jacobson I., Booch G., *Unified Modeling Reference Manual*. Addison Wesley, 1998.
- [**Schneider-Hufschmidt 93**] *Adaptive User Interfaces*. Ed. by Schneider-Hufschmidt M., Kühme T., Malinkowski U., Netherlands : Elsevier Science Publishers, 1993.
- [**Tarby 93**] Tarby J.C., Gestion automatique du dialogue homme-machine à partir des spécifications conceptuelles. Thèse de doctorat, Université de Toulouse, septembre 1993.
- [**Tardieu 85**] Tardieu H., Rochfeld O., Colleti R., Panet G., Vahee G., *La méthode Merise, démarche et pratiques (tome 2)*. Editions d'Organisation : Paris, 1985.
- [**Tardieu 91**] Tardieu H., Rochfeld O., Colleti R., *La méthode Merise, principes et outils (tome 1), 2ème édition*. Paris : Editions d'Organisation, 1991.
- [**Thimbleby 94**] Thimbleby H., Marsh S., Jones S., Cockburn A., Trust in CSCW. *CSCW : The multimedia and networking paradigm*, UNICOM, ed. by Scrivener S.A.R., Aldershot, England: Avebury Technical, pp. 253-272, 1994.
- [**Williges 87**] Williges R.C., Williges B.H., Elkerton J., Software Interface Design. Ed. by Salvendy G., *Handbook of Human factors*, pp. 1416-1449, New York, Wiley, 1987.

Chapitre III -

MAMOSACO : une Méthode Adaptable de Modélisation de Systèmes Administratifs Complexes

INTRODUCTION	92
III.1 BESOINS EN ANALYSE ET MODELISATION POUR UNE NOUVELLE METHODE	94
III.2 ELEMENTS DE MODELISATION DANS LA METHODE MAMOSACO	99
III.3 MODELE DYNAMIQUE ET SIMULATION	102
III.4 LES DIFFERENTES PHASES DE LA METHODE MAMOSACO	115
III.5 SPECIFICATION DE SYSTEME ORIENTE HOLON POUR L'AIDE AU TRAVAIL COOPERATIF (SOHTCO)	118
III.6 MODELE NABLA ET PRINCIPE DE REUTILISABILITE DANS LA METHODE MAMOSACO	130
CONCLUSION	133
BIBLIOGRAPHIE DU CHAPITRE III	134

Introduction

Un des objectifs de ce mémoire est de proposer une méthode d'analyse et de modélisation d'organisations qui permette de mettre en évidence les coopérations entre les acteurs d'une organisation de type holonique afin de spécifier et concevoir un système d'information distribué basé sur une approche multi-agent.

Comme l'a montré le chapitre II, aucune méthode ne répond totalement à ce besoin. Il convient donc de créer une nouvelle méthode, intégrant les parties les plus pertinentes de méthodes existantes. La méthode créée doit satisfaire les critères méthodologiques, organisationnels, représentationnels, coopératifs et technologiques définis dans le chapitre précédent.

Cette méthode devra couvrir l'ensemble du cycle de vie du projet. Dans cette thèse, nous nous intéressons particulièrement aux phases d'analyse, de modélisation d'organisations humaines et de spécification d'un système d'information distribuée¹. La dynamique est particulièrement importante dans les organisations humaines, et un modèle dynamique ne peut suffire à représenter et à valider les interruptions, les boucles et le parallélisme d'action. C'est pourquoi une phase de simulation doit être effectuée suite à la modélisation afin d'obtenir toutes les données nécessaires à la spécification, puis à la conception.

Nous visons à ce que deux types d'organisations soient considérés par la méthode : l'organisation humaine et l'organisation agent constituant le système d'information distribué, avec pour objectif que ces deux organisations soient de type holonique. Il est donc envisageable d'utiliser les mêmes formalismes pour, d'une part, modéliser l'organisation humaine, et d'autre part spécifier l'organisation agent. Ceci constitue un des points centraux de la méthode MAMOSACO (Méthode Adaptable de Modélisation de Systèmes Administratifs complexes) qui fait l'objet de ce chapitre constitué de six parties.

Dans la première partie de ce chapitre, les besoins en analyse et en modélisation que doit couvrir la méthode sont présentés.

Dans la seconde, les formalismes permettant la représentation des données, des flux de données au sein d'une organisation holonique, des traitements de données par les acteurs d'une

¹ Les phases de conception détaillée, de réalisation, de validation technique ainsi que d'évaluations ergonomiques a priori et a posteriori ne sont pas, dans un premier temps, intégrées à la méthode.

organisation et de la dynamique d'une organisation sont décrits du point de vue de leur application à une organisation humaine et à une organisation agent.

La représentation de la dynamique des processus de type workflow, ainsi que des acteurs et des problèmes d'interruptions qu'ils connaissent, nécessitent d'aller plus loin que l'utilisation de réseaux de Petri classiques (RdP) ou de machines à états finis. Nous expliquerons pourquoi nous avons choisi à ce sujet les réseaux de Petri paramétrés. Ce modèle dynamique, et la simulation à laquelle il mène, font l'objet de la troisième partie de ce chapitre.

La quatrième partie de ce chapitre résume l'ensemble de la méthode et montre que celle-ci peut mener à la spécification d'une organisation multi-agent holonique.

La spécification de cette organisation holonique multi-agent est présentée dans la cinquième partie de ce chapitre.

Finalement, nous proposerons dans la sixième partie de ce chapitre quelques principes de réutilisabilité de la méthode MAMOSACO concernant essentiellement les phases de spécification et de conception des SOHTCO

III.1 Besoins en analyse et modélisation pour une nouvelle méthode

Le but de la méthode visée est l'analyse et la modélisation d'organisations humaines de type holonique, fonctionnant par envoi de messages, dans un but de spécification d'organisations multi-agents, holoniques également, facilitant le travail coopératif et la gestion de l'information.

Cette méthode possède donc des besoins particuliers pouvant être regroupés autour des dimensions organisation, méthodologie, représentation, coopération et technologie présentées au chapitre II. Elle possède également des besoins vis-à-vis de l'analyse qui doit permettre de définir les rôles joués par les acteurs des organisations humaines ainsi que leurs relations (de responsabilité et de hiérarchie) en un minimum d'étapes. Ces besoins sont successivement, détaillés.

III.1.1 Besoins selon les cinq dimensions

La plupart des organisations humaines de travail reposent sur la création, la manipulation et l'échange de données, entre leurs acteurs. Ces flux de données symbolisent l'ossature de l'organisation. Dans le cas d'une organisation *holonique*, comme dans le cas d'une organisation administrative complexe, qui pourrait être qualifiée d'optimale (au sens de Koestler), cette structure doit refléter les niveaux de responsabilité différents que possèdent les acteurs. Les échanges de données sont évidemment très importants pour la conception de systèmes d'information distribués ; la méthode doit donc considérer les données de façon *quantitative* et *qualitative* (c'est-à-dire qu'elle doit considérer leurs structures). Pour permettre au système logiciel final d'apporter une aide appropriée aux acteurs, la méthode doit également considérer la *pertinence* des données, c'est-à-dire leur adéquation avec la procédure en cours d'exécution. Actuellement, dans le contexte de globalisation de l'économie, l'environnement des organisations possède une stabilité relative. Cependant, les organisations de type administratif complexe, auxquelles est principalement destinée la méthode, sont généralement considérées comme moins sensibles aux fluctuations de leur environnement que d'autres types d'organisations telles que les organisations de type unité de production ou unité de service. C'est pourquoi la méthode peut considérer l'environnement de l'organisation comme étant relativement structuré, mais stable et certain.

En ce qui concerne la méthodologie, nous proposons que le cycle de développement choisi suive le modèle *nabla* (noté ∇) [Kolski 1997] [Kolski 1998]. La démarche

méthodologique qui lui est inhérente intègre les acteurs et futurs utilisateurs dans les phases d'analyse, de modélisation et de spécification. Elle est particulièrement bien adaptée au problème car elle propose la séparation entre la conception des interfaces homme-machine et la conception des modules applicatifs (éventuellement intelligents) des systèmes interactifs (fig. III.1). Elle énonce également un principe de réutilisabilité pour le développement des interfaces et/ou pour le développement des modules applicatifs ou d'aide en fonction du domaine d'application, proposant ainsi d'établir une bibliothèque, une base de composants réutilisables aux différents niveaux du cycle. Une adaptation de ce modèle est proposée suite à la spécification du système multi-agent holonique dans le chapitre IV.

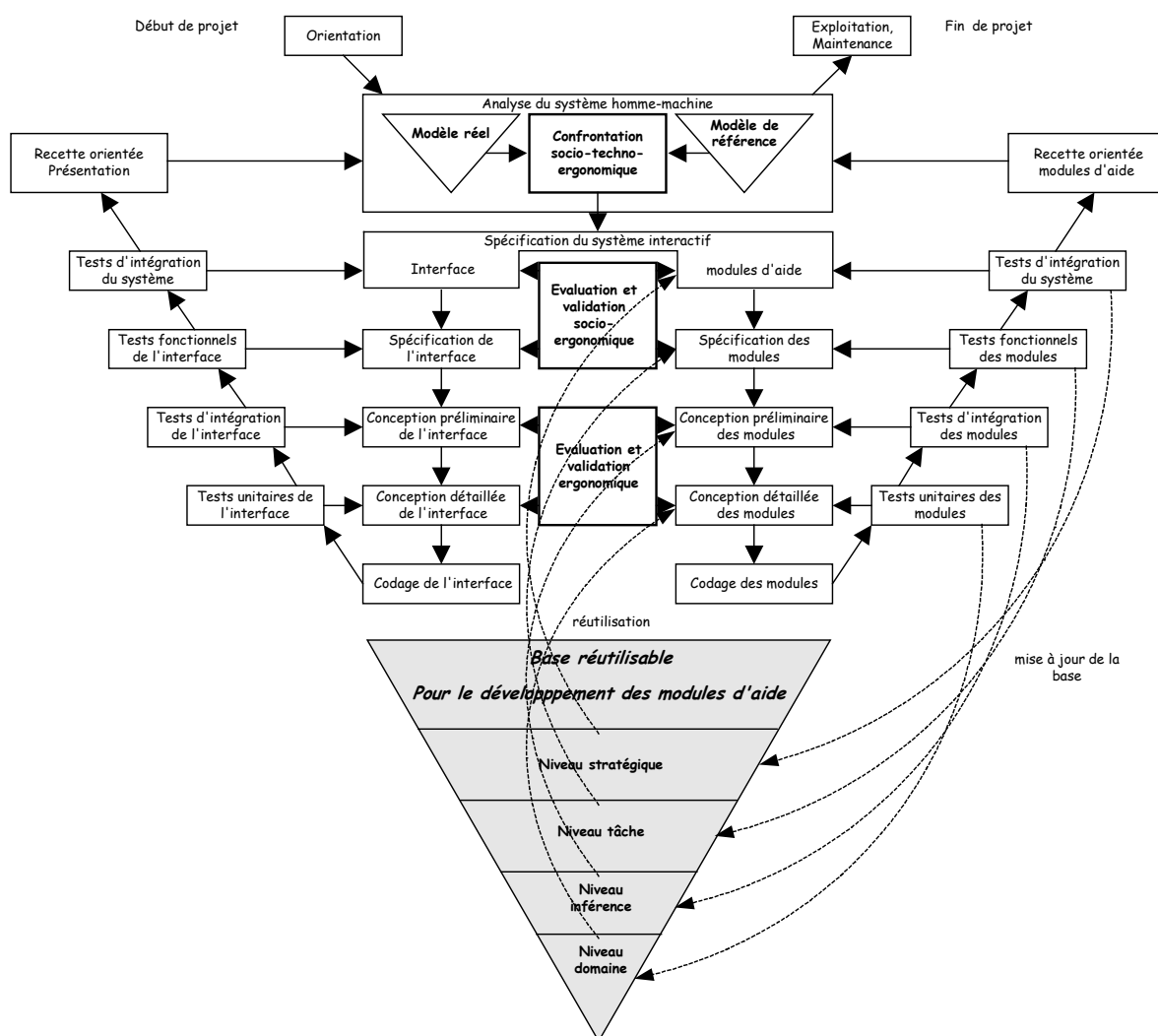


Figure III.1. Modèle ∇ pour le développement de modules d'aide adapté de [Kolski 1997]

La figure III.1 représente le modèle nabla, adapté à la conception d'un système d'aide composé de modules. Les différents niveaux de la base réutilisable s'inspirent de ceux présents dans la méthode KADS dans ses premières versions [Wielinga 93], mais il est

également possible de découper, de structurer cette base en fonction des différents niveaux de l'organisation multi-agent holonique que la méthode vise à spécifier.

La méthode doit suivre ce modèle et couvrir les phases classiques *d'analyse*, de *modélisation*, de *spécification* et de *conception*. Cependant, les organisations humaines étant fortement dynamiques, il est important d'ajouter une phase de simulation. Celle-ci doit permettre de recueillir des données quant à la dynamique du système et de tester, éventuellement, de nouvelles solutions organisationnelles sur un simulateur. L'interface homme-machine, la partie visible du système, n'est pas le point le plus difficile à mettre en place, des solutions simples de type office assistant de Microsoft © pouvant s'avérer suffisantes. Il s'agit en effet d'assister les acteurs de l'organisation humaine dans leurs activités. Du fait de la démarche participative proposée par le modèle ∇ , l'évaluation a priori est inhérente à celle-ci ; il n'est donc pas nécessaire de faire ressortir une phase à ce sujet. Cependant, une phase *d'évaluation a posteriori* 'sur le terrain', auprès des utilisateurs, est bien sûr recommandée. L'approche de développement *évolutive* est la plus pertinente dans le cas d'organisations administratives complexes. En effet, les activités, réalisées par les acteurs, et les procédures (les ensembles d'activités) étant le plus souvent très (trop) nombreuses, il est nécessaire de modéliser premièrement les procédures principales pour les décomposer en activités, et les assembler ensuite pour obtenir une vue globale du système.

En ce qui concerne les différents formalismes utilisés pour la représentation de l'organisation, ceux-ci doivent être accessibles aux différents acteurs, de manière à les impliquer dans l'exercice de modélisation et dans la mise en place du système visé comme le préconise le modèle ∇ . Afin de répondre aux exigences des organisations administratives complexes évoquées précédemment, il est impératif de disposer, à la fois, d'un modèle de données, d'un modèle d'activités, d'un modèle de traitements et d'un modèle dynamique. Les trois premiers modèles, statiques, font l'objet de la deuxième partie de ce chapitre. Le modèle dynamique est détaillé dans la troisième partie.

Afin de concrétiser l'implication des utilisateurs dans le projet de spécification d'un outil d'aide à la coopération, un simulateur doit être développé à partir du modèle dynamique. Un tel outil permet en effet la comparaison objective et commune, de divers scénarios possibles. La simulation permet également de visualiser et d'estimer la coordination entre les acteurs, ainsi que les communications asynchrones et directes. Cependant, pour répondre aux besoins

définis dans le chapitre II concernant la coopération, le simulateur doit également représenter les communications synchrones et indirectes.

La finalité de la méthode recherchée est la spécification d'une organisation multi-agent assistant les acteurs de l'organisation humaine. La méthode doit donc guider vers une programmation utilisant les technologies issues de l'I.A.D. (telles que les architectures agent).

III.1.2 Besoins vis-à-vis de l'analyse

Une organisation administrative complexe est caractérisée par des échanges de données entre différents acteurs ayant des rôles généralement différents pour la réalisation en commun d'un objectif donné.

L'analyse doit donc permettre, après identification des principaux objectifs de l'organisation (ses procédures principales), de préciser le rôle des différents acteurs et de décrire les tâches assignées à chaque rôle. L'analyse des interfaces externes doit aussi être effectuée afin d'identifier les entrées-sorties de l'organisation, par exemple les liens avec les fournisseurs, les clients ou d'autres organisations. L'analyse doit s'effectuer en trois phases successivement décrites.

La première phase est "orientée procédures". Chaque procédure doit y être décrite par un enchaînement d'activités réalisées par l'ensemble des acteurs. La figure III.2 est un extrait d'une analyse orientée procédure d'une organisation administrative complexe.

....
 - L'inventeur reçoit un avis de brevetabilité. Il constitue un dossier technique qu'il envoie au conseil en brevet.
 - Le conseil en brevet rédige alors un projet de demande de brevet, quelquefois en coopération avec l'inventeur.
 - Il place le projet dans le dossier et le transmet à l'adjoint au chef de département pour vérification.
 - Celui-ci lit le dossier et le retourne en cas de modification. Sinon, il transmet ce dossier au bureau administratif
 - Le bureau administratif copie le projet et le classe dans le dossier principal. Puis il transmet le dossier de circulation à l'inventeur.

Figure III.2. Exemple d'analyse orientée processus

La deuxième phase d'analyse concerne l'identification des rôles joués par les acteurs de l'organisation. Cette phase décrit toutes les activités des rôles en fonction des procédures dans lesquelles ils interviennent. Elle renseigne donc sur la charge de travail que peut rencontrer un acteur impliqué dans plusieurs procédures, ce qui explique qu'un acteur peut être interruptible ou non. La figure III.3 représente un extrait d'une analyse orientée rôle de la même organisation administrative complexe, pour l'acteur "conseil en brevet".

-
- Le conseil en brevet
 - procédure d'étude de brevetabilité
 - sur réception d'une demande de brevetabilité, le conseil crée une farde² bibliographique qu'il transmet à son responsable.
 - ...
 - procédure de dépôt de brevet
 - sur réception d'un avis de brevetabilité d'un inventeur, le conseil rédige un projet de demande de brevet, quelquefois en coopération avec l'inventeur.
 - il place le projet dans le dossier et le transmet à l'adjoint au chef de département pour vérification.
 - sur réception du dossier de circulation provenant du chef de service...
-

Figure III.3. *Exemple d'analyse orientée rôle*

La dernière phase d'analyse doit concerner les documents recensés lors des phases précédentes. Elle permet en outre de valider ces deux phases, car chaque document doit figurer dans celles-ci. Il ne peut, théoriquement, exister de documents cités lors d'une seule phase.

Les documents utilisés dans les organisations administratives complexes possèdent en général une structure, une forme, bien établie. L'analyse des documents doit donc transcrire cette structure, mais elle doit également renseigner sur le fait qu'un document compose ou est composé d'autres documents. L'objectif visé étant la conception d'un logiciel, la taille du document importe également ainsi que sa durée de vie (la date à partir de laquelle le document peut être jugé comme obsolète doit apparaître). La figure III.4 propose un extrait d'une analyse orientée document d'une organisation administrative complexe.

- ...
- Le projet de demande de brevet contient la référence à l'invention, le résumé de l'invention, l'état de l'art du domaine de l'invention, et la description de l'invention.
 - Le dossier de circulation contient un n° de dossier interne, les noms des inventeurs, les mots clé relatifs à l'invention et inclut le projet de demande de brevet.
 - Le dossier principal contient un exemplaire du dossier technique et du projet de demande.
- ...

Figure III.4. *Exemple d'analyse orientée document*

L'activité, ce qui est effectivement réalisé par l'acteur, est très souvent différente de la tâche prescrite, ce qui doit être fait [Montmollin 86]. Cette différence est encore plus sensible dans le cas de systèmes coopératifs où ce n'est plus un acteur qui doit s'adapter à la situation, mais un groupe d'acteurs. L'analyse d'activités, réalisée sur le terrain, est donc primordiale ; elle peut aller jusqu'à l'analyse de l'activité cognitive, pour la définition des règles

² Une farde est le terme utilisé en Belgique pour indiquer un dossier ou classeur.

d'adaptation personnelles ou communes à un groupe de travail. Ces règles sont alors répertoriées dans la phase de l'analyse orientée rôle. Ces phases d'analyse peuvent être effectuées par des techniques conventionnelles telles que les observations, les questionnaires, les interviews et l'analyse des protocoles [Diaper 1989], [Wilson 1990], [Kolski 1997].

Comme il est souvent peu réaliste d'analyser en détail toute une organisation complexe, il faut se concentrer sur l'analyse des procédures critiques, c'est-à-dire celles qui sont jugées stratégiques ou qui peuvent poser des problèmes, et donc celles où le besoin d'une assistance intelligente à la gestion d'information se fait ressentir.

III.2 Éléments de modélisation dans la méthode MAMOSACO

Dans la méthode visée, la modélisation doit refléter l'importance des données dans l'organisation humaine, être capable de souligner les lieux de communication et de coopération. Les modèles doivent être clairs et lisibles par toute personne impliquée dans l'organisation (afin de répondre au critère d'implication des utilisateurs).

Les modèles doivent aussi permettre la spécification de l'organisation agent, ceci afin de limiter le nombre de modèles utilisés par la méthode. En effet, on considère généralement que trop de modèles peuvent nuire à son applicabilité.

III.2.1 Modèle des données

Les données constituent la "matière première" de l'organisation, ce qu'elle est amenée à utiliser, à enrichir et à valoriser. Leur modèle doit permettre de représenter à la fois leur structure et les relations qui existent entre elles (appartenance, classe, étiquettes). Les données peuvent être statiques (un document papier), mais aussi dynamique (comme peut l'être un document électronique par exemple). Afin de répondre à ces critères, il est préférable d'utiliser un modèle de données issu d'une méthode orientée objet telle que OMT [Rumbaugh 91], BOOCH [Booch 94] ou UML [Rumbaugh 98]. Le modèle des données du langage UML répond au critère de lisibilité. Il permet en outre de représenter le passage d'un document papier à un document électronique, par un ajout de méthodes à la représentation de la donnée.

Dans le formalisme de UML, une classe d'objet est représentée par un rectangle constitué de trois parties : le nom de la classe, les attributs (propriétés) que possèdent les objets de cette classe, et les méthodes (opérations) que peuvent exécuter ces objets.

Trois types de liens peuvent associer les classes. Les associations relient plusieurs objets par des relations simples entre attributs (par exemple, une carte grise est associée à un conducteur et à une voiture par le nom et le numéro d'immatriculation). Les agrégations sont utilisées pour décrire le fait qu'un objet soit constitué d'autres objets. Les liens d'héritage, qui font la force du concept objet, facilite la description d'objets ayant le même comportement (par exemple, la classe véhicule regroupe les caractéristiques communes aux voitures, et motos. Les classes décrivant ces types de véhicules héritent de cette classe et l'enrichissent).

Ainsi, il est possible, avec un tel formalisme, de représenter lisiblement les données manipulées par les acteurs d'organisations administratives complexes. Par exemple, la figure III.5 représente une farde (un dossier), composée d'un formulaire, d'un rapport et d'un sous dossier qui lui-même contient d'autres documents.

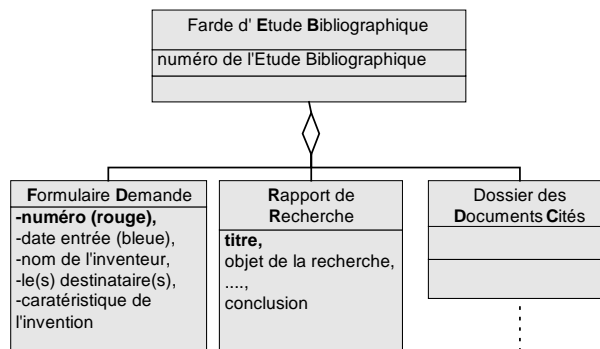


Figure III.5. Exemple de représentation de documents par le modèle de données de UML

III.2.2 Modèle des flux de données

Le modèle des flux de données peut décrire l'architecture d'une organisation s'il permet de représenter en particulier les relations de hiérarchie et/ou de responsabilité entre ses acteurs constituant les nœuds de communications.

Le modèle des actigrammes de la méthode SADT [IGL 89], très intuitif et fort répandu, permet de représenter clairement, pour chaque activité, les données en entrée et les données en sortie. Pour identifier les acteurs et représenter, en plus des liens de communication, leurs relations de hiérarchies et/ou de responsabilités, les actigrammes doivent être structurés en niveaux. Ceci est possible en associant à chaque niveau une ligne et en organisant ces lignes par niveaux de responsabilités décroissants de bas en haut, à l'image du modèle des traitements de OSSAD [Dumas 90].

La figure III.6 est un extrait de modélisation d'une procédure que réalisent cinq acteurs classés par ordre de responsabilité décroissant : le responsable secteur, le chef de service, le conseil en brevet, le chargé d'étude et la secrétaire. Dans cette procédure, ces acteurs s'échangent des documents dont une farde (un dossier), afin d'élaborer une stratégie. "EB" fait référence à une étude bibliographique et "dmd n°" à une demande de numéro.

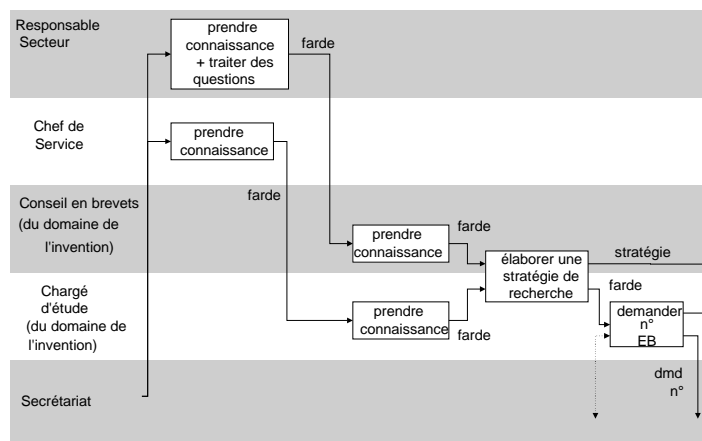


Figure III.6. Extrait d'un modèle de flux de données

Cette représentation des flux de données peut également être utilisée lors de la spécification de l'organisation multi-agent. Les agents peuvent en effet être classés selon leurs niveaux de responsabilité décroissant (de l'agent responsable de l'organisation aux agents responsables d'actions plus précises comme la réception d'informations par exemple).

III.2.3 Modèle des traitements de données

Suite à la modélisation du flux de données, qui donne une vue d'ensemble de l'organisation et de ses procédures, il est nécessaire de disposer un modèle représentant le traitement des données afin de détailler les procédures schématisées par le précédent modèle.

En effet, le modèle d'activité représente tous les flux possibles entre les acteurs, mais ne représente pas les conditions dans lesquelles ces flux suivent un chemin précis. Cette notion de condition se trouve dans le modèle des traitements.

Ce modèle doit également faire ressortir la coopération entre les acteurs, et bien sûr, les liens de hiérarchie et/ou de responsabilité. Parmi les méthodes comparées au chapitre II, seule la méthode OSSAD propose un modèle de traitement clair, inspiré du modèle de traitements de MERISE [Tardieu 91], et qui permet de représenter la coopération et les notions de hiérarchie.

La figure III.7 est un extrait d'un modèle de traitement. Les acteurs sont représentés par niveaux de responsabilité croissant de gauche à droite. Les conditions sont symbolisées par des losanges. Dans cet exemple, le chargé d'étude retourne le document 'EB' (Etude Bibliographique) au chef de service s'il a effectué d'importantes modifications sur ce document, sinon, il poursuit la procédure.

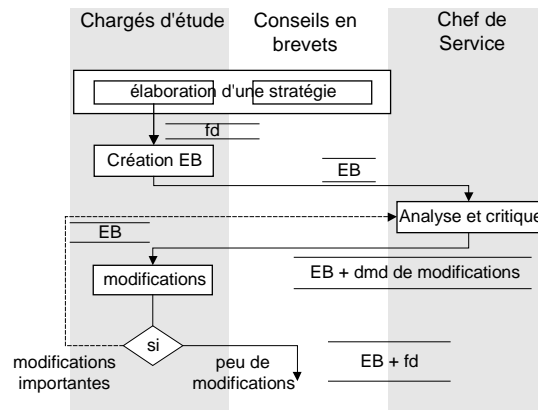


Figure III.7. Extrait d'un modèle des traitements

A l'instar du modèle d'activité, ce modèle peut être utilisé pour la spécification de l'organisation multi-agent holonique. Aux acteurs correspondent alors les agents, classés par niveau de responsabilité croissant et traitant des données pour les transmettre à d'autres agents sous certaines conditions.

Les deux derniers modèles permettent de vérifier facilement que l'organisation multi-agent spécifiée suit les règles holoniques de communication. En effet, les agents doivent communiquer en suivant la holarchie. Il ne peut donc exister de flux traversant le modèle et 'sautant' des niveaux de responsabilité.

Cependant, si ces modèles peuvent être réutilisés lors de la phase de spécification, ils ne sont pas suffisants pour y mener. En effet, il manque des informations importantes quant à la dynamique de l'organisation humaine étudiée [Vergison 98]. Ce point est étudié ci-dessous.

III.3 Modèle dynamique et simulation

Les trois précédents modèles (modèle de données, modèle de flux de données et modèle de traitements) ne peuvent suffire pour la modélisation d'une organisation humaine. Il manque en effet la prise en compte de la dynamique de ces organisations comme le travail parallèle, les interruptions et les boucles [Jambon 96].

Cette représentation de la dynamique peut être réalisée par des machines à état finis, ou plus habituellement par les réseaux de Petri (citons entre autres [Abed 94], [Bastide 00], [Palanque 95], [Jambon 96]). Un courant de recherche sur les réseaux de Petri représentant des processus de workflow existe d'ailleurs à ce sujet. Ces Workflow Nets [Van der Alst 98] permettent de représenter l'enchaînement des tâches caractérisant ces processus. Néanmoins, ils ne prennent pas véritablement en considération les acteurs, c'est pourquoi les réseaux de Petri paramétrés ont été utilisés.

III.3.1 Choix des réseaux de Petri paramétrés

Le réseau de Petri (Petri net) doit permettre de suivre les flux de données de l'organisation administrative complexe, ainsi que les déplacements des acteurs entre les espaces de travail. Ce réseau doit également représenter les états des documents ainsi que les activités des acteurs, sans oublier les interruptions, qui sont inhérentes aux organisations humaines.

L'utilisation des réseaux de Petri classiques ne donne pas une vue suffisamment claire de ces besoins. En effet, dans ces réseaux doivent figurer les acteurs, les documents manipulés et les activités réalisées par les acteurs. Plusieurs utilisations des réseaux sont possibles, par exemple, en associant aux places les activités ou les acteurs. Dans la figure III.8, l'acteur est symbolisé par un jeton qui change de place au même titre que l'acteur change d'activité. Les transitions représentent les événements tels que la fin d'une activité ou l'arrivée d'une interruption. Si une personne réalisant l'activité 1 est interrompue, le jeton change de place et reste dans la place de gestion d'interruption tant que l'acteur n'a pas terminé de gérer toutes les interruptions.

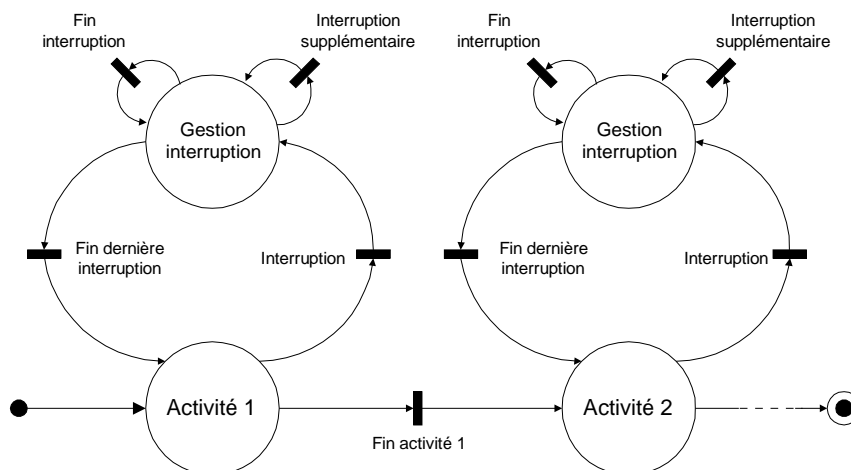


Figure III.8. Réseau de Petri où une place représente une activité.

Les réseaux de Petri de haut niveau tels que les réseaux de Petri à objets ou les réseaux de Petri colorés [Borusan 93] permettent d'ajouter du sens aux jetons. La figure III.9 est un exemple de réseaux de Petri colorés (pour des raisons de lisibilité, les couleurs ont été remplacées par des symboles).

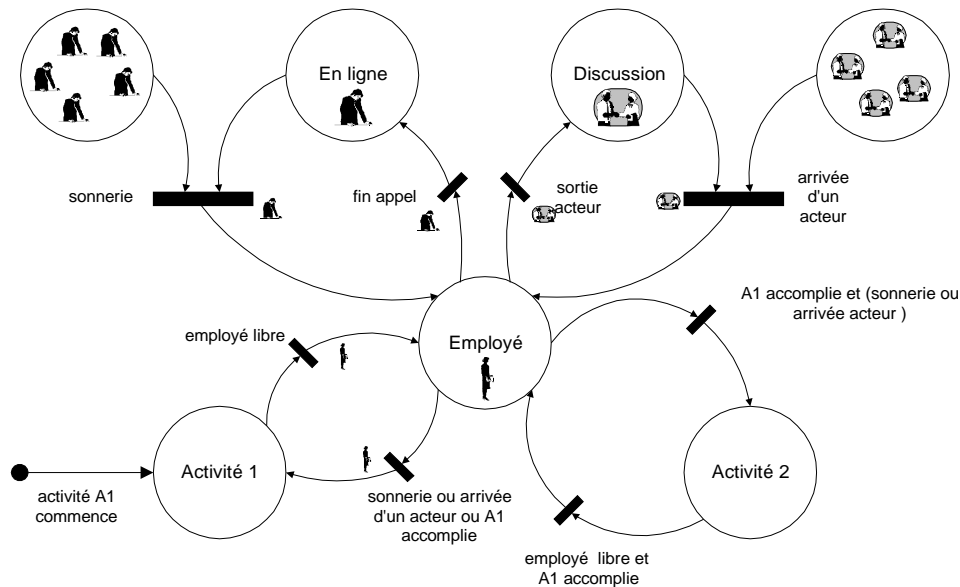


Figure III.9. Réseau coloré où un acteur est symbolisé par une place

La figure III.9 représente un réseau de Petri coloré. Un acteur (ici l'employé) est symbolisé par une place. Dans l'exemple, il gère l'activité A1. Les autres places représentent les activités et les interruptions possibles (ici, l'arrivée d'un acteur et ou la réception d'un appel téléphonique). La couleur (ici la forme) du jeton présent dans la place indique son état actuel.

Cette représentation peut sembler satisfaisante pour la représentation d'un acteur, mais n'est pas pertinente en ce qui concerne la modélisation de l'ensemble d'un processus de type workflow qui nécessite la représentation de l'enchaînement des activités des acteurs. Le réseau devient alors rapidement surchargé et ne répond plus à notre critère de lisibilité.

Des études sur la représentation de processus workflow à l'aide de réseau de Petri ont mené à la définition de Workflow net [Van der Alst 98] [Vorhoeve 96]. Dans un workflow net, il existe une place source (qui ne comporte pas de transition en amont) et une place puits (qui ne comporte pas de transition en aval). Elles sont les seuls éléments du réseau à avoir cette caractéristique, il existe toujours au moins un arc reliant un élément du réseau à un autre élément.

Dans un workflow net (voir figure III.10), les transitions représentent les tâches (telles que "envoyer le brevet", "compléter la farde"), et les places représentent des conditions pour que puissent s'effectuer ces tâches.

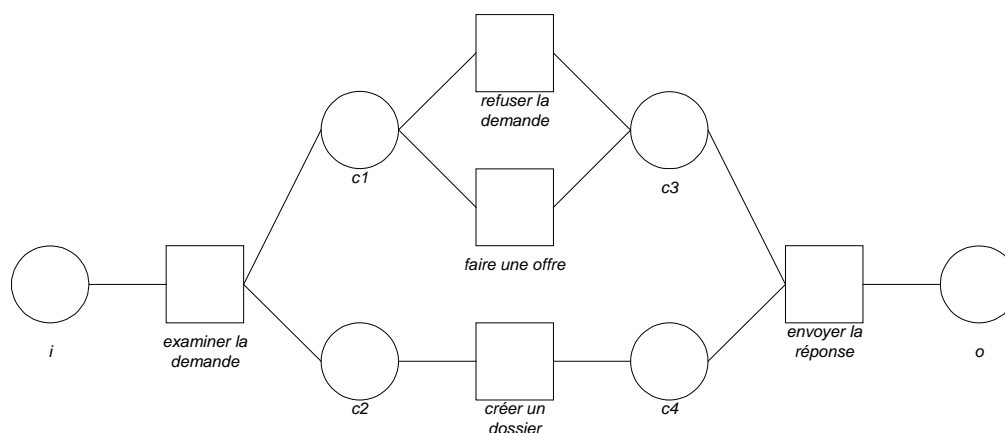


Figure III.10. Exemple de workflow net

La figure III.10 représente une procédure (simplifiée) de réponse à une demande. Les places représentent les conditions. Par exemple, la place initiale représente la condition "demande arrivée", la place c1 correspond à la condition "demande à traiter" et mène à une des deux tâches "refuser la demande" ou "faire une offre". Cette fois, dans le cas des workflow nets, la représentation de l'ensemble d'un processus est possible, mais il n'y a aucune trace des acteurs et de leur rôle. Cette représentation ne satisfait donc pas nos critères.

Les réseaux de Petri paramétrés font partie des réseaux de Petri de haut niveau et permettent de représenter clairement des processus complexes [Agimont 96] [Gracanian 94] grâce à leurs possibilités d'abstraction. En effet, dans les réseaux de Petri paramétrés, il est possible d'unir dans un paramètre un ensemble cohérent d'objets ou de valeurs. Ceci permet alors de créer une abstraction et de manipuler des ensembles d'objets plutôt que des objets individuellement. Nous nous orientons donc vers une application des réseaux de Petri paramétrés pour la modélisation des activités et des interruptions des acteurs de processus de type workflow.

III.3.1.1 Définition des réseaux de Petri paramétrés

Les réseaux de Petri paramétrés sont définis dans le tableau III.1.

Tableau III.1. Définition d'un réseau de Petri paramétré (extrait de [Agimont 96])

<p>Un réseau de Petri paramétré est défini par un sex-uplet $RPP = \{C, D, P_p, T, I, O\}$ où :</p> <p>1) $C = \{CV_1, \dots, CV_{ C }\}$ est l'ensemble des valeurs des paramètres CV_i. Un paramètre est une classe d'objets ou de valeurs prises par des objets.</p> <p>2) $D = CV_1 \times CV_2 \times \dots \times CV_{ C }$ est l'ensemble de tous les vecteurs. Un vecteur est donc une combinaison de paramètres. Dans un réseau de Petri paramétré, ce sont les vecteurs de D qui sont consommés ou produits. Ils seront donc symbolisés par des jetons.</p> <p>3) PP_i est une place paramétrée, contenant des vecteurs. C'est donc, par définition, un sous-ensemble de D.</p> <p>4) $Pp = \{PP_1, \dots, PP_{ Pp }\}$, l'ensemble des places du réseau est appelé un descripteur de paramétrisation. Il doit vérifier que l'ensemble des places forme l'ensemble D, et qu'aucune classe ne contient le même jeton (le même vecteur) qu'une autre place. Il doit donc vérifier :</p> $\bigcup_{i=1}^{Pp} PP_i = D \text{ et } PP_i \cap PP_j = \emptyset$ <p>5) $T = \{pt_1, \dots, pt_{ T }\}$ est l'ensemble de toutes les actions, associées aux transitions, qui peuvent être exécutées par le système.</p> <p>6) pt_i est une transition de vecteur définie par $pt_i : I(pt_i) \rightarrow O(pt_i)$, où $I(pt_i)$ est l'ensemble des vecteurs consommés par la transition de vecteur pt_i, et $O(pt_i)$, l'ensemble des vecteurs produits.</p> <p>7) Une transition paramétrée est un sous-ensemble contenant donc plusieurs transitions de vecteur. Dans une transition paramétrée, le nombre de vecteurs consommés doit être le même pour toutes les transitions de vecteurs qu'elle contient. Il en est de même pour le nombre de vecteurs produits. La notation formelle est la suivante :</p> $t_i = \{pt_{i1}, \dots, pt_{i t_i }\} \subseteq T, \text{ est une transition paramétrée } \Leftrightarrow \forall i, j, k : pt_i, pt_j \in t, PP_k \in P_p,$ $ I(pt_i) \cap PP_k = I(pt_j) \cap PP_k \text{ et } O(pt_i) \cap PP_k = O(pt_j) \cap PP_k $ <p>8) L'ensemble PT de toutes les transitions paramétrées est défini par le descripteur de paramétrisation Pp et par l'ensemble T des transitions de vecteurs.</p> <p>9) Une paramétrisation est une correspondance entre un réseau de Petri paramétré (RPP ou PPN en anglais pour Parameterised Petri Net) et un ensemble de RPP représentant tous le même système. Seul le descripteur de paramétrisation, c'est-à-dire le nombre de places et ce qu'elles représentent, change entre ces RPP.</p>
--

Le paragraphe suivant met en œuvre ces définitions sur un exemple.

III.3.1.2 Exemple d'utilisation de réseaux de Petri paramétrés

L'exemple suivant reprend les concepts du désormais traditionnel problème du dîner des philosophes³ (qui a été modélisé par RPP par [Agimont 96]) : six personnes disposent chacune d'un ordinateur. Elles doivent scanner des documents, les imprimer (elles les photocopient donc numériquement) et classer les documents scannés. Le problème vient du fait qu'il n'existe que trois scanners et trois imprimantes.

Ce problème, tout comme le problème du dîner des philosophes peut être modélisé par des réseaux de Petri classiques, mais les réseaux de Petri paramétrés permettent de simplifier l'écriture en augmentant la représentation abstraite du problème.

Un RPP est donc défini par $RPP = \{C, D, P_p, T, I, O\}$

$C = \{CV_1, CV_2\}$ est l'ensemble des paramètres avec

$CV_1 = \{ordi_0, ordi_1, ordi_2, ordi_3, ordi_4, ordi_5, scan_0, scan_1, scan_2, imp_0, imp_1, imp_2\}$, l'ensemble des objets (les 6 ordinateurs, les 3 scanners et les 3 imprimantes).

$CV_2 = \{gestion, copie, libre, occupé\}$, l'ensemble des valeurs que peuvent prendre ces objets.

$D = CV_1 \times CV_2$ est l'ensemble des combinaisons possibles entre un objet et sa valeur, ce qui correspond à l'ensemble des vecteurs. Il y en a en tout 24 caractérisant le fait qu'un ordinateur gère les documents ou les copie et le fait qu'un scanner, ou qu'une imprimante, soit occupé ou libre.

Il existe, d'après la définition, plusieurs représentations possibles du même réseau. Ceci dépend de la paramétrisation, c'est-à-dire du nombre de places paramétrées. Les figures III.11, III.12 et III.13 en sont trois représentations. Par contre, les transitions de vecteurs sont uniques. Du problème ressortent deux actions : photocopier et classer (gérer) les documents. Si les imprimantes et scanners sont intercalés entre les ordinateurs de la façon suivante, deux transitions de vecteurs sont alors définies :

Si les objets sont classés selon l'ordre suivant :

$ordi_0, imp_0, ordi_1, scan_0, ordi_2, imp_1, ordi_3, scan_1, ordi_4, imp_2, ordi_5, scan_2$

³ Dans le problème du dîner des philosophes, cinq philosophes sont assis autour d'une table pour y manger des pâtes de façon traditionnelle, c'est-à-dire avec deux fourchettes. Le problème est qu'ils doivent se partager les cinq fourchettes présentes de façon optimale. Si un philosophe ne peut manger, il se met à penser.

Les transitions de vecteurs sont :

lancer la photocopie:

$$I(pt_i) = \{ (ordi_j, gestion), (scan_{(j-1) \bmod 2}, libre), (imp_{j \bmod 2}, libre) \}$$

$$O(pt_i) = \{ (ordi_j, copie), (scan_{(j-1) \bmod 2}, occupé), (imp_{j \bmod 2}, occupé) \}$$

pour $i \in [0,5]$ (car la règle s'applique aux six ordinateurs) et $j = i \bmod 2$

lancer le classement:

$$I(pt_i) = \{ (ordi_j, copie), (scan_{(j-1) \bmod 2}, occupé), (imp_{j \bmod 2}, occupé) \}$$

$$O(pt_i) = \{ (ordi_j, gestion), (scan_{(j-1) \bmod 2}, libre), (imp_{j \bmod 2}, libre) \}$$

pour $i \in [6,11]$ et $j = i \bmod 2$

Il existe donc plusieurs représentations dépendantes du nombre de places paramétrées. Une place étant un sous-ensemble de D, elle peut ne contenir qu'un seul élément de D. Une telle place correspond donc à l'état d'un objet défini par un vecteur. La représentation la plus détaillée correspond donc à 24 places paramétrées, comme le montre la figure III.11.

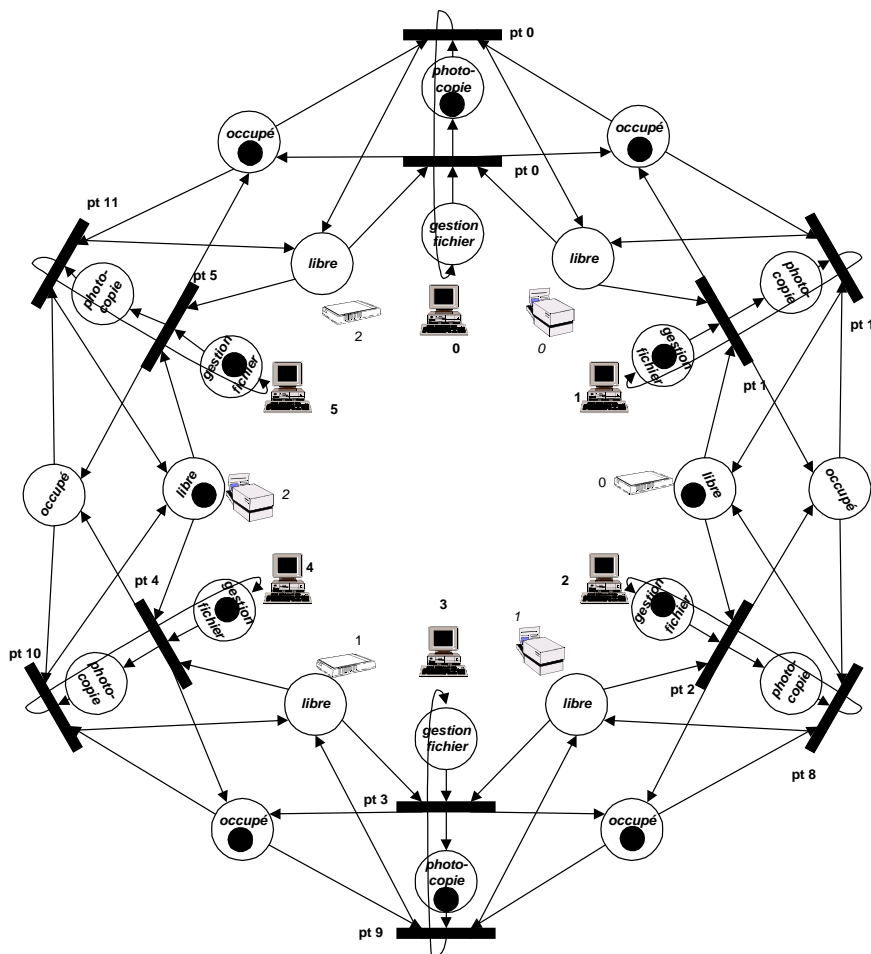


Figure III.11. Représentation détaillée au maximum

Dans la figure III.11 sont également représentées 24 transitions paramétrées, chacune ne contenant qu'une seule transition de vecteur.

Il est possible de regrouper les objets par classes et de définir ainsi trois places contenant les vecteurs paramétrés qui correspondent aux trois types d'objets (ordinateurs, scanners et imprimantes) associés à leurs caractéristiques (fig. III.13). Les vecteurs de transitions (lancer la photocopie et lancer le classement) consomment et produisent chacun le même nombre de vecteurs ; ils peuvent également être regroupés. Ainsi dans la figure III.12, deux transitions paramétrées contiennent les deux types de transitions de vecteurs.

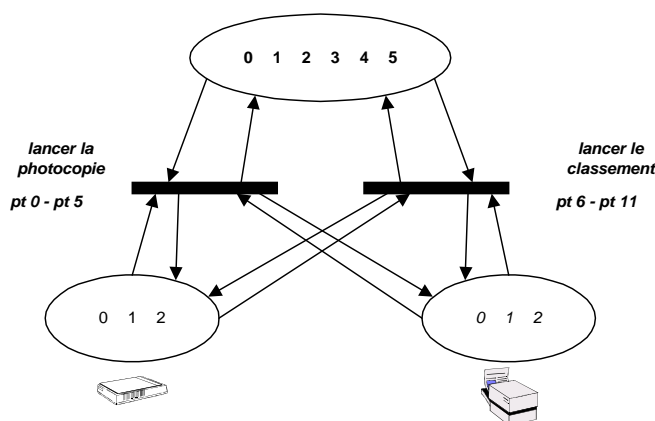


Figure III.12. Représentation moyennement abstraite du problème

La figure III.13 représente la plus grande abstraction du problème. Tous les vecteurs paramétrés sont contenus dans une unique place. Pour les mêmes raisons que précédemment, les vecteurs de transitions peuvent être réunis en une seule transition.

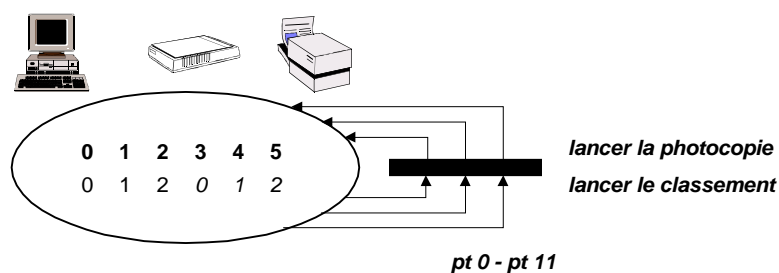


Figure III.13. Représentation du problème au degré d'abstraction maximum

Si la paramétrisation permet donc de simplifier un réseau, le problème est de se limiter dans la volonté d'abstraire le système modélisé afin d'obtenir un réseau lisible. En effet, le réseau représenté dans la figure III.13 est beaucoup moins explicite que le réseau de la figure III.12 qui offre un bon compromis entre l'abstraction et la représentation détaillée du système.

Cette capacité d'abstraction permet de simplifier la modélisation des processus coopératifs. En effet, les flux de données et d'acteurs peuvent être représentés par les réseaux de Petri classiques ou de haut niveau. Néanmoins, la représentation devient vite complexe lorsque des mécanismes tels que les interruptions ou les changements d'activités des acteurs entrent en jeu.

III.3.2 Application des réseaux de Petri paramétrés à la représentation des interruptions

D'après les figures III.9 et III.10, ce sont les interruptions qui compliquent la modélisation des processus coopératifs. L'abstraction permise par l'utilisation d'un réseau de Petri paramétré doit permettre de le simplifier.

Ce réseau doit permettre de représenter les acteurs, les documents sur lesquels travaillent les acteurs, les interruptions et le schéma d'action suivant : lorsqu'un acteur, gérant une activité x , est interrompu pour traiter une activité y , il stoppe son activité pour gérer cette interruption (il suspend alors son état, et passe de l'état *en_activité_x* à l'état *en_activité_y*). Il retourne ensuite à son activité initiale après avoir géré cette interruption (il restaure son état). Le changement d'activité d'un acteur peut également bénéficier d'une abstraction.

A l'instar du précédent réseau, le réseau de Petri paramétré permettant de gérer les interruptions contient deux paramètres. Un paramètre contenant l'ensemble des objets (documents, interruptions, représentations des acteurs), et un paramètre contenant l'ensemble des valeurs associées [Adam 98a].

Le réseau de Petri paramétré est maintenant défini par $RPP = \{C, D, Pp, T, I, O\}$, avec

$$C = \{CV_1, CV_2\} \text{ et}$$

$$CV_1 = \{a_1, \dots, a_{|nb_acteurs|}, doc_1, \dots, doc_{|nb_documents|}, int_1, \dots, int_{|nb_interruptions|}\}$$

$$CV_2 = \{activité_1, \dots, activité_{|nb_activités|}, en_attente, en_traitement, interrompu, traité\}$$

D , l'ensemble des vecteurs paramétrés possibles est déterminé par :

$$D = \{(a_1, activité_1), (a_1, activité_i), \dots, (doc_1, en_attente), (doc_1, en_traitement), (doc_1, interrompu), (doc_1, traité), \dots, (int_1, activité_1), (int_1, activité_i), \dots\}$$

Les vecteurs de transitions sont définis autour de trois règles :

Stocker l'état : cette règle est déclenchée à l'arrivée d'une interruption ; une interruption consiste à traiter l'arrivée d'un nouveau document (papier, électronique

ou verbal dans le cas d'un appel téléphonique). Dans le cas où l'interruption intervient lorsque l'acteur réalise une activité k , l'acteur reçoit un nouveau document à traiter. Le document en cours devient alors interrompu, le nouveau document passe de l'état "en attente de traitement" à l'état "en traitement". L'état initial de l'acteur est stocké dans l'objet interruption. Les transitions de vecteur associées sont :

$$I(pt_i) = \{ (acteur_a, activité_k), (doc_d, en_traitement), (doc_{d'}, en_attente) \}$$

$$O(pt_i) = \{ (acteur_a, activité_k), (doc_d, interrompu), (doc_{d'}, en_traitement), (int_i, activité_k) \}$$

pour $i \in [1, nb_interruptions]$, $a \in [1, nb_acteurs]$, $d \in [1, nb_docs]$.

Restaurer l'état : cette règle est déclenchée lorsque l'acteur a terminé de gérer une interruption ; le document interrompu est alors repris et l'acteur retourne à son activité initiale. Ce qui donne :

$$I(pt_i) = \{ (acteur_a, activité_k), (doc_d, interrompu), (doc_{d'}, complété), (int_i, activité_k) \}$$

$$O(pt_i) = \{ (acteur_a, activité_k), (doc_d, en_traitement) \}$$

Pour $i \in [nb_interruptions, 2 \times nb_interruptions]$, $a \in [1, nb_acteurs]$,
 $d \in [1, nb_docs]$.

Changer d'état : cette règle est déclenchée lorsqu'un acteur termine une activité, et a complété un document, l'acteur prend alors en charge une nouvelle activité correspondant au traitement d'un autre document. Les transitions de vecteurs sont plus simples que dans les deux autres cas :

$$I(pt_i) = \{ (acteur_a, activité_k), (doc_d, complété) \}$$

$$O(pt_i) = \{ (acteur_a, activité_k), (doc_{d'}, en_traitement) \}$$

pour $i \in [2 \times nb_interruptions, 2 \times nb_interruptions + nb_activités]$,
 $a \in [1, nb_acteurs]$, $d \in [1, nb_docs]$.

Ces transitions de vecteurs peuvent être regroupées en trois transitions paramétrées. C'est d'ailleurs le degré maximum d'abstractions autorisées par la définition car le nombre de vecteurs produits et le nombre des vecteurs consommés diffèrent entre les trois règles.

Afin de coupler l'efficacité des workflow nets et l'abstraction des réseaux de Petri paramétrés, la structure générale du réseau est gardée, mais les places sont "doublées". En effet, les places représentent alors les espaces de travail dans lesquels se trouvent les acteurs et les documents. Chacune de ces places principales est associée à une place qui contient

l'ensemble des interruptions. Elles agissent en quelque sorte comme des mémoires de travail. La figure III.14 représente une double place modélisant un espace de travail.

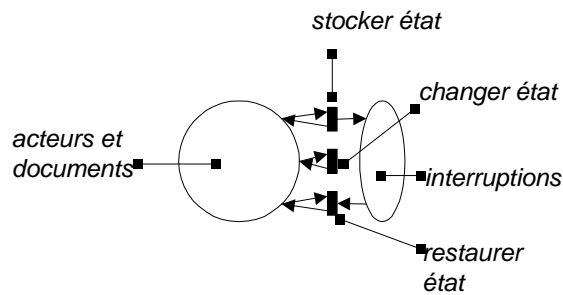


Figure III.14. Représentation d'un espace de travail par réseau de Petri paramétré

La figure III.15 donne un exemple d'utilisation de ces places paramétrées.

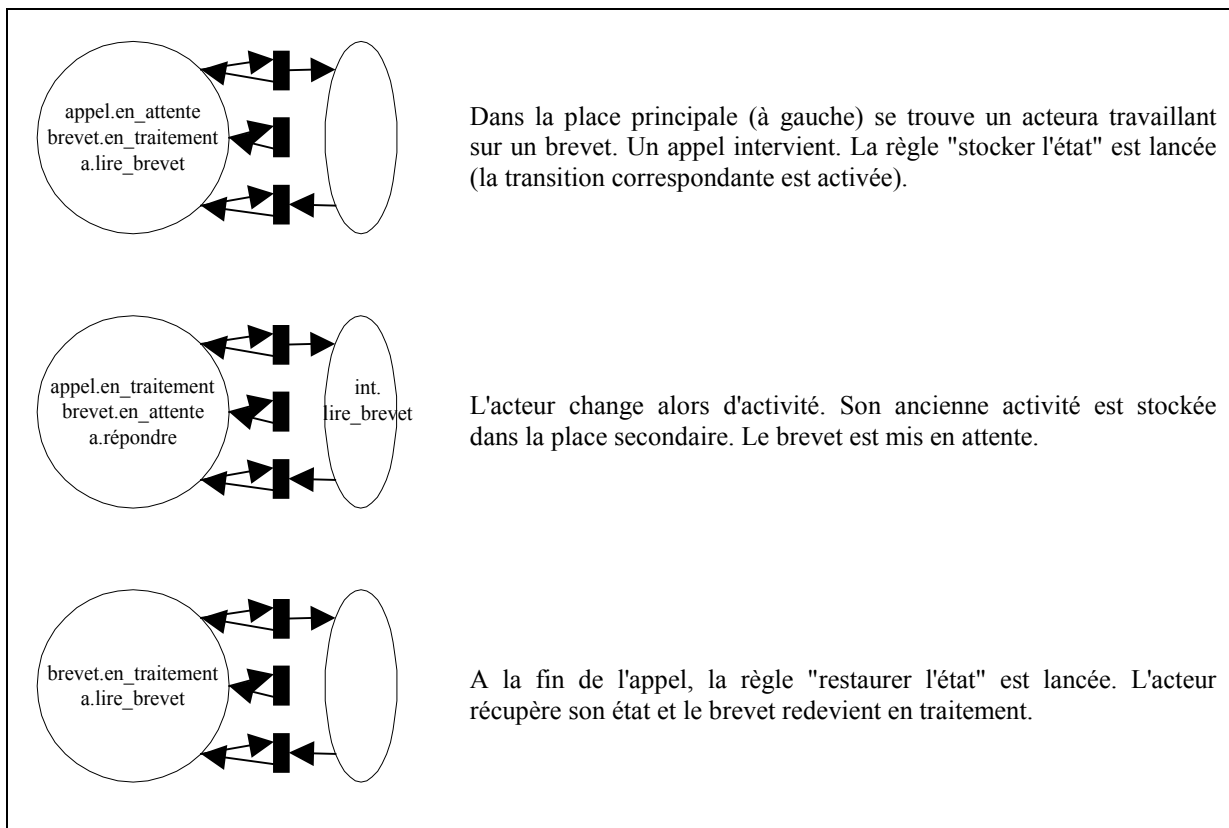


Figure III.15. Exemple de traitement d'une interruption

En pratique, l'interruption d'une activité pour une autre dépend de l'acteur, c'est-à-dire les règles de priorités sont définies pour chaque acteur.

En ce qui concerne la représentation d'une procédure, les places sont organisées à l'image du modèle de traitements (fig. III.16), c'est-à-dire par colonne, par ordre croissant de responsabilité de gauche à droite.

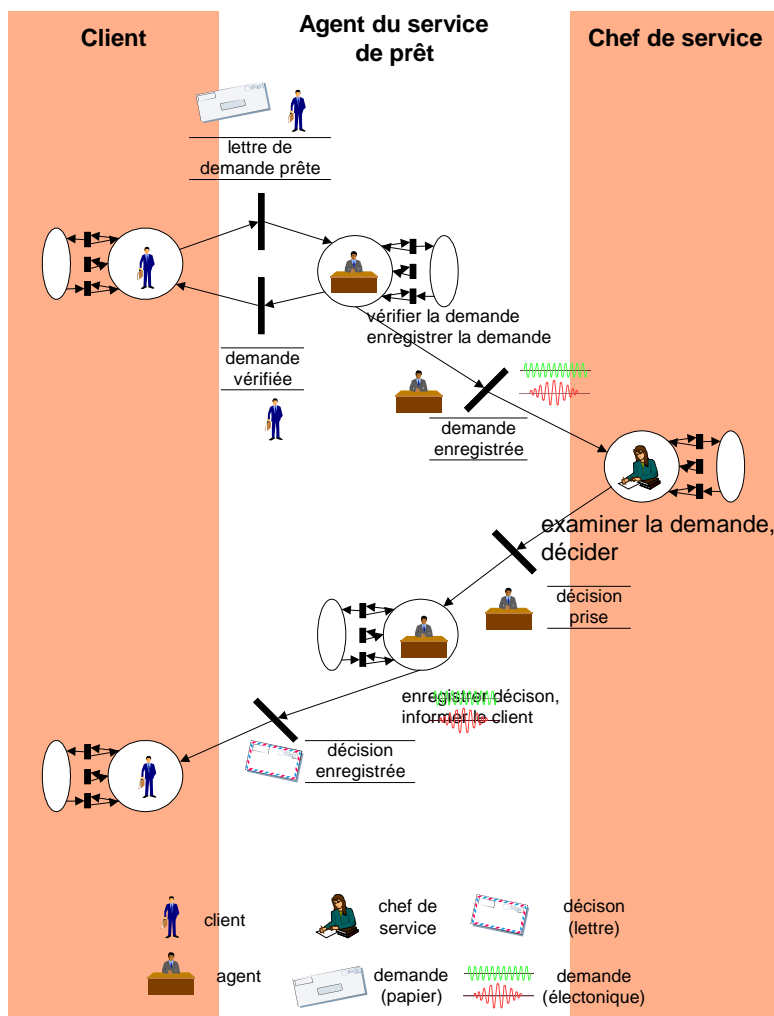


Figure III.16. Extrait d'un modèle dynamique

La figure III.16 représente un extrait de modèle dynamique d'une demande de prêt. Les "doubles places" représentent des espaces de travail. Dans cet exemple, le client se déplace au guichet pour déposer une demande de prêt, si cette demande est correctement remplie, il retourne à son lieu de travail. Le guichetier porte cette demande au contrôleur qui décide de la faisabilité du prêt. Le guichetier retourne à son lieu de travail avec la réponse et l'envoie au client. Dans cet exemple, apparaissent plusieurs fois les doubles places relatives au guichet et à la localisation du client. En effet, pour des raisons de lisibilité, le réseau a été déplié dans le temps afin de suivre la chronologie du processus modélisé.

Bien que plus synthétique, la représentation et surtout la visualisation de la dynamique du réseau paramétré, à l'exemple de tout modèle dynamique, n'est pas évidente. L'utilisation d'un simulateur est nécessaire.

III.3.3 La simulation de procédures administratives complexes

Un simulateur permet de suivre l'évolution d'un processus dans le temps de façon interactive et surtout de façon plus évidente pour les non-experts, ce qui répond à l'objectif d'intégration des utilisateurs dans le cycle de développement. De plus la simulation permet de tester interactivement de nouvelles solutions organisationnelles. Mais la fonction première de cette simulation est qu'elle permet d'obtenir les règles de fonctionnement d'un processus à partir desquelles une spécification du système d'information distribué peut être envisagée. Trois types de règles sont nécessaires :

- Les premières règles correspondent aux règles personnelles de descriptions d'actions. En effet, les acteurs sont libres de choisir les activités qui leur semblent les plus importantes en fonction des documents et des autres acteurs présents dans l'espace de travail.
- Puis, au niveau local, au niveau des espaces de travail, il peut exister des protocoles d'accord entre les acteurs (par exemple : si un fax arrive, c'est la première personne libre qui doit gérer le fax).
- Enfin au niveau général, des règles doivent décrire les flux de données entre les acteurs du système. Ces règles correspondent aux règles de circulation (dans l'exemple présenté à la figure III.16, une règle peut être : si la décision est enregistrée, alors envoyer une lettre au client).

Le simulateur doit permettre de représenter un réseau de Petri objet défini comme un ensemble de transitions, possédant chacune des places en entrée et des places en sortie. Chaque place peut contenir des acteurs (humains ou non) ainsi que des informations ou des interruptions [Adam 97].

Les transitions, les places et les acteurs possèdent donc des règles, considérées comme des listes d'activités. Chaque activité est composée de préconditions et d'actions exécutées si les préconditions sont validées⁴. La validation d'une précondition s'effectue si un jeton (acteur ou document ou interruption) existe dans un état donné dans une place. La réalisation de l'action consiste selon le cas à modifier l'état d'un jeton (par exemple, le faire passer de l'état

⁴ Valette R. Les réseaux de Petri. Polycopiés de cours. Université de Toulouse I. LAAS-CNRS. 1995.

"en attente" à l'état "en traitement"), le déplacer, le supprimer ou encore à créer un nouveau jeton.

A chaque étape de la simulation, où le temps est défini par une suite d'événements réguliers (c'est également le parti pris lors de la définition formelle des systèmes holoniques au chapitre I), les parties actives du réseau sont testées, puis exécutées si elles sont validées.

Ce type de simulateur permet donc de suivre l'évolution d'un processus coopératif tout en représentant les acteurs, leurs actions et leurs déplacements possibles dans l'organisation. Il doit être conçu afin d'être utilisé par les acteurs de l'organisation, permettant ainsi :

- d'intégrer les acteurs dans la mise en œuvre de la solution,
- de valider les résultats issus de la modélisation (par comparaison de la simulation avec le fonctionnement réel de l'organisation humaine),
- d'obtenir les règles de fonctionnement de l'organisation structurées en trois niveaux (global correspondant à la circulation des données et des acteurs, local correspondant aux protocoles d'accord, personnel correspondant aux tâches). Ces règles constituent les règles d'assistances du système final d'aide au travail coopératif.

III.4 Les différentes phases de la méthode MAMOSACO

Une méthode permettant de mettre en place un système d'aide au travail coopératif dans des organisations administratives complexes nécessite d'intégrer les acteurs dans le cycle de vie, particulièrement lors des phases d'analyse, de modélisation et de simulation.

C'est pourquoi MAMOSACO utilise les trois axes classiques de modélisation : un modèle de données, un modèle de flux de données et un modèle de traitement. Un quatrième modèle, le modèle dynamique, est nécessaire pour la prise en compte de la dynamique des organisations humaines. Ces modèles doivent également représenter les notions de responsabilité et de coopération entre les acteurs de l'organisation humaine étudiée. Afin d'intégrer ces acteurs, les modèles doivent être clairs, c'est-à-dire accessibles non seulement aux modélisateurs, mais surtout aux acteurs, aux futurs utilisateurs du système d'aide.

C'est toujours dans une volonté d'implication des utilisateurs que la conception d'un simulateur est recommandée. De plus, le simulateur permet d'extraire les règles de

fonctionnement des processus de l'organisation afin de les réutiliser dans le système final d'information distribué, basé sur une organisation multi-agent holonique.

Cependant, avant de spécifier un système d'aide au travail coopératif dans les processus de l'organisation, il est opportun de vérifier si ces processus ne nécessitent pas un aménagement (on parle aussi d'optimisation) préalable. Pour ce travail, outre l'informaticien (au sens du concepteur de logiciel), des décideurs, des acteurs concernés et des spécialistes des sciences cognitives (psychologues du travail par exemple) sont nécessairement impliqués (d'après le modèle ∇) dès la phase d'analyse et de modélisation du système réel, c'est-à-dire de l'organisation administrative complexe.

L'optimisation des processus coopératifs est facilitée par la mise en évidence, à l'aide des modèles, d'éventuels problèmes de communication, de coopération et de responsabilité (par exemple : une présence importante de la hiérarchie dans les modèles signifie un manque d'autonomie des acteurs ; un encodage ou un stockage redondant de la même information peut signifier une absence de communication ou de coopération ; des aller-retours d'informations entre plusieurs acteurs peut signifier un problème de communication).

A ce propos, le concept holonique propose d'améliorer les processus en renforçant l'autonomie et la stabilité des acteurs, et la coopération entre eux-ci. Ainsi, il est souvent possible de simplifier la structure organisationnelle des processus avant la spécification et la mise en place du système d'information coopératif intelligent. Les figures III.17 et III.18 constituent deux exemples de simplification. Dans la figure III.17, la présence du responsable, qui contrôle chaque action des acteurs A et B peut être diminuée en augmentant l'autonomie de ces acteurs et plaçant une étape de contrôle à la fin de la série d'actions ; cette simplification dépend du degré de responsabilité que les acteurs sont prêts à recevoir. Dans la figure III.18, les va-et-vient entre les acteurs A et B peuvent être supprimés si ces deux acteurs travaillent de façon coopérative.

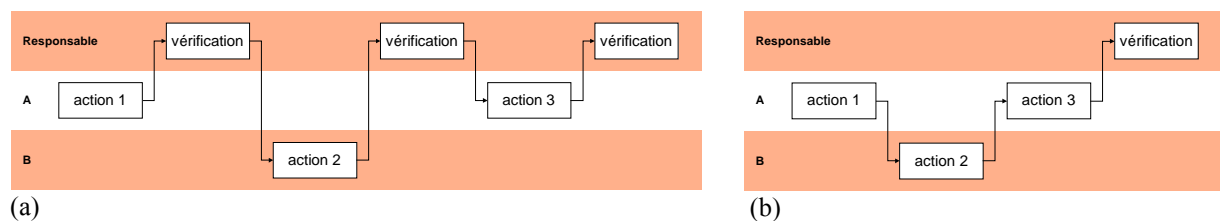


Figure III.17. Exemple de simplification organisationnelle par augmentation de l'autonomie
 a) forte présence du responsable
 b) autonomie renforcée au niveau des acteurs A et B

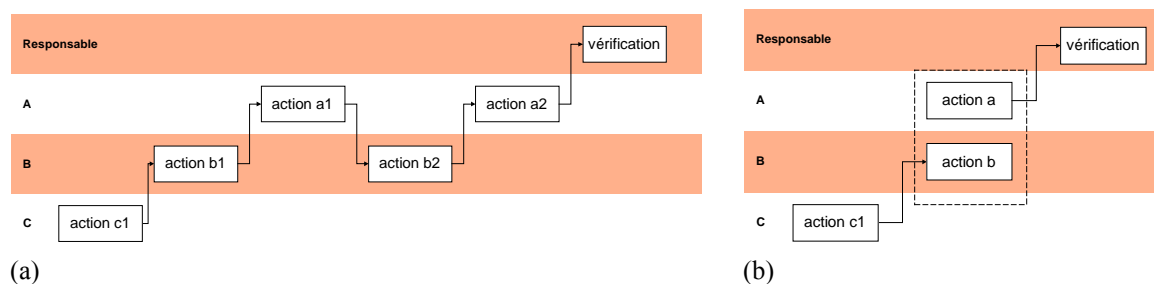


Figure III.18. Exemple de simplification organisationnelle par la coopération
 a) faible coopération entre les acteurs A et B
 b) coopération entre les acteurs A et B (symbolisée par un cadre)

Selon le cas, il n'est donc pas nécessaire d'utiliser tous les modèles. En effet, si le doute est permis sur l'optimalité des processus, il est préférable d'appliquer d'abord les modèles qui permettent de révéler les dysfonctionnements. L'ensemble des modèles est alors appliqué suite à l'optimisation des processus. Enfin, le simulateur est utilisé afin d'obtenir les règles de fonctionnement des processus validés par les acteurs.

L'enchaînement des phases de la méthode MAMOSACO ne suit donc pas un ordre séquentiel. Les acteurs sont consultés lors de chaque phase (analyse, modélisation et simulation), ce qui permet de les valider. Des retours sont possibles, soit à la phase d'analyse pour recueillir des données manquantes, soit à la phase de modélisation en cas de modifications du processus par les acteurs, ou en cas de modification à apporter aux modèles. Les validations des modèles et de la simulation doivent être effectuées sous forme de réunions avec l'ensemble des acteurs impliqués dans les processus modélisés. Ceci afin d'amener les acteurs à partager leurs points de vue sur les modèles et les processus et à proposer coopérativement des améliorations de nature organisationnelle.

De même, l'enchaînement des modèles ne doit pas suivre un ordre strict. En effet, à l'instar de la méthode Merise, des tests de cohérence (syntaxiques et sémantiques) doivent être menés entre les différents modèles. La figure III.19 représente l'enchaînement des phases de la méthode MAMOSACO.

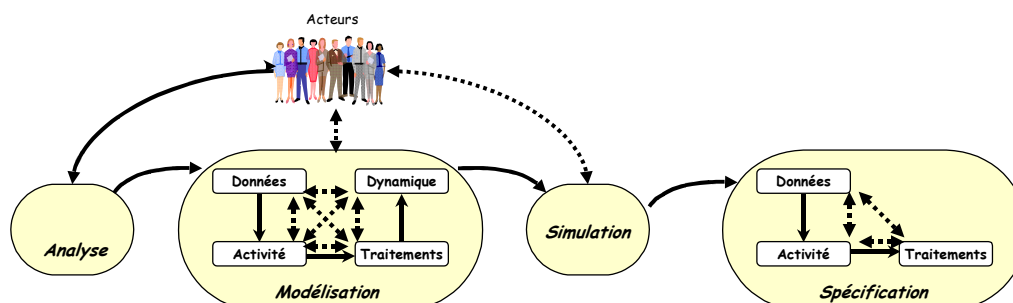


Figure III.19. Enchaînement des phases dans la méthode MAMOSACO [Adam 98b]

La méthode constituée des phases d'analyse, de modélisation et de simulation de systèmes administratifs complexes peut donc être utilisée dans un double but : celui d'amélioration des processus coopératifs étudiés et celui d'aide à la spécification d'un système d'aide au travail coopératif par l'obtention des règles de fonctionnement, sur trois niveaux, du système administratif complexe. C'est à ce titre que la méthode peut être qualifiée d'adaptable.

Dans le chapitre I, il a été proposé de concevoir le système au travail coopératif sous la forme d'une organisation multi-agent holonique. Or, la méthode MAMOSACO est justement conçue pour modéliser le fonctionnement d'organisations composées d'acteurs ayant des niveaux de responsabilité différents et s'échangeant des données. Il est donc possible de réutiliser les formalismes pour spécifier le fonctionnement coopératif du système d'aide au travail coopératif basé sur une architecture holonique. La partie suivante propose la définition et la spécification du système multi-agent holonique en décrivant tout d'abord les rôles joués par les agents, puis le fonctionnement de l'ensemble par la méthode MAMOSACO.

III.5 Spécification de Système Orienté Holon pour l'aide au Travail Coopératif (SOHTCO)

En partant de l'hypothèse qu'il se trouve dorénavant dans tout système administratif au moins un poste de travail (PC ou autre) par bureau ou local, il est alors envisageable de répartir le système multi-agents holonique autour de chaque acteur de ces systèmes administratifs. Par l'utilisation de règles d'assistance, ce système doit les conseiller dans leurs activités et faciliter la coopération en faisant apparaître la notion de groupe. Le but étant d'augmenter à la fois l'autonomie des acteurs par une aide appropriée, la coopération par la prise de conscience de la nature commune des activités et la stabilité des acteurs par une veille des actions. L'objectif est en effet d'obtenir des organisations administratives « plus holoniques », condition nécessaire de viabilité et de pérennité (au sens de A. Koestler). Ce Système Orienté Holon pour l'aide au Travail Coopératif est, par la suite, appelé SOHTCO (un SOHTCO étant dédié à une procédure administrative impliquant k acteurs sur les n de l'organisation) [Adam 99a], [Adam 99b].

III.5.1 Règles d'assistance

Afin de garder une cohérence dans un système distribué, de nombreux travaux ont montré que l'utilisation d'au moins deux types de règles est nécessaire : des règles individuelles au

niveau de chaque module et des règles sociales définissant les interactions possibles entre les modules. Par exemple, Boissier et Demazeau proposent une architecture pour un contrôle social et individuel d'un système d'agents destiné à la reconnaissance de forme [Boissier 96].

Dans le cas d'un système multi-agent assistant les acteurs de processus coopératifs, trois niveaux de règles sont nécessaires.

- Pour veiller à la cohérence du fonctionnement d'une procédure d'une organisation administrative, le SOHTCO doit posséder un premier niveau constitué de règles générales décrivant la circulation des données entre ses acteurs (ce qui correspond aux règles sociales).

Puis, la phase de modélisation de la dynamique et la phase de simulation a montré que deux autres niveaux de règles d'assistances sont requis :

- au niveau de chaque espace de travail, de chaque bureau, le SOHTCO doit posséder un ensemble de règles locales qui correspondent aux protocoles d'accord entre personnes situées dans le même espace, et aux communications de données avec les bureaux précédents ou suivants dans le processus,

- au dernier niveau, le SOHTCO doit posséder les règles de conduite personnelles des acteurs associés aux postes de travail (règles individuelles). Ces règles ne sont pas fixées, mais représentent les actions généralement entreprises par les acteurs dans des situations bien définies. Ce dernier niveau contient également les règles de transmission de données avec les acteurs voisins dans le processus, c'est-à-dire ceux dont l'acteur associé au poste reçoit les données et ceux auxquels il envoie les données.

Les règles d'assistance générales, locales et personnelles sont issues directement de la simulation des processus coopératifs. Les règles de communications d'informations entourant les espaces de travail (les bureaux) et les postes de travail (les acteurs) sont extraites des règles de circulations générales. Le tableau III.2 présente des exemples de règles d'assistances.

Tableau III.2. Exemples de règles d'assistance

Exemple de règle générale	Exemple de règle liée à un bureau (ici, le bureau administratif)	Exemple de règle personnelle (associée à un conseil en brevet).
Si le brevet est prêt dans le bureau de l'inventeur, Alors le brevet est transmis au bureau administratif.	Si une demande de dossier arrive, Alors un(e) secrétaire libre traite la demande.	Si un dossier technique est en attente, Alors un avis doit être rédigé.

Ces trois niveaux de règles ne reflètent pas réellement la structure du SOHTCO car pour un bureau donné, les règles locales doivent se trouver sur chaque poste de travail qu'il contient. Ainsi chaque acteur présent peut être averti de l'arrivée d'une donnée et, selon son état et selon les protocoles, être chargé de traiter cette donnée. Les trois niveaux de règles forment donc deux couches du SOHTCO, une couche de responsables de processus coopératifs et une couche de responsables de poste. Cependant, pour répondre aux concepts holoniques, le SOHTCO doit être formé de trois couches comme le montre la partie suivante.

III.5.2 Structure de l'organisation

Le SOHTCO a pour objectif d'assister les acteurs d'une organisation administrative complexe dans les différents processus coopératifs dans lesquels ils sont impliqués.

Selon les propriétés des systèmes holoniques énoncées au chapitre I, la base du SOHTCO constitue son interface avec son environnement qui est composé d'acteurs et de documents. La première couche doit donc contenir des agents holoniques responsables des interactions avec l'utilisateur et de la gestion des documents. De plus, afin de faciliter et de renforcer les communications entre les postes de travail, cette première couche doit également contenir les agents holoniques responsables de l'émission et de réception de messages électroniques (par la suite, le symbole \triangle désignera un agent exécutant situé à la base du SOHTCO d'architecture pyramidale).

La seconde couche d'agents holoniques est constituée d'agents responsables des actions des agents exécutants. Ces agents 'responsables de poste' sont dédiés chacun à un acteur dans le cadre d'une procédure donnée. Ces agents contiennent donc les règles personnelles et locales du fonctionnement de la procédure (ces responsables de poste sont notés \triangle).

Afin de maintenir la cohérence de l'aide apportée dans le cadre d'un processus coopératif, la troisième et dernière couche est composée d'un d'agent holonique responsable du processus. Cet agent holonique responsable de procédure est associé à l'acteur humain responsable de la

procédure et contient les règles générales du fonctionnement de la procédure (un responsable de procédure, situé au troisième niveau est noté Δ).

Chaque agent responsable (le responsable de couche comme le responsable de procédure), doit assister un acteur. Il doit donc pouvoir interagir avec lui, l'aider à gérer ses documents, et communiquer avec les autres responsables (c'est-à-dire émettre et recevoir des informations). Chaque agent responsable a donc sous son contrôle des agents exécutants regroupés en quatre groupes : les responsables de l'interface avec l'utilisateur, les responsables d'envois de données, les responsables de réceptions de données et les responsables de la gestion de données. Ces ensembles, constitués d'agents exécutants et de leur responsable, sont appelés sous-SOHTCOs. La figure III.20 représente un sous-SOHTCO (noté Δ) qui possède quatre agents exécutants.

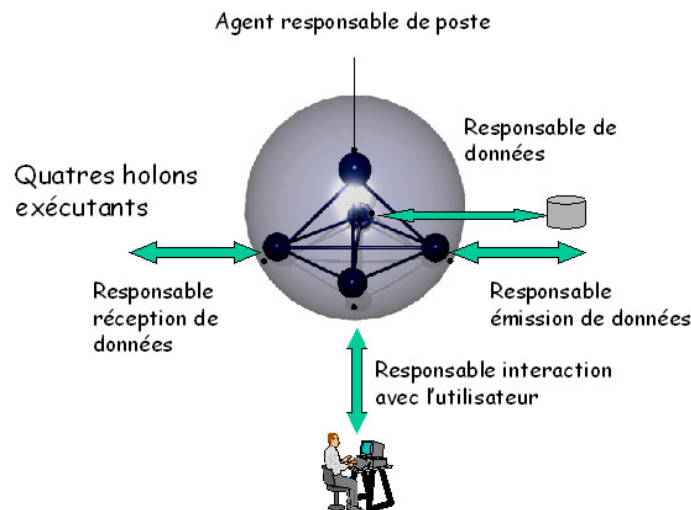


Figure III.20. L'architecture d'un sous-SOHTCO

En fait, il est possible de représenter tous les sous-SOHTCO de la même façon. En effet, selon le principe de regroupement holonique présenté au chapitre I, tout ensemble d'agents holonique peut être considéré comme un agent holonique. Ainsi, l'ensemble des agents exécutants responsables d'une même action peut être considéré comme un holon responsable de cette action. Un sous-SOHTCO est donc toujours composé d'un agent holonique responsable des actions de quatre agents holoniques, non nécessairement atomiques, responsables de l'interaction avec l'utilisateur et des actions de gestion, d'émission, de réception de données.

La figure III.21 représente l'architecture d'un SOHTCO, composé de cinq Δ , assistant des acteurs d'une organisation humaine. Le sous-SOHTCO (Δ) correspondant au responsable

holonique de procédure (Δ) est associé au responsable humain de la procédure. Chaque autre acteur humain est assisté par un \triangle qui possède une représentation des activités que l'acteur peut entreprendre et qui peut donc l'assister dans son travail.

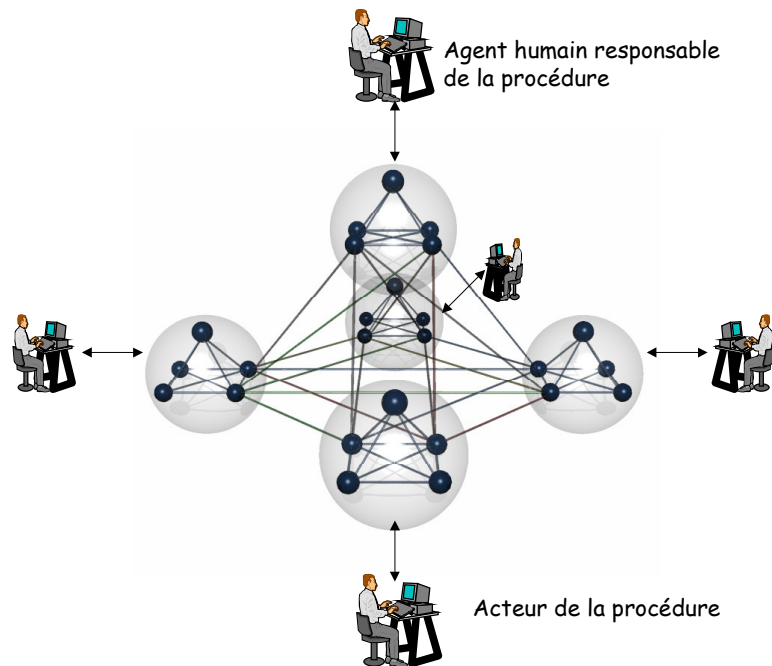


Figure III.21. L'architecture d'un SOHTCO

Dans l'architecture d'un SOHTCO proposée dans la figure III.21, le responsable de procédure (Δ) ne communique pas directement avec les responsables de poste (Δ), mais par l'intermédiaire des agents exécutants (\triangle) responsables d'émission et de réception de données. Ceci ne constitue pas un manquement aux règles holoniques de communications hiérarchiques de données présentées au chapitre I. En effet, chaque sous-SOHTCO (\triangle) peut être considéré comme un holon ; la figure III.21 présente alors la communication hiérarchique entre le holon responsable de procédure et les holons de plus bas niveau, responsables de poste.

Le SOHTCO est donc un système multi-agent holonique composé de trois niveaux, regroupant chacun des agents jouant le même rôle : le premier niveau contient l'agent jouant le rôle du responsable de procédure, le deuxième niveau contient les agents jouant les rôles de responsables de poste et le dernier niveau contient les agents jouant les rôles d'agents exécutants. La partie suivante propose, à partir de cette définition, une modélisation et une spécification du SOHTCO en deux étapes, tout d'abord en détaillant les rôles joués par les agents, puis en spécifiant le comportement coopératif de l'ensemble du SOHTCO.

III.5.3 Modélisation et Spécification du SOHTCO

Si la définition de la structure du SOHTCO a été facilitée par l'utilisation des principes holoniques, la modélisation de l'organisation du système et la caractérisation des fonctionnalités des agents restent problématiques. En effet, les travaux apparus à ce sujet sont surtout théoriques. Seuls quelques récents travaux permettent la modélisation d'organisations dans un but applicatif. Citons dans ce cadre CASSIOPEE [Collinot 98], méthode permettant la modélisation d'un système multi-agent et de son organisation. Mais cette méthode est surtout dédiée à la robotique collective, plus qu'à la représentation de la coopération dans une organisation holonique où les agents ont des degrés de responsabilités différents.

La modélisation et la spécification doivent s'effectuer en deux étapes : la première concerne le fonctionnement individuel de chaque type d'agent holonique ; la seconde concerne le fonctionnement de groupe, décrivant les communications dans l'ensemble du système.

III.5.3.1 Description individuelle des agents holoniques

Avant de spécifier le fonctionnement coopératif du SOHTCO, il faut détailler les caractéristiques des trois types d'agent, à savoir le rôle responsable de procédure (\triangle), le rôle responsable de poste (\triangle), et le rôle exécutant (\triangle). La grille d'analyse fonctionnelle proposée par Ferber [Ferber 95] et présentée au chapitre I permet de caractériser des agents en fonction de leurs fonctionnalités. Il est possible d'adapter cette grille afin de, non seulement identifier les fonctionnalités des agents en fonction des dimensions personnelles, sociales et environnementales, mais aussi de spécifier ces fonctionnalités.

Cette grille de spécification décrit trois dimensions, au lieu des cinq initiales proposées par Ferber. La dimension physique, trop dépendante du système, n'apparaît pas pour cette phase de spécification. La dimension relationnelle, qui concerne la relation d'un agent avec les autres agents, est fusionnée avec la dimension sociale. La fonction conative concerne les besoins, désirs que ne possèdent pas encore nos agents holoniques, elle n'apparaît donc pas dans la grille de spécification.

La grille permet donc de définir pour chaque agent holonique les fonctions relatives:

- Aux connaissances, procédurales ou non (la fonction représentationnelle). Ces connaissances sont plus ou moins détaillées selon la position de l'agent dans le SOHTCO.

- A la planification des actions (la fonction organisationnelle). Chaque agent étant autonome, il doit pouvoir gérer ses propres actions.
- Aux interactions (la fonction interactionnelle). Les agents holoniques doivent effectivement coopérer entre eux et donc interagir.
- A la maintenance (la fonction conservatrice), ceci pour maintenir la stabilité de l'agent et du système en général.
- Aux actions spécifiques au rôle de l'agent (la fonction productive). Par exemple, l'agent effectue l'interface avec l'utilisateur ou gère des données situées sur le poste de travail.

Ces fonctions sont décrites relativement à l'environnement de l'agent, aux autres agents et à l'agent lui-même. Le tableau III.3 représente ces fonctions.

Tableau III.3. Grille de conception adaptée de la grille d'analyse de Ferber [Ferber 95]

Fonctions \ dimensions	<i>Sociale</i>	<i>Environnementale</i>	<i>Personnelle</i>
<i>Représentationnelle</i>	Représentation du groupe, des rôles, des autres	Représentation du monde	Représentation de soi, de ses capacités
<i>Organisationnelle</i>	Planification des actions sociales, des communications	Planification des actions dans l'environnement	Contrôle des planification, méta-planification
<i>Interactionnelle</i>	Description interactions agent-société, performatifs	Mécanismes de perception et d'action par rapport à l'environnement	Auto-communication, auto-action
<i>Productive</i>	Tâches de gestion, d'administration, de coordination, de négociation	Tâches d'analyse, de modification et de création	Auto-modification, apprentissage
<i>Conservative</i>	Conservation de la société, des relations, du réseau d'acointances	Conservation des ressources, défense et entretien du territoire	Conservation de soi, réparation, entretien

Dans une organisation multi-agent holonique, chaque agent possède la même structure, seuls le comportement et les connaissances varient selon la position et le rôle. Les tableaux suivants montrent l'utilisation de la grille de spécification pour la définition des fonctions des trois types d'agents du SOHTCO.

Le tableau III.4 présente l'application de la grille pour la définition du comportement de l'agent responsable de procédure.

Tableau III.4. La définition de l'agent responsable de procédure

Le responsable de procédure Δ

Fonctions \ dimensions	Sociale	Environnementale	Personnelle
Représentationnelle	Il possède une représentation de toute la procédure, il 'connaît' tous les Δ dont il est responsable.	Il connaît les documents manipulés et les acteurs de la procédure.	Il possède un nom, un état, des plans d'actions.
Organisationnelle	Il agit sur les Δ en fonction de sa représentation de la procédure.	Il tient compte du départ et de l'arrivée d'acteurs, il signale les espaces disques pleins et libres.	Il choisit un plan d'action en fonction de l'état de la procédure.
Interactionnelle	Il interagit avec les Δ dont il est responsable.	Il interagit au travers de l'interface utilisateur, et par interrogation du réseau.	Il suit un plan d'action en fonction de son environnement.
Productive	Il veille à ce que les actions des Δ suivent la procédure, à ce qu'ils soient coordonnés.	Il analyse la fonctionnalité du réseau.	Il apprend en augmentant sa représentation de la procédure des actions non définies qu'ont effectués les acteurs.
Conservative	Il veille sur les Δ , recrée un Δ si besoin, veille sur les relations, avertit son responsable humain en cas de panne réseau.	Il demande les sauvegardes de données, tente de restaurer les données défectueuses.	La maintenance du Δ est déferée aux Δ dont il est responsable.

En ce qui concerne le responsable de poste, l'application de la grille entraîne le tableau suivant.

Tableau III.5. La définition de l'agent responsable de poste

Le responsable de sohtco Δ

Fonctions \ dimensions	Social	Environnemental	Personnel
Représentationnel	Il possède une représentation locale de la procédure, il connaît ses voisins directs ainsi que les Δ dont il est responsable.	Il connaît les documents et les acteurs liés au poste sur lequel il est fixé.	Possède une représentation de lui-même et de ses actions.
Organisationnel	Il suit la procédure, planifie les actions des Δ .	Il planifie les actions des Δ en fonction des documents et des demandes des acteurs.	Il choisit un plan d'action en fonction de l'état de la procédure.
Interactionnel	Il communique avec son responsable Δ , avec ses voisins Δ , et les Δ dont il est responsable.	Il interagit avec les acteurs par l'interface, interroge le réseau pour l'existence des documents.	Il suit un plan d'action en fonction de son environnement.
Productif	En fonction des demandes des acteurs, de l'état de la procédure, des messages, il demande aux Δ d'effectuer des actions ou envoie des messages (également aux acteurs).	Il analyse l'état du poste de travail et du réseau en local.	Il apprend en augmentant sa représentation de la procédure des actions non définies qu'a effectué l'acteur.
Conservatif	Il veille sur les Δ , en recrée si besoin, il teste l'existence de son Δ , le recrée après concertation avec tous les Δ . Il signale une perte de liaison à un responsable humain.	Il demande les sauvegardes de données, la restauration de données défectueuses.	La conservation des Δ est déferée à Δ .

Quant à l'agent exécutant, l'application de la grille de spécification permet d'obtenir le résultat du tableau III.6.

Tableau III.6. *La définition de l'agent exécutant*

L'exécutant \triangle

Fonctions \ dimensions	Social	Environnemental	Personnel
Représentationnel	Il connaît son responsable \triangle et ses voisins \triangle	Il possède une représentation des documents du poste de travail auquel il est associé, quelques \triangle possèdent une représentation de l'acteur.	Il possède une connaissance procédurale de sa spécialité.
Organisationnel	Il répond aux objectifs fixés par le \triangle	Il gère ses actions en fonction de l'espace et des documents.	Il classe ses objectifs en fonction du manque de place ou manque de documents.
Interactionnel	Il communique avec son responsable \triangle et ses voisins \triangle .	Selon la spécialité, il interagit avec l'utilisateur, avec le poste de travail.	Le \triangle n'agit pas de lui-même.
Productif	En fonction des demandes de \triangle , il réalise sa spécialité et communique ses résultats à \triangle .	En fonction de sa spécialité, le \triangle produit, modifie, efface des documents.	Le \triangle ne possède pas la capacité de se modifier.
Conservatif	Il vérifie les liens avec ses voisins, communique les problèmes à \triangle .	Il effectue des copies de sauvegarde, tente de restaurer les fichiers défectueux.	La conversation des \triangle est déferée à \triangle .

Ces grilles permettent donc de décrire les actions des agents en fonction de leurs environnements et en fonction des autres agents. Cependant, elles ne permettent pas de définir le fonctionnement de l'ensemble du SOHTCO, c'est-à-dire qu'elles ne permettent pas la définition et la modélisation du fonctionnement coopératif de l'ensemble du système multi-agent.

Ce problème est en partie levé par la structure holonique du système. En effet, l'organisation des agents et les flux de communications étant bien définis, il est possible d'utiliser les formalismes de la méthode MAMOSACO.

III.5.3.2 Fonctionnement coopératif interne du SOHTCO.

La méthode MAMOSACO a été conçue pour la modélisation du fonctionnement coopératif de procédures se déroulant au sein des organisations humaines à structure holonique. Elle peut donc être appliquée pour la modélisation et la spécification de processus coopératifs se déroulant au sein d'une organisation multi-agent à structure holonique.

Des quatre modèles que constitue la méthode, tous ne doivent pas être réutilisés. En effet, le but est surtout de spécifier le fonctionnement coopératif du SOHTCO, non de le modéliser. Les modèles de données et de traitements sont suffisants. En effet, le modèle des traitements fournit plus d'informations que le modèle d'activité, qui est plus destiné à fournir une vue d'ensemble du système qu'il représente.

Le modèle de données permet de représenter la classe principale Holon ainsi que les classes associées aux représentations (représentation de la procédure, de l'acteur, du poste de travail, du responsable, des subordonnés). Chaque agent possède des connaissances sur lui-même, sur ses subordonnés, sur son responsable, sur les agents holoniques voisins avec lesquels il peut et doit interagir, sur l'acteur auquel il est attaché, sur le poste de travail où il opère, sur la procédure à laquelle il est dédié. Il possède également 5 fonctions principales : initialiser (lui permettant d'acquérir les connaissances au lancement du système), planifier (correspondant à la fonction organisationnelle), recevoir et émettre (correspondants à la fonction interactionnelle), agir (correspondant à la fonction productive) et surveiller (correspondant à la fonction conservative).

La figure III.22 représente une modélisation objet des classes définissant le SOHTCO. Les trois types d'agents héritent de la classe Holon. Les quatre types agents exécutantsinstancient la classe des agents exécutants. Les agents responsables de procédure comme les agents responsables de postes sont composés d'au moins un exemplaire de ces agents exécutants.

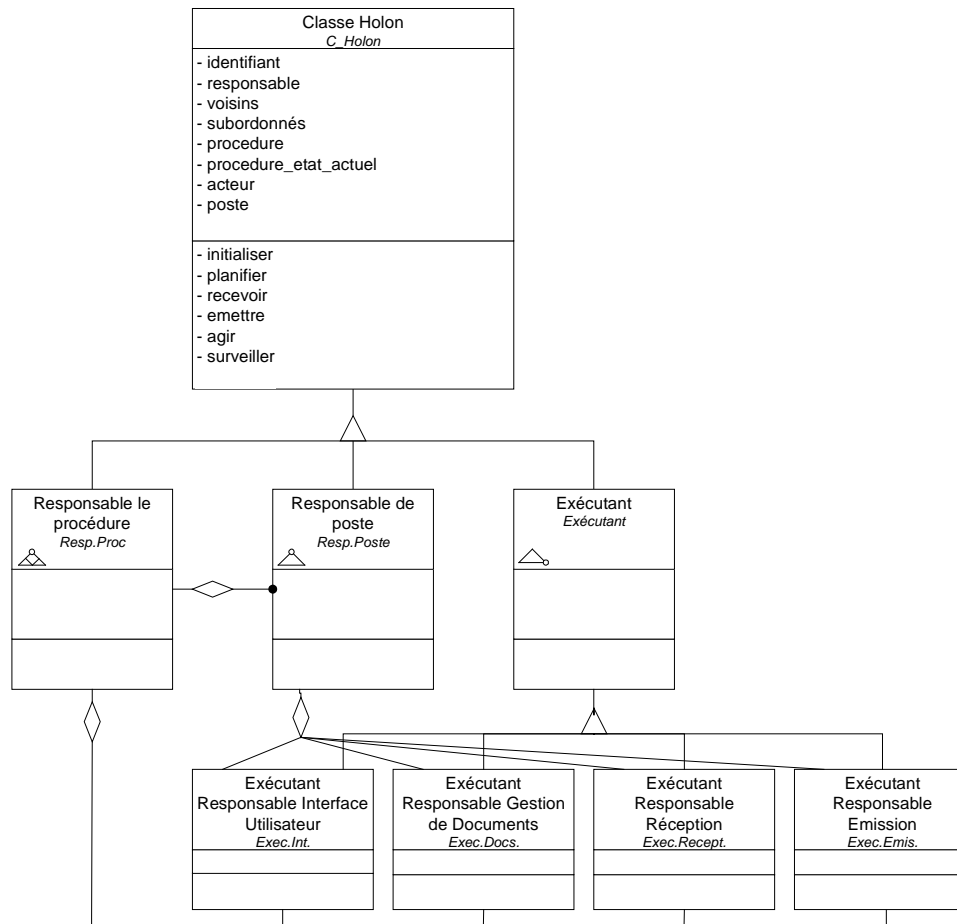


Figure III.22. Extrait d'une modélisation objet du SOHTCO

Le SOHTCO, comme chacune de ses parties, doit répondre à la définition de l'agent holonique, c'est-à-dire qu'il doit pouvoir planifier ses actions. Il doit suivre les émissions et réceptions de données effectuées par les acteurs de l'organisation humaine à laquelle il s'applique. Pour respecter la notion de stabilité, le SOHTCO possède une fonction de maintenance qui effectue une surveillance de chacune de ses parties. La fonction productive du SOHTCO, qui correspond à la fonction agir de la classe Holon, est liée aux besoins de l'organisation humaine. Par exemple, le SOHTCO peut assister les acteurs dans la recherche d'un document ou d'une personne. Une sixième et dernière fonction est nécessaire, il s'agit de la fonction d'initialisation de l'ensemble des parties du SOHTCO.

Ces procédures peuvent être modélisées par le modèle des traitements de la méthode MAMOSACO. Dans ce modèle doivent apparaître, pour une procédure donnée, les différents rôles, classés par niveau de responsabilité. Le SOHTCO interagit avec les acteurs de l'organisation humaine à laquelle il s'applique, leurs représentations sont donc nécessaires dans le modèle. Deux types d'organisations sont alors pris en compte : l'organisation humaine

et l'organisation logicielle. Le SOHTCO a pour objectif d'assister les acteurs de l'organisation humaine, les acteurs doivent donc précéder les agents logiciels en ce qui concerne les degrés de responsabilités. En outre, ceci permet de ne pas créer un amalgame entre les deux organisations.

La figure III.23 montre un extrait de la procédure d'émission de document assistée par le SOHTCO. Dans ce modèle apparaissent les acteurs humains de la procédure qui possèdent, bien sûr, un niveau de responsabilité plus important que les agents logiciels.

Dans cet exemple, l'utilisateur A envoie une information à l'utilisateur B. Il en informe le sous-SOHTCO qui lui est associé au travers de l'agent exécutant responsable de l'interface. Cet agent avertit le responsable de poste. Celui-ci demande alors, si le document est présent, au responsable d'émission de transmettre le document au sous-SOHTCO associé à l'acteur B.

Ce sous-SOHTCO reçoit le document par le responsable de réception qui en informe le responsable de poste. Ce dernier demande au responsable d'interface de signaler l'arrivée du document.

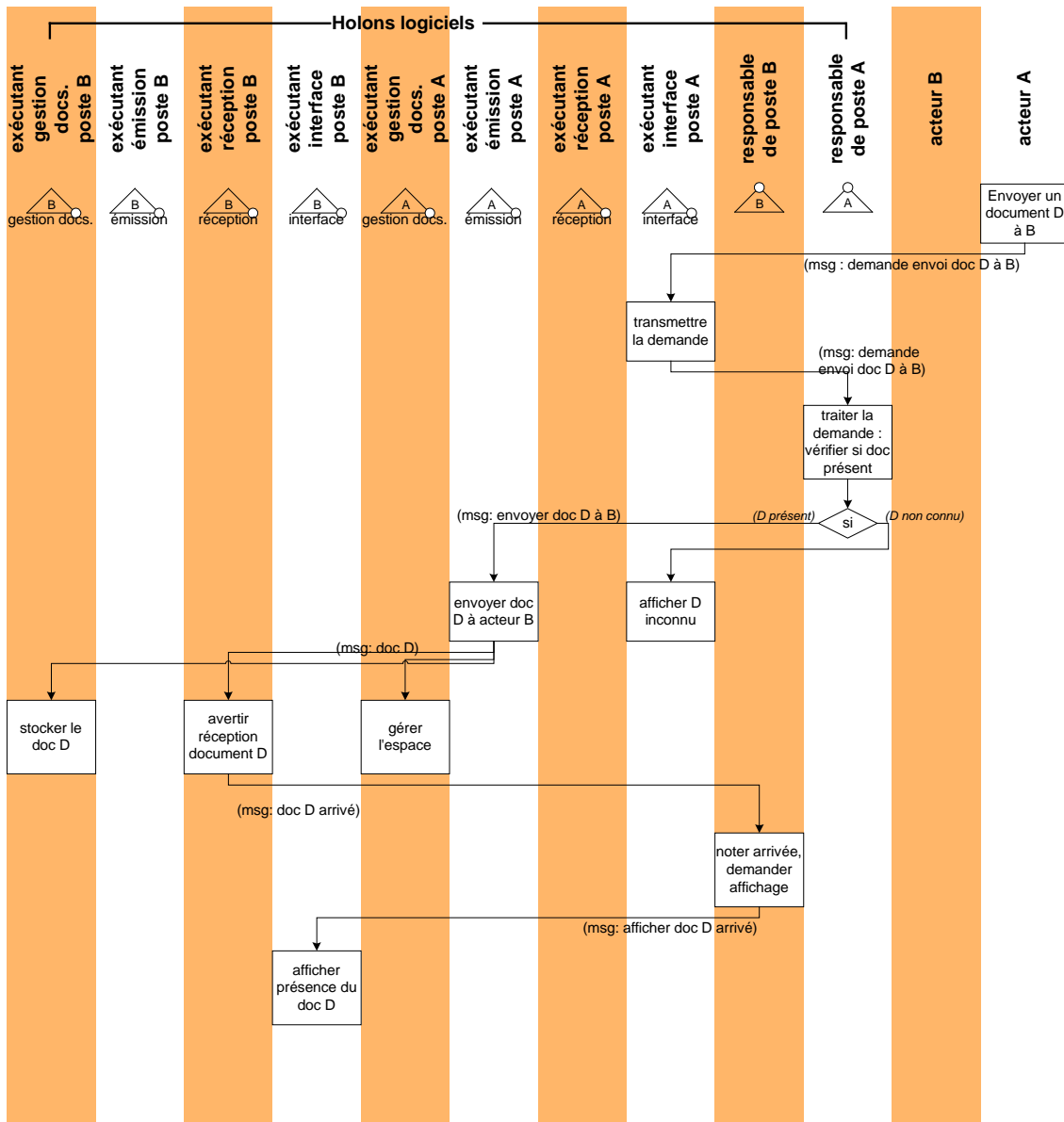


Figure III.23. Exemple de Transmission d'un document assisté par le SOHTCO

La figure III.23 est un exemple de spécification d'une procédure coopérative du SOHTCO. La spécification des différentes procédures dépend de l'organisation humaine étudiée pouvant nécessiter un nombre variable d'agents exécutants.

III.6 Modèle nabla et principe de réutilisabilité dans la méthode MAMOSACO

MAMOSACO se base sur le modèle nabla présenté au début de ce chapitre. Ce modèle présente un principe de réutilisabilité [Bailly 87] concernant les composants d'une interface ou de modules d'aide. Cette base réutilisable est constituée de différents niveaux, qui correspondent aux phases de spécification et de conception. Cette base est décrite en vue du

développement d'interfaces et de modules d'aide. En ce qui concerne le développement du SOHTCO, trois niveaux de réutilisation sont envisageables :

- Le premier niveau concerne la définition des objectifs du SOHTCO, c'est-à-dire quel type d'assistance supplémentaire, outre le suivi des actions, il doit apporter (par exemple l'aide à la recherche de documents). Ce niveau concerne également la spécification individuelle des rôles joués par les agents du SOHTCO.
- Le deuxième niveau concerne le fonctionnement coopératif du SOHTCO. Il contient une base de spécification de procédures coopératives qui reprend les procédures de bases telles que les procédures d'initialisation, de planification, d'émission, de réception de documents, de maintenance. Cette base contient également les différentes procédures coopératives particulières déjà spécifiées telles que la recherche d'un document, d'une personne (pour un contact plus rapide), la classification d'un document dans une base de données entre autres.
- Le dernier niveau concerne le niveau physique, c'est-à-dire les différentes implémentations possibles des éléments du SOHTCO en fonction de l'environnement d'application. Par exemple, ce niveau peut regrouper les implémentations des agents holoniques en Java et en Visual Basic.

Ces différents niveaux de réutilisation sont appelés niveau structurel et niveau individuel, niveau coopératif et niveau environnement. Ils sont identifiés dans la figure III.24.

L'objectif est d'obtenir, suite à plusieurs applications de la méthode MAMOSACO, des bases de composants, de différents niveaux, dépendant des cas applications. Ces bases permettront une réutilisation des composants et donc permettraient un développement de plus en plus rapide de solutions de type SOHTCO.

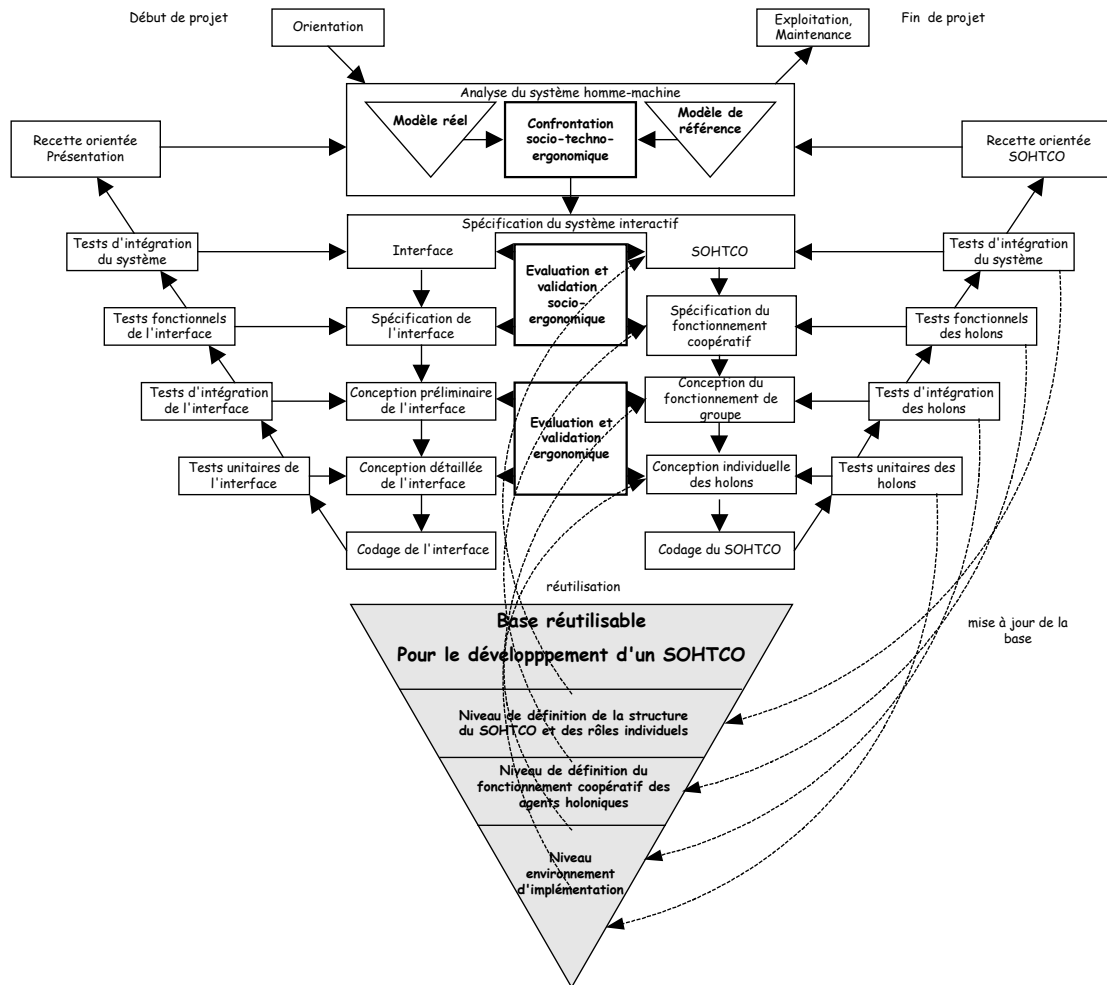


Figure III.24. Modèle ∇ et principe de réutilisabilité pour le développement de SOHTCO

Conclusion

Ce chapitre a montré que l'intégration de méthodes permet de concevoir de nouvelles méthodes, plus adaptées aux besoins d'un domaine d'application spécifique en un temps relativement court. Bien sûr, cette combinaison de méthodes doit se faire en fonction de besoins rigoureusement définis.

La méthode conçue, MAMOSACO, utilise les trois axes classiques de modélisation : un modèle de données, un modèle de flux de données et un modèle de traitement. L'originalité de cette méthode consiste à représenter les notions de responsabilité ainsi que la coopération entre les acteurs de l'organisation humaine étudiée. MAMOSACO utilise des formalismes clairs, c'est-à-dire accessibles non seulement aux modélisateurs, mais surtout aux acteurs de l'organisation humaine afin de les impliquer dans le cycle de développement du SOHTCO.

La nécessité de prendre en compte la dynamique des organisations humaines a été montrée. Un réseau de Petri paramétré a été défini pour modéliser cette dynamique et surtout les principes d'interruptions d'activités et de changement d'activité.

C'est toujours dans une volonté d'implication des utilisateurs que la conception d'un simulateur est recommandée. Ce simulateur permet en plus d'extraire les règles de fonctionnement de l'organisation qui sont ensuite utilisés par le système d'aide au travail coopératif.

Le SOHTCO (Système Orienté Holon pour l'aide au Travail Coopératif) possède la même structure holonique que les organisations humaines auxquelles s'appliquent la méthode MAMOSACO étudiée. Les formalismes décrits peuvent donc être réutilisés pendant la phase de spécification des procédures jouées par l'ensemble des parties du SOHTCO.

Un atelier de conception dédié à MAMOSACO faciliterait cette utilisation répétée des modèles. Le chapitre suivant décrit, à ce propos, l'atelier de modélisation et de simulation associé à la méthode MAMOSACO. Cet atelier simplifie l'utilisation de la méthode et permet d'obtenir les règles de fonctionnement des organisations administratives complexes étudiées.

Bibliographie du chapitre III

- [Abed 94] Abed M., Angue J.C., A new method for conception, realisation and evaluation of man-machine interfaces. *Proceedings IEEE Systems, Man and Cybernetics conference*, (2-5 Octobre 1994 ; San Antonio, USA).
- [Adam 97] Adam E., Vergison E., Kolski C., Mandiau R., Holonic User Driven Methodologies and Tools for Simulating Human Organizations, *Proceedings of the European Simulation Symposium*, (19-23 Octobre 1997 : Passau, Allemagne), ESS'97, Ed. By Han W. & Lehmann A., pp. 57-61, 1997.
- [Adam 98a] Adam E., Mandiau R., Vergison E., Parameterized Petri nets for modelling and simulating human organisations in a workflow context. *Proceedings of the Workflow Management Workshop (WfM) in the 19th International Conference on Application and Theory of Petri Net*, (juin 1998 : Lisbonne, Portugal), ICAPTN'98.
- [Adam 98b] Adam E., Mandiau R., Vergison E., Méthode adaptable basée sur la modélisation de processus pour l'analyse et l'optimisation de systèmes coopératifs dans l'Entreprise. *Actes du Colloque International Ergonomie et Informatique Avancée*, (Novembre 1998 : Biarritz), Ergo'IA 98, Biarritz : ESTIA/ILS, 1998.
- [Adam 99a] Adam E., Specifications of intelligent human-machine interfaces for helping cooperation in human-organizations. *Ergonomics and user interfaces, volume 1*. Ed. by Bullinger H.J. & Ziegler J. London: Lawrence Erlbaum Associates. pp. 311-315. 1999.
- [Adam 99b] Adam E., Mandiau R., Kolski C., Approche holonique de modélisation d'une organisation orientée workflow : SOHTCO. *Ingénierie des Systèmes multi-agents*. Ed. par Gleizes M.P. et Marcenac P., Paris: HERMES Science publications, pp. 121-134, 1999.
- [Agimont 96] Agimont G., Le Strugeon E., Mandiau R., Libert G., Parametrized petri nets for organizational simulation and systems design. *Proceedings of the Second International Conference on the Design of Cooperative Systems*, (12-14 June 1996 ; Juan-les-Pins), COOP'96, Ed. by COOP Group, INRIA Press.
- [Bailly 87] Bailly C., Challine J.F., Gloess P.Y., Ferri H.C., Marchesin B., *Les langages orientés objet: Concepts, Languages et Applications*. Cepadues editions, 1987.
- [Bastide 00] Bastide R., Spécification comportementale par réseaux de Petri : application aux systèmes distribués à objets et aux systèmes interactifs. Habilitation à Diriger des Recherches, Université de Toulouse I, 18 janvier 2000.
- [Boissier 94] Boissier O., Demazeau Y.,
An Architecture for Social and Individual Control and its Application to Computer Vision. *Proceedings of the Modelling Autonomous Agents in a Multi-Agent World conference*, (2-4 Août 2-4 1994 ; Odense, Danemark), MAAMAW.
- [Booch 94] Booch G., *Conception orientée objets et applications. 2ème édition*, USA : Addison-Wesley, 1994.

- [Borusan 93]** Borusan A., Colored Petri net based modeling of FMS. *Proceedings of IEEE Internationale Conference on Systems, Man, and Cybernetics, Vol 1*, (17-20 Octobre 1993 : Le Touquet), SMC'93, pp. 54-59.
- [Collinot 98]** Collinot A., Drogoul A., Approche orientée agent pour la conception d'organisations: application à la robotique collective. *Revue d'intelligence artificielle*, 12, 1, 1998, pp. 125-147.
- [Diaper 89]** Diaper D., *Task analysis for human-computer interaction*. Chichester, Angleterre : Ellis Horwood Limited, 1989.
- [Dumas 90]** Dumas P., Charbonnel G., *La méthode OSSAD, pour maîtriser les technologies de l'information. Tome 1 : principes*. Paris : Les éditions d'organisation, 1990.
- [Ferber 95]** Ferber J., Les systèmes multi-agents, vers une intelligence collective. IIA, Paris : InterEditions, 1995
- [Gracanion 94]** Gracanion D., Srinivasan P., Valavanis K.P., *Parameterized Petri nets and their application to planning and coordination in intelligent systems*. IEEE Transactions on Systems, Man and Cybernetics, 24, pp. 1483-1497, 1994.
- [IGL 89]** I.G.L. Technology, *SADT, un langage pour communiquer*. Paris : Eyrolles, 1989.
- [Jambon 96]** Jambon F., Erreurs et interruptions du point de vue de l'ingénierie de l'interaction homme-machine, Thèse de doctorat, Université de Grenoble, France, Décembre 1996.
- [Kolski 97]** Kolski C., *Interfaces homme-Machine, application aux systèmes industriels complexes (2^{ème} édition revue et étendue)*. Paris : Editions Hermes, 1997.
- [Kolski 98]** Kolski C., A "call for answers" around the proposition of an HCI-enriched model. *ACM SIGSOFT Software Engineering Notes*, 23, 3, 1998, pp. 93-96.
- [Montmollin 86]** De Montmollin M., *L'intelligence de la tâche, éléments d'ergonomie cognitive*. Berne : Peter Lang.
- [Palanque 95]** Palanque P., Bastide R., Spécifications formelles pour l'ingénierie des interfaces homme-machine. *Technique et Science Informatiques*, 14, 4, 1995, pp. 473-500.
- [Rumbaugh 91]** Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W., *Object-oriented modeling and design*. Prentice Hall, 1991.
- [Rumbaugh 98]** Rumbaugh J., Jacobson I., Booch G., *Unified Modeling Reference Manual*. Addison Wesley, 1998.
- [Tardieu 91]** Tardieu H., Rochfeld O., Colleti R., *La méthode Merise, principes et outils (tome 1), 2ème édition*. Paris : Editions d'Organisation, 1991.
- [van der Aalst 98]** van der Aalst W.M.P., The application of Petri nets to workflow management. *Journal of Circuits, Systems and Computers*, 8, 1, 1998, pp. 21-66.

[Vergison 98] Vergison E., Adam E., Problèmes de supervision dans les systèmes industriels complexes. *Actes des Journées Automatique et Homme du Club EEA*. (Septembre 1998 ; Valenciennes).

[Voorhoeve 96] Voorhoeve M., van der Aalst W.M.P., Conservative Adaption of Workflow. *Proceedings of the International Conference on Practical Aspects of Knowledge Management, Workshop on Adaptive Workflow*, (October 1996 : Basel, Switzerland), *PAKM'96*, Ed. by Wolf M. & Reimer U., pp. 1-15, 1996.

[Wielinga 93] Wielinga B., Van de Velde W., Schreiber G., Akkermans H., Towards an unification of knowledge modelling approaches. *Second generation expert systems*, Ed. by David J.M., Krivine J.P., Simmons R., pp. 299-335, Londres : Springer-Verlag, 1993.

[Wilson 90] *Evaluation of human works : a practical ergonomics methodology*. Ed. by Wilson J., Corlett N., London, UK : Taylor and Francis, 1990.

Chapitre IV -
Conception d'un atelier de génie logiciel supportant
la méthode MAMOSACO

INTRODUCTION	138
IV.1 PRESENTATION GLOBALE DE L'ATELIER MAMOSACO	139
IV.2 FONCTIONNALITES DE MODELISATION POUR MAMOSACO	141
IV.3 FONCTIONNALITES DE SIMULATION D'ORGANISATION ADMINISTRATIVE COMPLEXE	153
CONCLUSION	160
BIBLIOGRAPHIE DU CHAPITRE IV	161

Introduction

Le chapitre précédent a décrit les principes de MAMOSACO, en insistant sur sa composition en quatre modèles (données, activités, traitements et dynamique). Ces modèles ont été choisis pour leur lisibilité et leur compréhensibilité de la part des utilisateurs (non seulement les modélisateurs mais également les lecteurs de ces modèles), aspect sur lequel nous reviendrons d'ailleurs dans le chapitre suivant.

Cependant, l'utilisation régulière de MAMOSACO n'est possible qu'au travers d'un atelier. En effet, MAMOSACO propose dans un premier temps de simplifier l'organisation des procédures administratives complexes. Cette amélioration est réalisée par les acteurs de ces procédures. Le but est donc de fournir aux responsables de procédures (les responsables de département et/ou les responsables de la qualité) un atelier de génie logiciel, simple d'utilisation, leur permettant de modéliser les procédures.

Après un rappel de la notion d'atelier de génie logiciel, ce chapitre décrit premièrement l'atelier de modélisation facilitant l'utilisation de la méthode MAMOSACO qui permet la modélisation de systèmes administratifs complexes (cette méthode ayant été présentée au chapitre précédent).

Ce chapitre présente ensuite le simulateur d'organisation humaine (SIMOH) utilisé pour recueillir les règles de fonctionnement d'un département, dans notre application, celui de Propriété Industrielle au sein de SOLVAY S.A.

IV.1 Présentation globale de l'atelier MAMOSACO

L'atelier MAMOSACO a pour objectif de faciliter l'utilisation de la Méthode Adaptable de MODélisation de Systèmes Administratifs COMplexes qu'il supporte. Il ne vise donc pas la conception et le codage automatique de solution comme le proposent certains AGL¹ (Atelier de Génie Logiciel) surtout tournés vers les méthodes objets (par exemple, l'atelier Rational Rose 98² permet l'implémentation en langages Java, C++, Visual Basic, Ada, SmallTalk d'objets décrits par les modèles issus des méthodes Booch, OMT et UML).

IV.1.1 Rappel sur les ateliers de génie logiciel

Un AGL (Atelier de Génie Logiciel) est un logiciel facilitant la réalisation de logiciels. Il peut être considéré comme un système de conception assisté par ordinateur. Un AGL doit intégrer, de façon cohérente, les éléments nécessaires aux différentes phases de conception et/ou de développement d'un logiciel. En effet, les AGL se distinguent généralement en deux catégories : les ateliers de conception et les ateliers de développement [Constantinidis 98].

Les ateliers de conception (upper-case en anglais) se composent d'outils (tels que des éditeurs de diagramme, des générateurs de dictionnaires de données, des générateurs de code entre autres) facilitant les phases d'analyse, de modélisation, de spécification et de conception préliminaire pour la réalisation d'un logiciel. Ces composants reposent généralement sur une librairie qui consiste en une base de données permettant le passage des modèles à la spécification puis au codage. Certains ateliers proposent plusieurs librairies correspondant à différents modèles et offrent la possibilité de passer automatiquement d'un modèle à l'autre. Par exemple, WithClass³ de Microgold permet, à partir de diagrammes réalisés à partir des modèles de UML, OMT, Booch, Coad-Yourdon et Shlaer-Mellor, de générer des classes en Java, Delphi, C++ et Visual Basic et permet également de passer d'une représentation de type UML à une représentation de type OMT. La figure IV.1 représente la structure générale d'un atelier de conception.

¹ Les AGL sont traduits en anglais par CASE Tools pour Computer Aided Software Engineering (différentes combinaisons sont possibles, ainsi la lettre A est également utilisée pour Assisted ou Automated, et la lettre S pour System).

² Le site relatif au produit Rational Rose se trouve à l'URL : <http://www.rational.com/products/rose/index.jtmpl>.

³ Des informations sur withclass sont disponibles à l'URL : <http://www.microgold.com/>

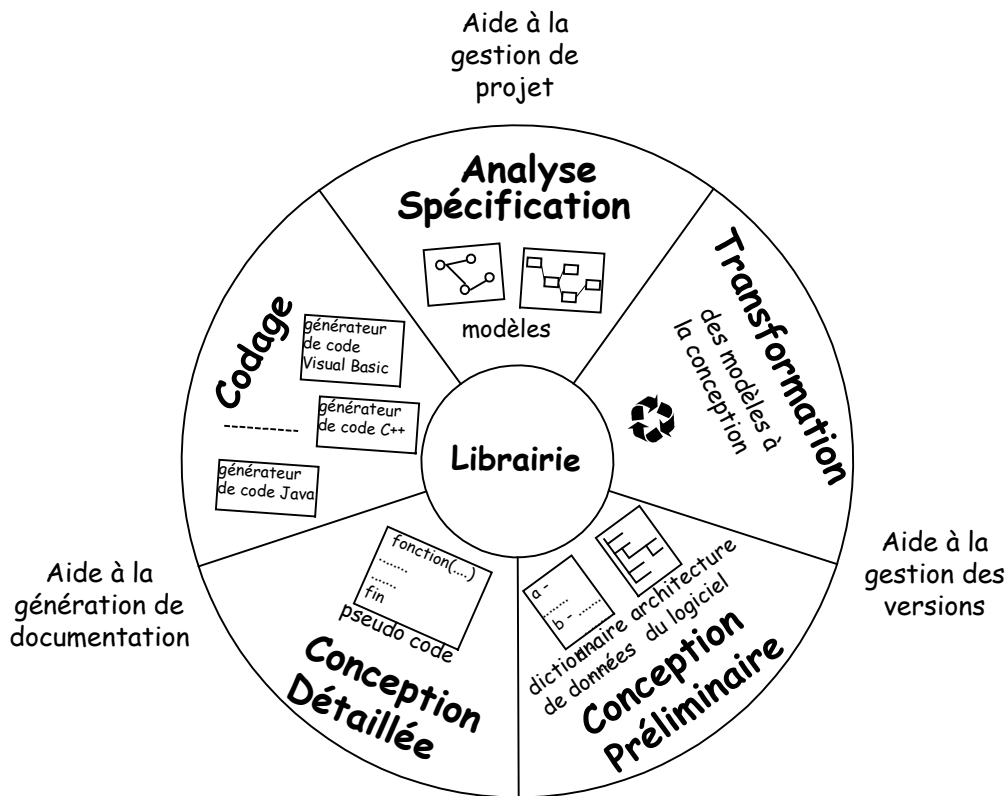


Figure IV.1. Les composants principaux d'un atelier de conception, inspiré de [Jaulent 90]

Les ateliers de développement (lower-case en anglais) sont plus tournés vers l'implémentation et le test des composants du logiciel visé. Ils intègrent généralement un éditeur de code, une aide à la conception d'interfaces homme-machine, un compilateur et un debugger. Il existe actuellement de nombreux ateliers de développement utilisés en industrie tels que Visual C++ et Visual Basic de Microsoft⁴, Borland C++ et Delphi de Borland. La Figure IV.2 représente les composants principaux d'un atelier de développement.

⁴ Visual Basic et Visual C++ sont des ateliers de développement de Microsoft® très orientés vers la réalisation de systèmes interactifs.

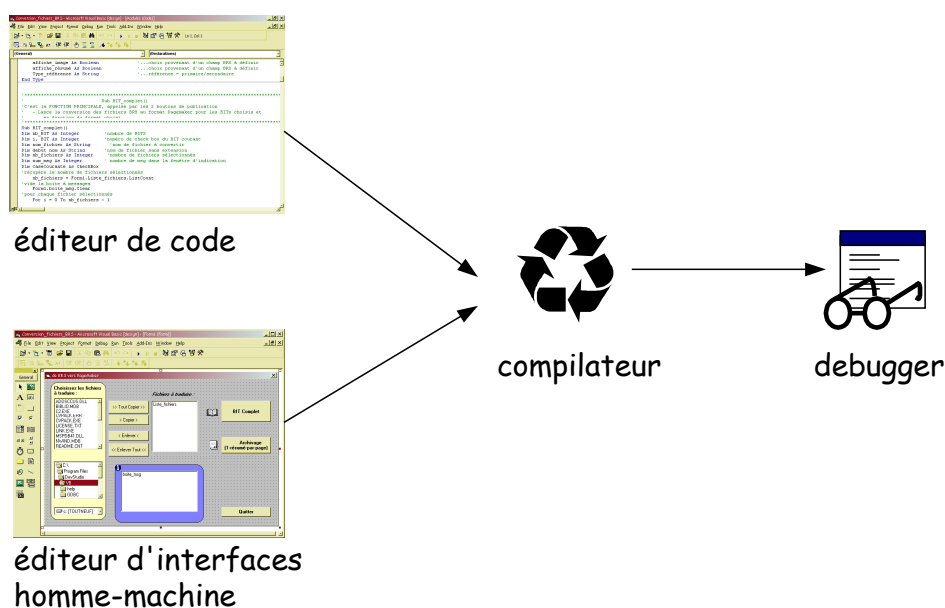


Figure IV.2. Architecture générale d'un atelier de développement

IV.1.2 Architecture de l'atelier MAMOSACO

L'atelier MAMOSACO se situe au niveau des ateliers de conception. Il permet en effet, de modéliser le système réel, de le simuler afin d'obtenir ses règles de fonctionnement qui seront utilisées lors de la phase de développement du SOHTCO (Système Orienté Holon pour l'aide au Travail COopératif). L'atelier intervient également lors de la phase de spécification des procédures coopératives du SOHTCO. Les phases de conception préliminaire, de conception détaillée et de codage ne sont pas prises en compte dans cet atelier.

L'atelier MAMOSACO a été conçu pour pouvoir être utilisé par les acteurs de l'organisation ayant des responsabilités dans la gestion de processus (par exemple les décideurs ou les responsables qualité). En général, ces personnes possèdent une bonne maîtrise du concept de processus mais ne sont pas expertes en modélisation. Les deux composantes de l'atelier, liées d'une part à la modélisation, d'autre part à la simulation [Adam 97], doivent donc être très intuitives. C'est pourquoi un ensemble de fonctionnalités, associées à ces deux étapes, a été développé à l'aide des ateliers de développement Visual Basic et Visual C++.

IV.2 Fonctionnalités de modélisation pour MAMOSACO

La composante associée à la modélisation de l'atelier MAMOSACO se base sur le logiciel VISIO© (VISIO Corporation), qui est un logiciel graphique sur PC, conçu pour schématiser les systèmes d'information et les processus d'entreprise. Plus de 150 boîtes à

outils dédiés sont à la disposition du développeur afin de lui faciliter, entre autres, la modélisation de réseaux, de circuits électroniques ou d'organisations.

En ce qui concerne la conception d'un atelier de modélisation, VISIO n'offre pas simplement une aide à la mise à plat graphique des systèmes ou des processus, il permet aussi d'automatiser certaines actions, voire de générer des fichiers (comme par exemple un dictionnaire ou du code informatique) dans le cadre de l'utilisation de modèles particuliers (UML par exemple). Ces automatismes sont possibles principalement par le langage Visual Basic.

De plus, VISIO possède une interface utilisateur qui rend accessible directement et clairement, toutes les options utiles à la conception de diagrammes. En effet, le "vocabulaire" de VISIO est limité. L'objet principal est le *document*, entité logique constituée de pages. La *page* est composée de diagrammes. Le *diagramme (shape)* peut être lui-même décliné en sous-diagrammes. Des *liens* (connectors) sont utilisés pour relier les diagrammes. Les diagrammes possédant des formes ou des comportements particuliers peuvent être regroupés dans des *boîtes à outils* (par exemple, il existe une boîte à outil contenant les diagrammes nécessaires à la modélisation par UML ou par OMT). Les diagrammes inclus dans les boîtes à outils sont alors considérés comme des classes. L'instantiation de ces classes s'effectue en glissant le diagramme de la boîte à outil sur une page. La sémantique de VISIO exploitée par l'atelier logiciel se réduit aux deux actions élémentaires suivantes : placer, glisser, un diagramme sur une page ; lier deux diagrammes.

L'atelier de modélisation se compose d'une boîte à outils, de six pages de tailles différentes dont quatre correspondent aux modèles utilisés par MAMOSACO. La figure IV.3 représente l'enchaînement de ces pages. La première page est une page de présentation de la méthode et permet de définir le titre de la procédure à modéliser. La seconde page correspond à une fiche descriptive décrivant brièvement les acteurs et les sous-ensembles de la procédure. Cette étape permet de définir une matrice activités/rôles inspirée de Grudin [Grudin 94]. Cette matrice est détaillée dans la partie utilisation de l'atelier. Les autres pages correspondent aux modèles utilisés par MAMOSACO. Les trois premiers sont des points de vue statiques sur l'organisation : les objets manipulés (document, informations, etc.) ; les flux d'informations ; la manière dont les informations sont traitées. Le quatrième mode de représentation, plus complexe et à base de règles (règles globales, locales et personnelles de manipulation des

objets), prend en compte la dynamique de fonctionnement du système. Les fonctionnalités associées à ces pages sont présentées dans cette partie.

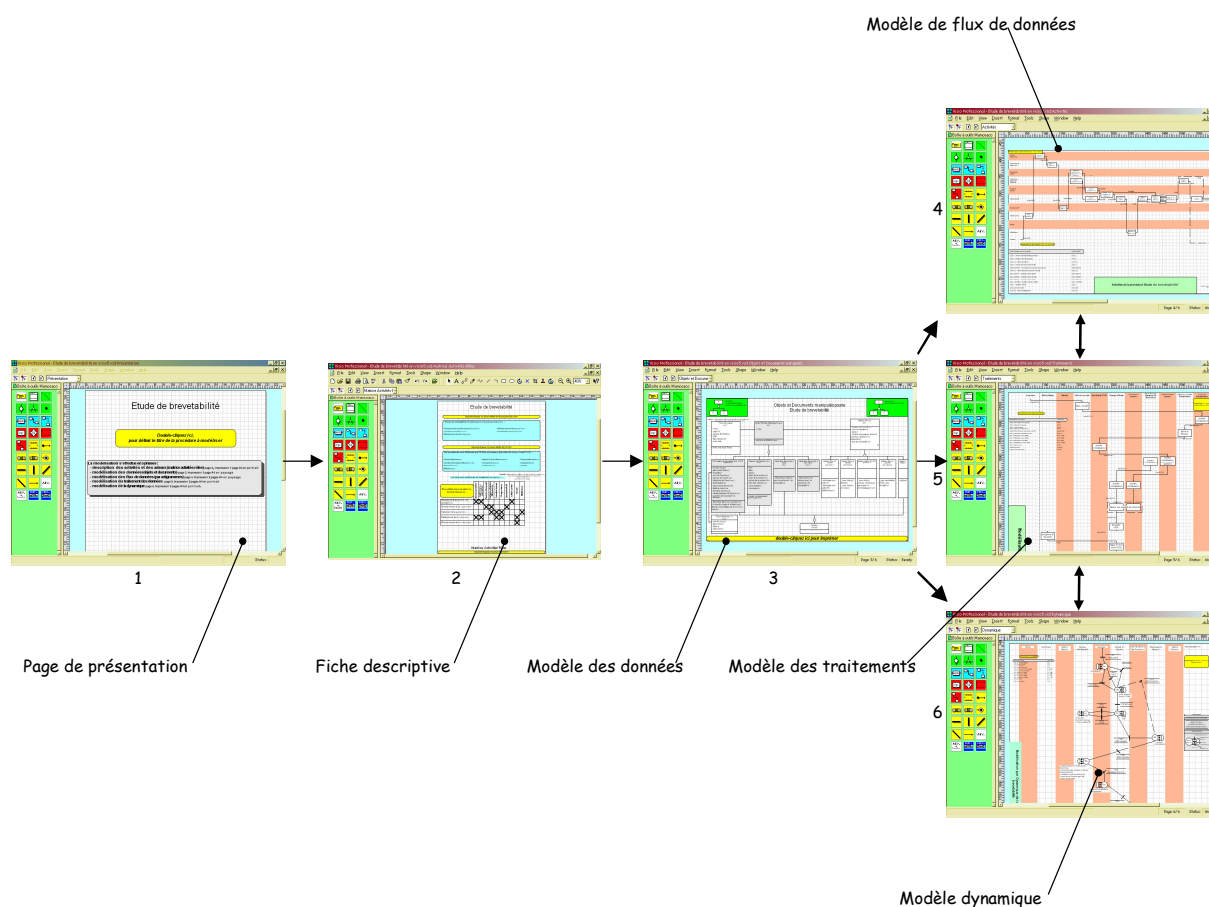


Figure IV.3. Les six pages de l'atelier MAMOSACO

IV.2.1 Conception de l'atelier

Le logiciel VISIO© permet d'ajouter des comportements aux diagrammes, en fonction de clics ou de double-clics. Il permet également de modifier les caractéristiques telles que la mobilité (pour fixer un objet sur la page), la protection contre l'effacement, la géométrie du diagramme par des règles. Des règles peuvent être associées aux caractéristiques des objets. Ainsi, il est possible de définir un triangle isocèle par des règles de façon à ce que les actions de l'utilisateur (tels que l'étirement) n'affectent pas ses propriétés (égalité d'au moins deux côtés).

Les diagrammes de VISIO peuvent être considérés comme des objets. Il est en effet possible d'associer à ces diagrammes des attributs et des méthodes particulières (par exemple il est possible d'ajouter un champ texte expliquant la nature de l'objet et une fonction affichant ce texte dans une boîte de messages).

Sous VISIO, toutes ces caractéristiques (propres à l'objet et ainsi que celles définies par l'utilisateur) sont définies sous forme de tableau. La figure IV.4 présente un tableau associé à une instance de la classe de diagramme "Objet_Attributs_Fonction". Cette instance, qui porte dans la figure le numéro 76, se trouve dans la page Objets et Documents Manipulés du document Drawing1. Elle représente un Document de Synthèse. Lorsqu'un objet de ce type reçoit un événement de type double-clic, une macro (un add-on) est activée.

Shape Transform						
User-defined Cells		Value	Prompt			
User.Row_1		"Document de 'synthèse'"	"Nom de l'objet"			
User.Row_2		"D.S."	"Abréviation"			
Connection Points		X	Y	A	B	
Controls		X	Y	X Dynamics	Y Dynamics	X Behav
Scratch		X	Y	A	B	C
Protection						
Events						
TheData	No Formula	EventDbClick	RUNADDON("EDITER...			
TheText	No Formula	EventXFMod	No Formula			
		EventDrop	No Formula			
Miscellaneous						
Glue Info						

Figure IV.4. Exemple de tableau de définition de diagramme sous VISIO

Ce type de programmation par tableau est très contraignant et fait preuve d'une ergonomie d'utilisation relativement faible. Heureusement, les versions récentes de VISIO proposent une librairie permettant d'accéder à toutes les parties des documents à partir de programmes rédigés en Visual Basic. Cette capacité est très importante car il est alors possible de concevoir une couche logicielle facilitant la réalisation des modèles et bénéficiant d'interfaces graphiques. Par exemple, la figure IV.5 correspond à la «couche» Visual Basic facilitant la définition de la donnée «Document de Synthèse». Elle guide l'utilisateur pour la définition de données, lui évitant ainsi d'éditer manuellement les différentes zones du diagramme correspondant (figure IV.6).

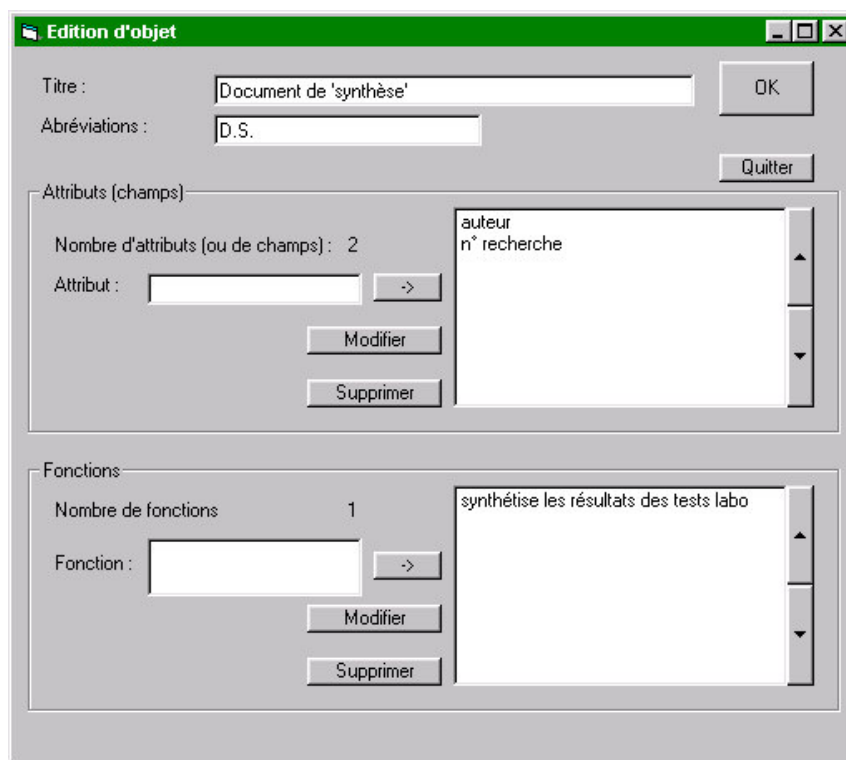


Figure IV.5. Couche logicielle facilitant l'édition d'une donnée dans VISIO

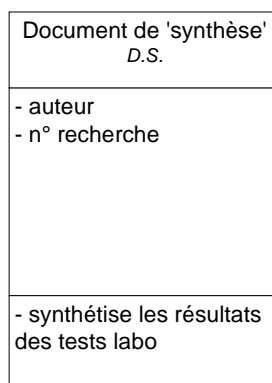


Figure IV.6. Exemple de modèle objet d'un document

Le lien entre le diagramme du modèle objet et la couche Visual Basic est possible grâce aux attributs personnalisés que possède le diagramme. Le support Visual Basic, en fonction des actions de l'utilisateur, modifie ces attributs personnalisés que le diagramme peut alors afficher.

D'autres couches et diagrammes ont ainsi été programmés en Visual Basic facilitant la définition des acteurs, le découpage de procédures en sous-procédures, la préparation des modèles d'activités, de traitements, du modèle dynamique et la génération des légendes. D'autres supports tels que la saisie du titre de la procédure, la mise en page et l'impression des modèles, le tracé d'une grille ont également été mis en place. Ces outils ont été définis dans le but d'un emploi par des utilisateurs occasionnels de l'atelier et de la méthode

MAMOSACO ainsi que de VISIO. Les critères d'ergonomie des logiciels tels que l'accessibilité ou la cohérence dans les formats de présentation ont guidé la création des aides à la modélisation. En somme, quatre fenêtres utilisateur et neuf macros ont été conçues (facilitant la définition du titre de la procédure modélisée, l'édition de rôles, l'édition des sous procédures, la création de la matrice activité-rôle, l'édition des objets, la rédaction de la légende, la modélisation des activités, des traitements et de la dynamique).

IV.2.2 Utilisation des fonctionnalités de modélisation

L'utilisateur (par exemple le chef de service ou le responsable qualité) doit ouvrir sous Visio le modèle MAMOSACO. Le modèle s'ouvre alors sur la page d'accueil et propose la boîte à outil MAMOSACO présentant les divers diagrammes nécessaires à la modélisation (fig. IV.7). Cette boîte à outils est composée de quatre ensembles d'outils identifiables par des couleurs relatives aux pages dans lesquels ils doivent être utilisés.

Sur les six pages que contient le modèle se trouvent des zones particulières de fond jaune. Ce sont les zones dites cliquables, qui font appel aux supports Visual Basic et qui n'apparaissent pas à l'impression. Ces zones se démarquent volontairement du modèle, afin de répondre au critère d'accessibilité des interfaces telle que définie dans [Bastien 93].

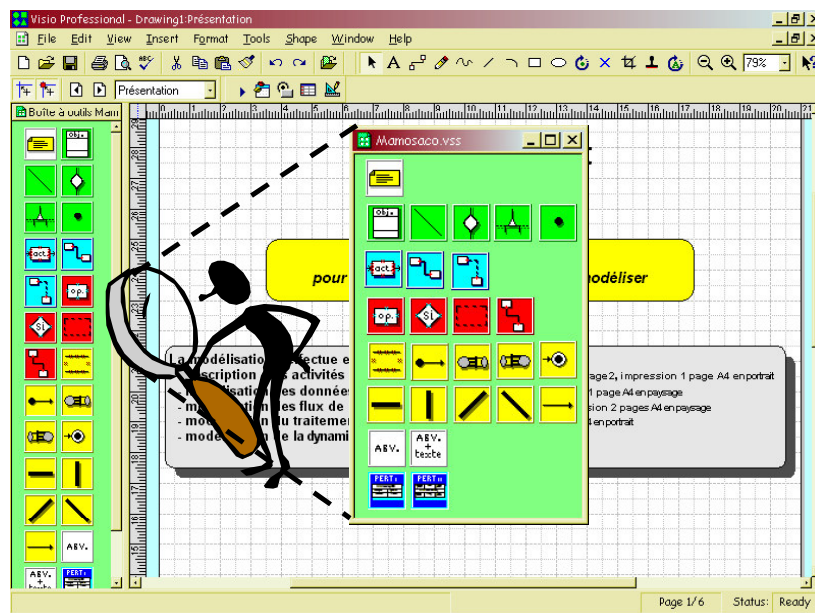


Figure IV.7. Boîte à outils de l'atelier de MAMOSACO

IV.2.2.1 Fonctionnalités associées à la matrice Activité-Rôle

La deuxième page permet donc de créer la fiche descriptive de la procédure sous la forme d'une matrice activités/rôles [Grudin 94]. Le but de cette page est de définir les rôles

intervenant dans la procédure et de les classer selon leurs niveaux de responsabilité. Cette page a également pour but de découper la procédure en étapes et d'indiquer à quelles étapes prennent part les rôles. Cette étape est obligatoire : outre le fait qu'elle entraîne une première réflexion de la part de l'utilisateur sur l'ensemble de la procédure, elle permet à Visio de mémoriser le nom des acteurs. Trois macros sont associées à cette page ; elles guident l'utilisateur, par l'intermédiaire de formulaires, dans la définition des rôles, des sous-procédures et lors du tracé de la grille. La figure IV.8 est un exemple de grille activités-rôles correspondant à une procédure de dépôt de brevet prioritaire au sein d'un département de propriété industrielle.

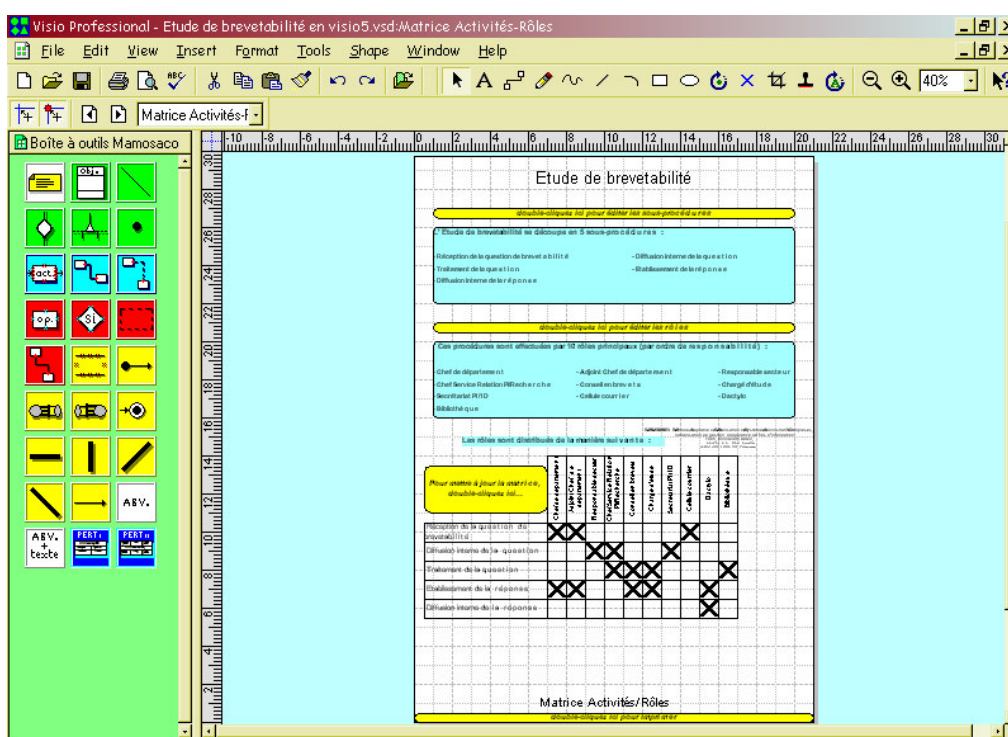


Figure IV.8. Exemple de matrice activités/rôles pour une procédure administrative complexe (cas d'une procédure de demande de brevet).

IV.2.2.2 Fonctionnalités associées au modèle des données

La troisième page est relative au modèle des données inspiré de UML. Elle permet de décrire les documents et leurs relations (d'appartenance, de type, de correspondance). Un objet est défini par son nom, une abréviation, des attributs (ou champs) et des fonctions (pour un document cela correspond à une brève description de son utilisation). Chaque diagramme représentant un objet est associé à une macro facilitant l'édition (figure IV.5).

Il ne peut exister, lors de la modélisation, d'objet, de document, non lié à d'autres objets. En effet, chaque document doit être lié à un autre document par au moins une association,

c'est le cas de documents ou de données possédant un ou des attributs communs (par exemple une référence à une invention ou un nom d'auteur).

Les relations d'agrégation sont symbolisées par un losange. Elles sont utilisées pour signaler le fait qu'un document contienne d'autres documents.

Les relations d'héritage sont symbolisées par des triangles. Elles sont utilisées lorsqu'un ensemble de documents possède les mêmes caractéristiques. Par exemple, des rapports de synthèse de départements possédant différentes activités au sein d'une même entreprise peuvent posséder la même structure générale et ainsi que des points propres aux départements auxquels ils se rattachent.

La figure IV.9 représente, un dossier de candidature contenant un curriculum vitae, un ensemble d'annexes et un texte de demande. Il existe plusieurs versions de ce texte, seule la version définitive est placée dans le dossier.

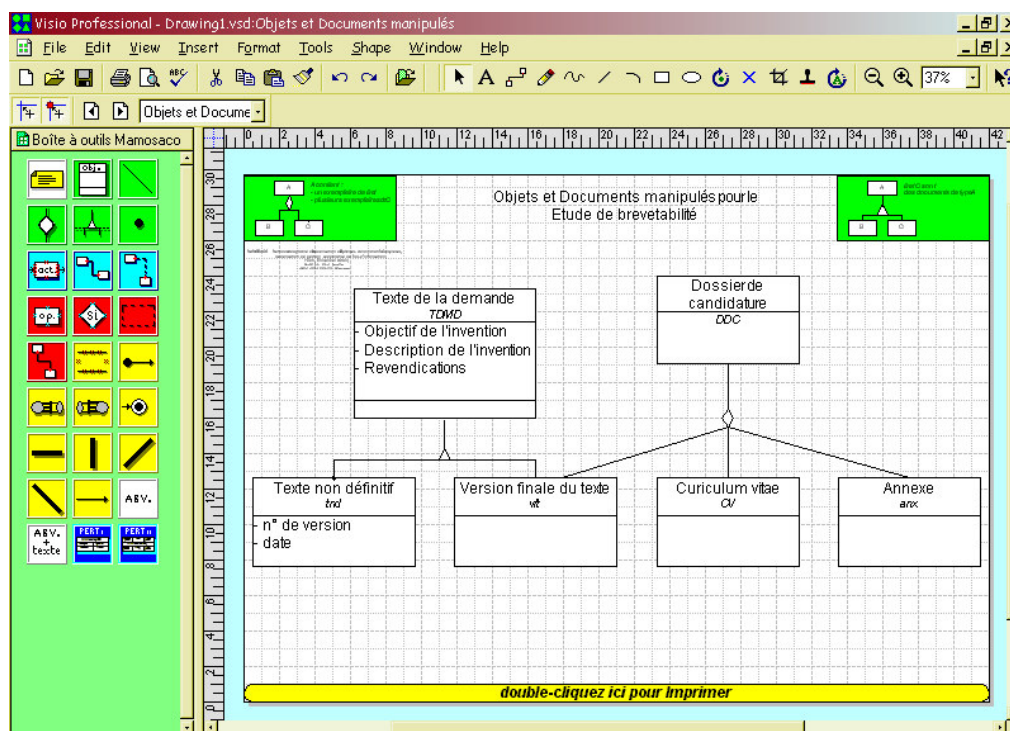


Figure IV.9. Exemple de représentation des données par le symbolisme de UML

Les outils associés à cette étape sont au nombre de quatre (la Figure IV.10 est un extrait de la boîte à outils présentée en fig. IV.7), correspondant respectivement au diagramme représentant un objet, aux liens de relations, d'agrégation et de hiérarchie, et au symbole représentant le lien à un ensemble de données.

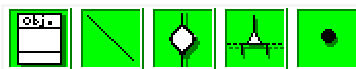


Figure IV.10. Outils relatifs au modèle des données

IV.2.2.3 Fonctionnalités associées au modèle des activités

La quatrième page concerne le modèle des activités. Son but est de donner une vue d'ensemble de la procédure, sans entrer dans le détail. Il représente tous les flux de données possibles, quelles que soient leur importance et les conditions dans lesquelles ils apparaissent.

Deux macros sont également associées à cette page. L'une crée automatiquement, à partir des données entrées dans la deuxième page relative à la matrice activité-rôle, les lignes correspondant aux acteurs, classés par ordre de responsabilité, et les zones correspondant aux sous-procédures. La seconde macro génère la légende, à partir du modèle des données.

Ces macros ont pour objectif de faciliter la modélisation. De même, le nombre d'outils nécessaires est limité à trois : une boîte d'actigramme, une flèche orientée de gauche à droite et une flèche pointillée orientée de droite à gauche pour les retours de données (fig IV.11).

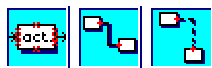


Figure IV.11. Outils nécessaires à la modélisation des activités
(extraits de la boîte à outils présentée en fig. IV.7)

L'utilisation des actigrammes est également simplifiée. En effet, la norme IDEF0⁵ impose que l'unité de traitement des données, ici le rôle, soit indiquée sous le diagramme. Cependant, pour ne pas surcharger le modèle, les unités de traitements, les rôles, sont indiquées aux extrémités de chaque page. Si une donnée de contrôle doit être représentée, elle est indiquée sur le dessus de la boîte, conformément à la norme.

La figure IV.12 représente un extrait d'un modèle d'activité pour une procédure d'étude de brevetabilité.

⁵ La norme IDEF0 est utilisée dans la méthode SADT afin de représenter les activités d'un système et les flux de données entre ces activités (d'autres normes sont également disponibles sur le site www.idef.com).

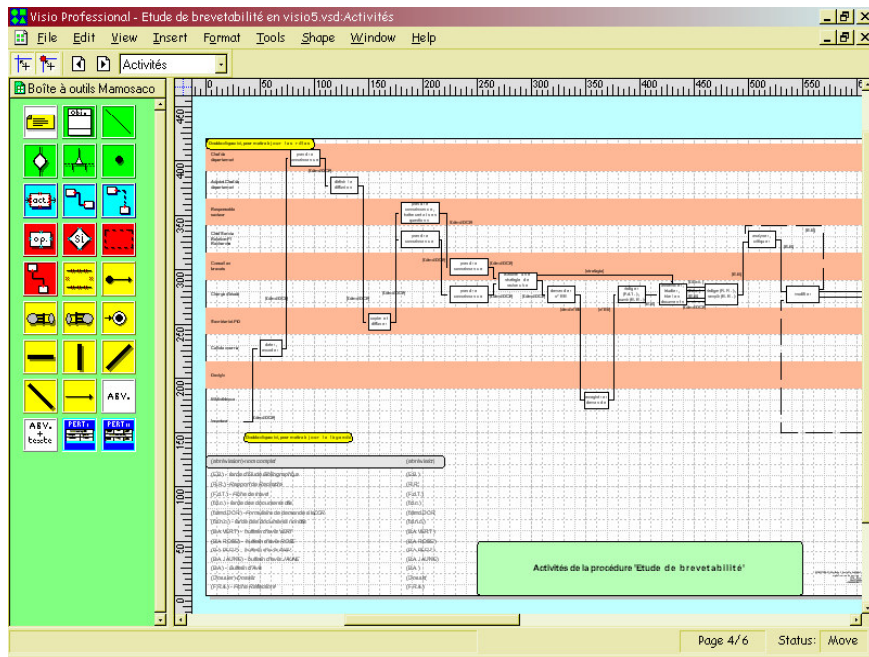


Figure IV.12. Exemple de modèle d'activité

IV.2.2.4 Fonctionnalités associées au modèle des traitements

La cinquième page permet de créer le modèle de traitements. Ce modèle apporte plus de précision que le modèle précédent, notamment sur les conditions dans lesquelles les données passent d'un acteur à un autre acteur. Il permet aussi de représenter la coopération entre acteurs et peut être utilisé en complément du modèle des activités.

Dans ce modèle, le temps s'écoule de haut en bas et les acteurs, à l'instar du modèle d'activité, sont positionnés sur des colonnes selon un niveau croissant de responsabilité allant de gauche à droite.

Comme dans le précédent modèle, deux macros facilitent la mise en page, notamment en ce qui concerne la création des colonnes, l'écriture de la légende et le découpage en sous-procédures.

Quatre outils permettent de modéliser les traitements. Il s'agit respectivement du diagramme représentant une boîte de traitement, une condition, une coopération (par un cadre en pointillé), un flux de données allant de haut en bas (fig. IV.13).



Figure IV.13. Outils disponibles pour la modélisation des traitements (extraits de la boîte à outils présentée en fig. IV.5)

La figure IV.14 représente un extrait d'un modèle de traitements pour une procédure de dépôt de brevet.

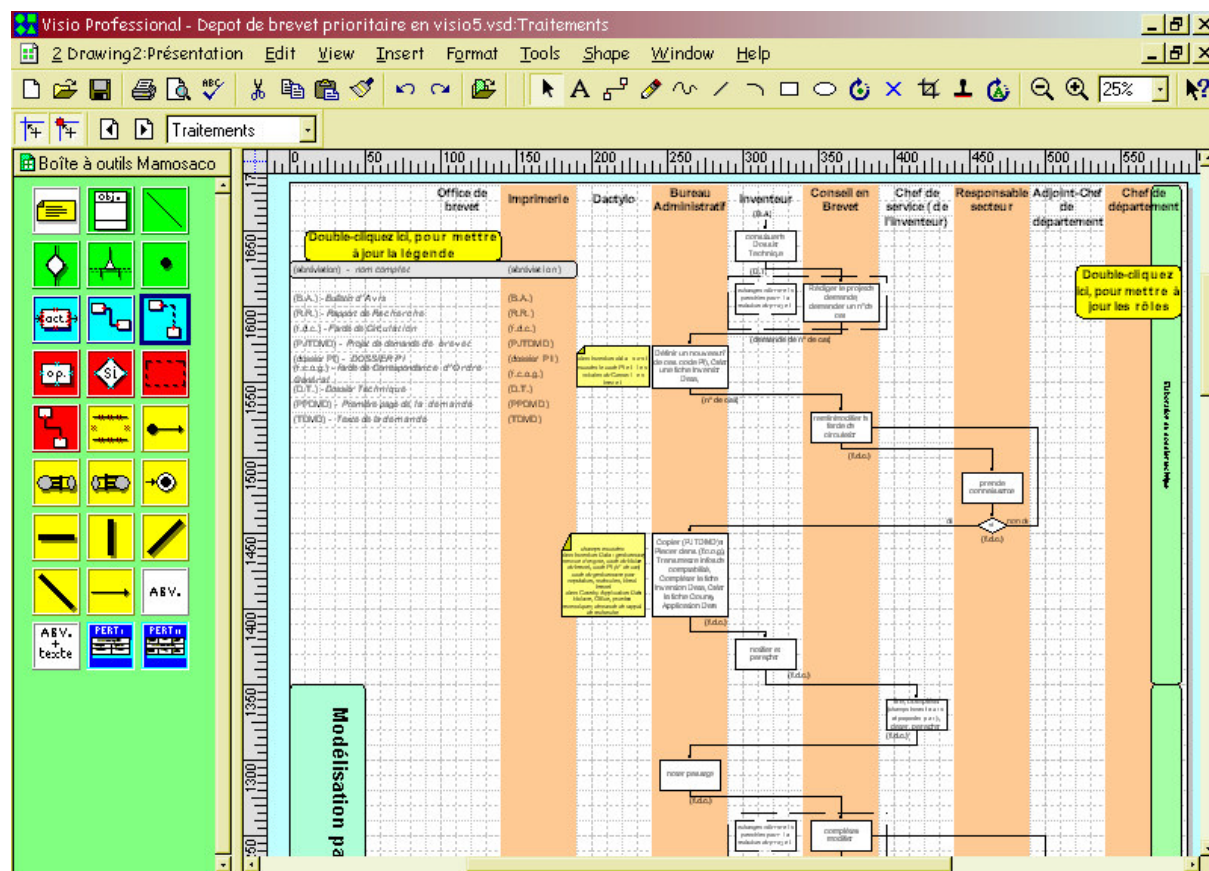


Figure IV.14. Exemple de modèle de traitement (cas d'une procédure de dépôt de brevet)

IV.2.2.5 Fonctionnalités associées au modèle dynamique

La sixième et dernière page se rapporte au modèle dynamique. En raison des concepts mis en jeu, ce modèle est celui qui, potentiellement, est le plus à même de poser des difficultés à l'utilisateur. Pour simplifier cette tâche, notamment en ce qui concerne l'écriture des règles qui sous-tendent la procédure, un simulateur a été conçu, il sera présenté au paragraphe IV-3. Deux macros sont associées à cette page permettant, comme pour les deux précédents modèles, de placer les acteurs, de décomposer la page en sous-procédures, et de générer la légende.

Ce modèle se base sur les réseaux de Petri paramétrés. Seulement, pour plus de lisibilité, le réseau est "déplié dans le temps", c'est-à-dire qu'une même place peut apparaître plusieurs fois selon que l'espace de travail associé est sollicité.

Ce modèle, plus complexe que les précédents, nécessite l'utilisation de dix outils correspondant aux places paramétrées, aux transitions, aux états initiaux et finaux, aux

éléments et aux règles utilisées. La figure IV.15 représente un extrait d'un modèle dynamique pour une procédure d'étude de brevetabilité.

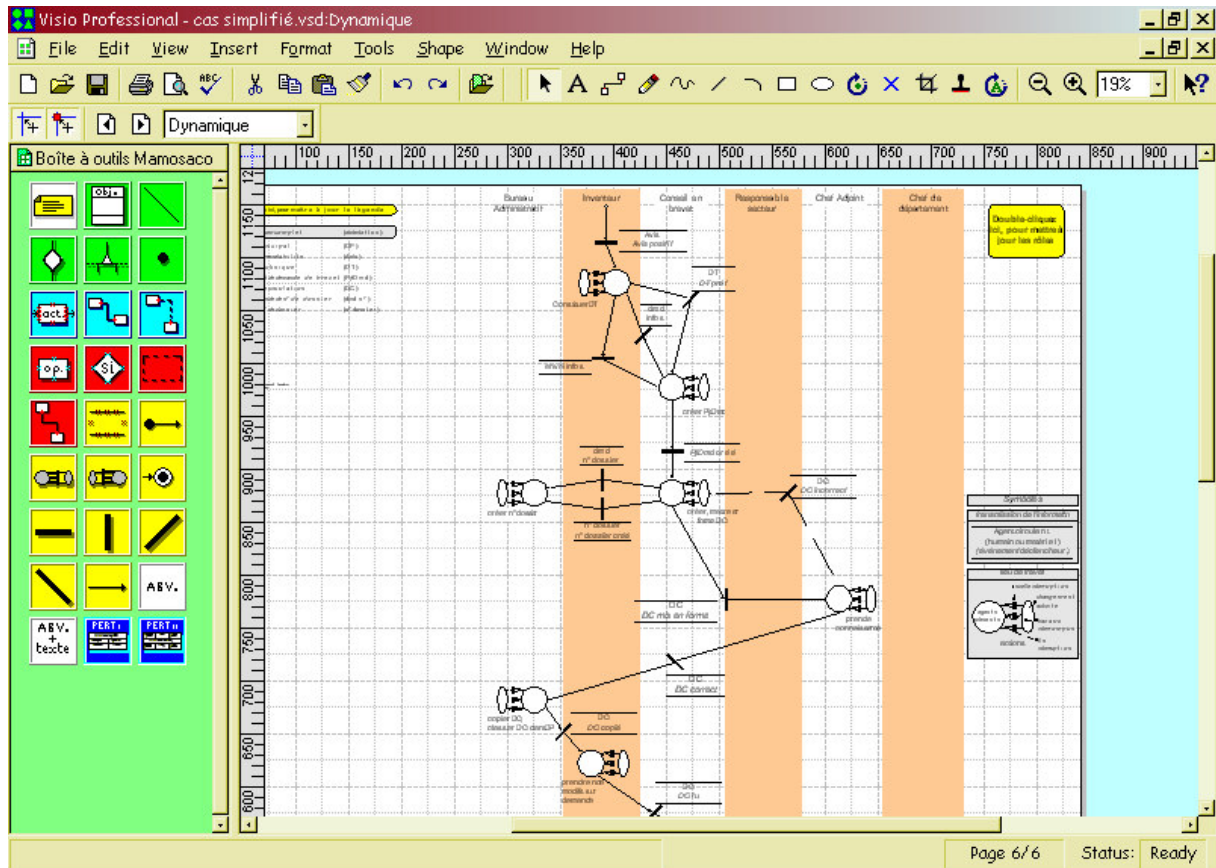


Figure IV.15. Exemple de modèle dynamique (cas d'une procédure d'étude de brevetabilité)

IV.2.3 Autre fonctionnalité disponible dans l'atelier MAMOSACO

Outre les macros spécifiques à chaque modèle, il figure, sur chacune des pages, une macro guidant l'utilisateur pour l'impression. En effet, celles-ci étant de divers formats, une mise en page spécifique est nécessaire. Celle-ci nécessite onze actions élémentaires de la part de l'utilisateur (huit clics et trois saisies) pour chaque page. Une interface a donc été développée afin de faciliter cette étape obligatoire (fig. IV.16).

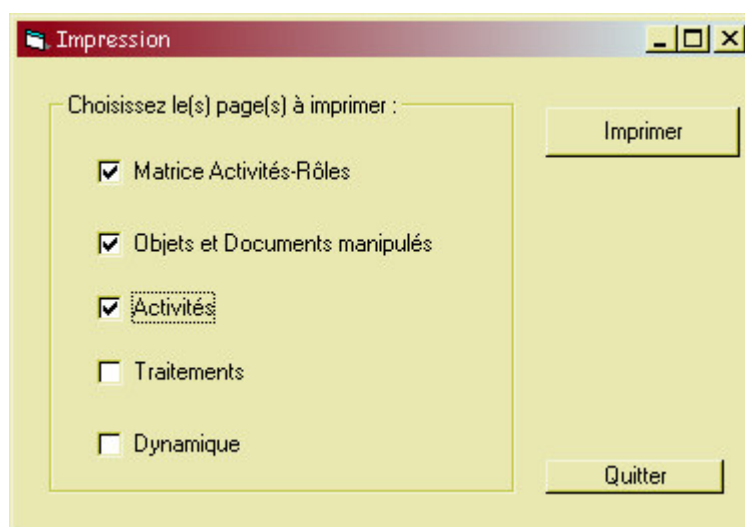


Figure IV.16. Interface utilisateur pour l'impression des modèles sous VISIO

IV.2.4 Conclusion sur les fonctionnalités de modélisation

Cet atelier permet, grâce aux fonctionnalités de VISIO © telles que les connexions dynamiques, aux macros Visual Basic © et à la boîte à outils, un emploi par des utilisateurs occasionnels en diminuant la surcharge cognitive (au sens de l'ergonomie des logiciels [Bastien 93]) inhérente aux utilisations de nouveaux logiciels.

Cependant, malgré les facilités offertes par l'atelier, la description des règles de fonctionnement par le modèle dynamique n'est pas aisée, le recours à un simulateur s'impose alors.

IV.3 Fonctionnalités de simulation d'organisation administrative complexe

Les simulateurs sont généralement utilisés soit pour comprendre le fonctionnement d'un système et tenter de prévoir son évolution, soit pour tester de nouvelles solutions, soit dans un but pédagogique, afin de faciliter l'apprentissage du fonctionnement de systèmes complexes [Lant 94].

C'est dans ces deux dernières optiques que nous nous inscrivons. Le simulateur doit en effet faciliter la prise de conscience de la part des acteurs des procédures de l'organisation du fonctionnement de ces procédures. Il doit également permettre de tester de nouvelles solutions organisationnelles à ces procédures. Néanmoins, son rôle premier reste l'assistance à la description des règles de fonctionnements des procédures nécessaires à la spécification des SOHTCO.

La figure III-17 représente l'interface de la simulation d'un extrait d'une procédure de dépôt de brevet.

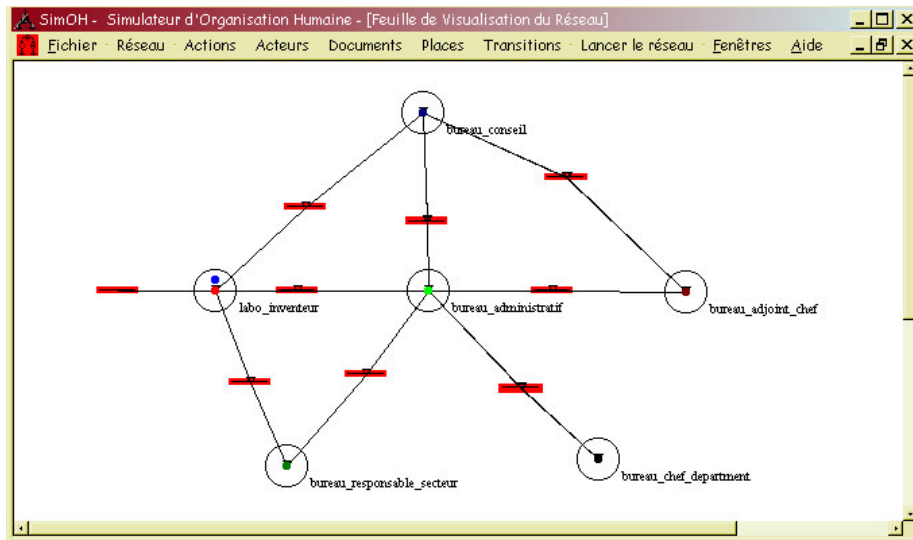


Figure IV.17. Interface SIMOH pour une procédure de dépôt de brevet

IV.3.1 Conception du simulateur SIMOH

Le simulateur SIMOH (SIMulateur d'Organisation Humaine) a été développé en Visual C++ [Gransac 97]. Il se compose de deux parties : une partie visible constituée de l'interface homme-machine et des fenêtres de saisie du réseau ; une partie interne constituée d'un moteur d'inférence balayant les règles associées aux transitions, aux places et aux acteurs.

La partie interne du simulateur est définie de la façon suivante (chaque règle, appelée activité, est décomposée en préconditions et en actions exécutées si les préconditions sont validées) :

```

Réseau = {Transitions, Places, Jetons }
Transition = {Place en entrées, Places en sorties, Activités}
Place = {Jetons, Interruptions, Activités}
Jeton = {Acteur, Document}
Acteur = {nom, état, Activités}
Document = {nom, état}
Interruption = {nom, état}
Activité = {Préconditions, Actions}

```

Cette structure correspond au modèle objet présenté à la figure IV.18.

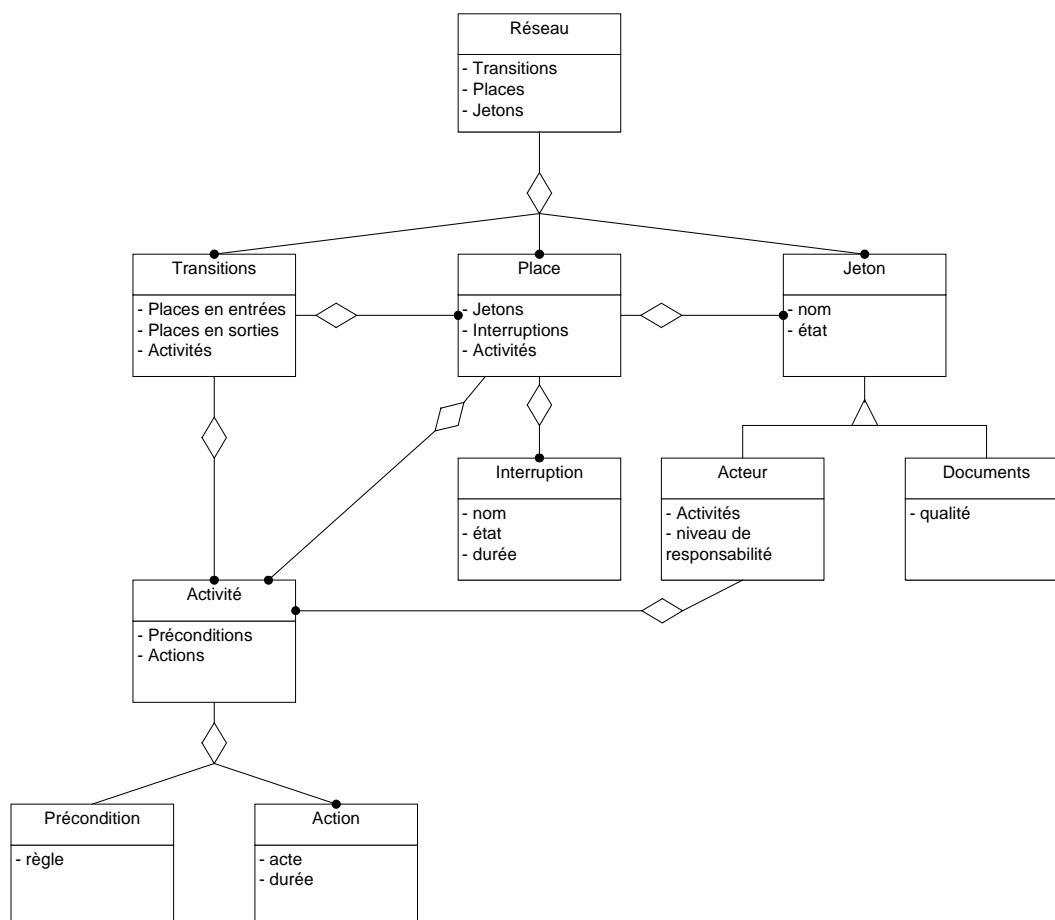


Figure IV.18. *Modèle objet UML de la structure interne du simulateur*

A chaque étape de la simulation, où le temps est défini par une suite d'événements, les activités de toutes les parties du réseau sont testées et exécutées selon le cas. SIMOH permet donc de suivre l'évolution d'un processus coopératif tout en représentant les acteurs, leurs actions et leurs déplacements possibles dans l'organisation.

Le test d'une précondition consiste à vérifier si un jeton (acteur ou document) existe dans un état donné dans une place. La réalisation de l'action consiste selon le cas à modifier l'état du jeton, l'interrompre, le "reprendre" (c'est-à-dire le faire passer de l'état "en attente" à l'état "en traitement"), le déplacer, le supprimer ou encore à créer un nouveau jeton.

Dans un objectif d'intégration des utilisateurs, il est nécessaire que ceux-ci puissent utiliser, de façon autonome, le simulateur (afin de tester leurs solutions par exemple). La saisie des règles, des activités, doit donc s'effectuer dans une forme aussi proche possible du langage naturel. Ainsi l'activité, associée au chef de département, traduisant le fait qu'il consulte un projet principal dès sa réception s'écrira : `Activité = {projet_principal}; {projet_principal devient consulté}`.

Les mots clés et la grammaire du langage associés aux règles sont relativement simples. En effet, trois mots clés ('dans', 'et' et 'ou') sont associés aux préconditions, qui se déclenchent lorsqu'un document ou un acteur se trouvent dans une place donnée. Ainsi, la précondition levée lorsque le dossier de circulation approuvé se trouve dans le bureau administratif et que le projet principal est situé dans le bureau du chef de département s'écrit : "dossier_de_circulation.approuvé dans bureau_administratif et projet_principal dans bureau_chef_département".

En ce qui concerne les actions, quatre mots clés ont été retenus : 'apparaît' utilisé pour la création de documents; 'disparaît' utilisé pour la suppression de document; 'est_copié' utilisé pour la photocopie de documents et 'devient' utilisé pour la modification de documents. Par exemple, l'action consistant à vérifier un dossier de circulation et à en faire une copie s'écrit en deux étapes : 'dossier_de_circulation.prêt devient vérifié'; 'dossier_de_circulation est_copié 1 fois'.

Ce vocabulaire est volontairement limité, afin de ne pas 'perdre', gêner l'utilisateur du simulateur. Les termes ont été retenus en fonction des premiers exercices de modélisation, des premiers modèles effectués, et en accord avec les futurs utilisateurs.

Toutes ces règles, ainsi que la définition des acteurs, des documents, des places et des transitions sont stockées dans un fichier texte. Néanmoins, pour faciliter la définition du réseau, un ensemble de fenêtres d'édition a été réalisé.

IV.3.2 Utilisation des fonctionnalités de simulation

L'utilisation du simulateur SIMOH fait suite aux étapes de modélisation. Elle nécessite de décrire premièrement les acteurs (fig. IV.19) jouant un rôle dans la procédure, leurs niveaux de responsabilité, leurs états (libres ou non) ainsi que leurs qualités (responsable de l'espace de travail par exemple). Il est également nécessaire de décrire les documents par leur nom, leur état et leur qualité (confidentiel par exemple).

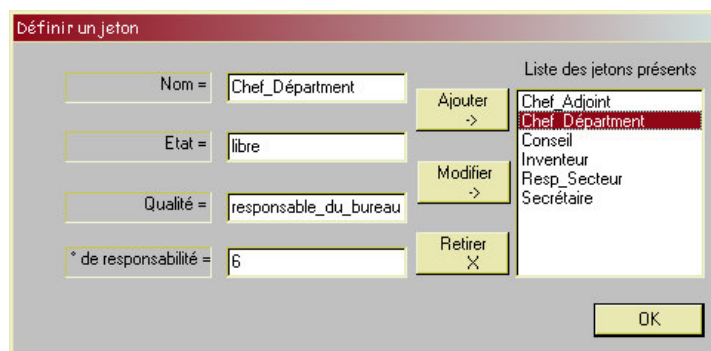


Figure IV.19. Définition d'acteurs dans SIMOH

La seconde étape consiste à définir les places relatives aux espaces de travail en indiquant leurs noms, le degré de responsabilité des rôles associés, la chronologie (c'est-à-dire que l'ordre d'apparition de l'espace de travail intervient dans la procédure, le commencement de la procédure étant signalé par le nombre 1), les acteurs et les documents présents dans la place. La figure IV.20 représente la définition de la place correspondant au bureau de l'inventeur dans lequel se trouvent l'inventeur et un avis (de brevetabilité). Cet inventeur possède un degré de responsabilité d'ordre 2 et intervient dès le début de la procédure.

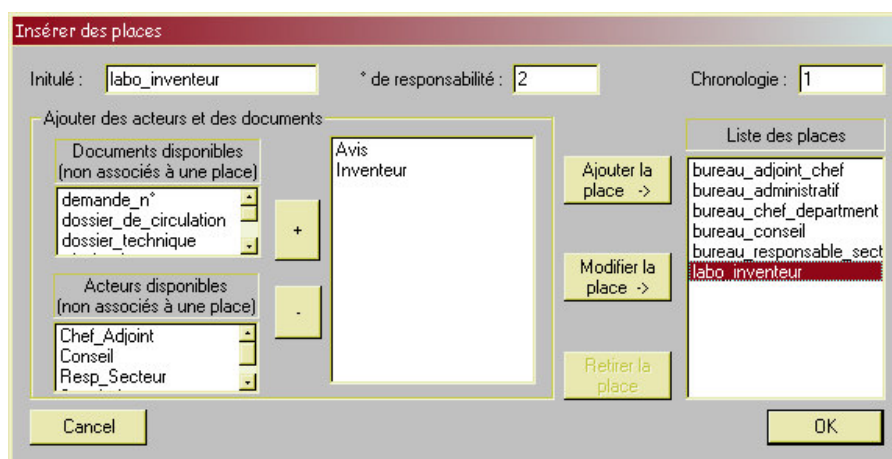


Figure IV.20. Fenêtre d'édition des places

La troisième étape consiste à définir les transitions entre les places en indiquant simplement les places en entrées et en sorties. La figure IV.21 représente une fenêtre d'édition des transitions, où la transition reliant le laboratoire de l'inventeur et le bureau du conseil en brevet porte l'intitulé "trans2".

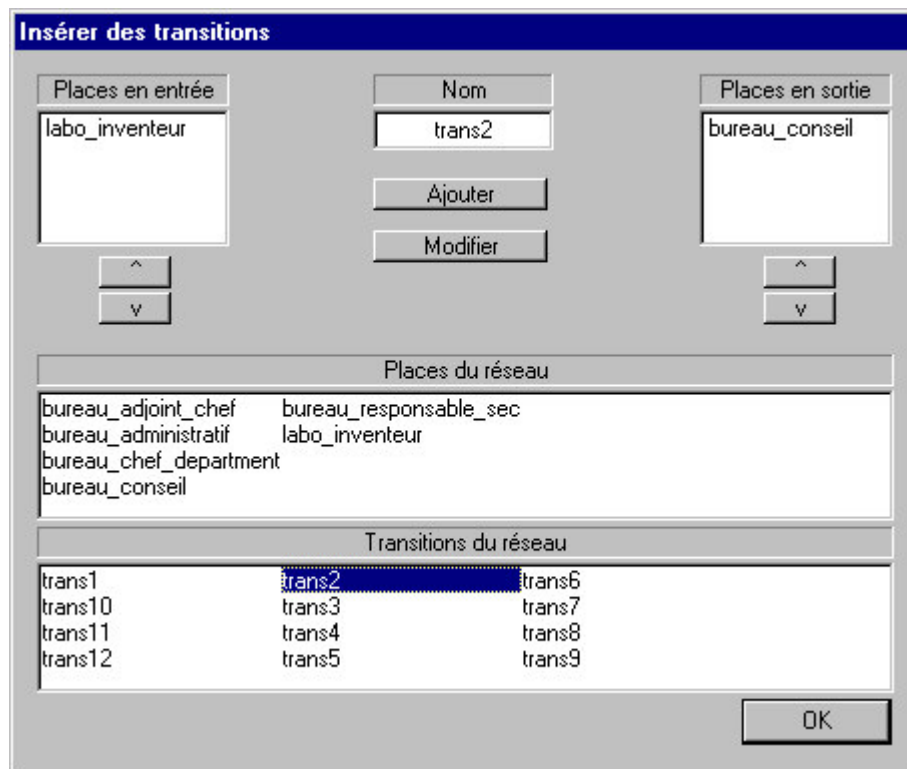


Figure IV.21. Exemple de définition des transitions dans SIMOH

La quatrième étape demande plus d'investissement de la part de l'utilisateur de SIMOH. Il s'agit de définir les règles de fonctionnement de l'ensemble de la procédure liées aux transitions, aux places et aux acteurs. Un ensemble de trois interfaces l'assiste dans cette démarche, associées respectivement aux activités, à leurs préconditions et à leurs actions. Par exemple, les figures IV-22 a,b,c représentent ces interfaces pour la définition d'une règle associée à un inventeur.

La dernière étape consiste à compiler le réseau, ce qui entraîne une vérification de la structure et des règles et une compilation de ces règles. Cette compilation consiste à les traduire sous une forme préfixée (par exemple, la postcondition "Avis devient traité ET dossier_technique apparaît" sera traduite en ["ET" ; "devient Avis traité" ; "apparaît dossier_technique"]). La validation des règles est effectuée par les acteurs qui vérifient le fonctionnement de la procédure simulée, par analogie avec le fonctionnement de la procédure réelle.

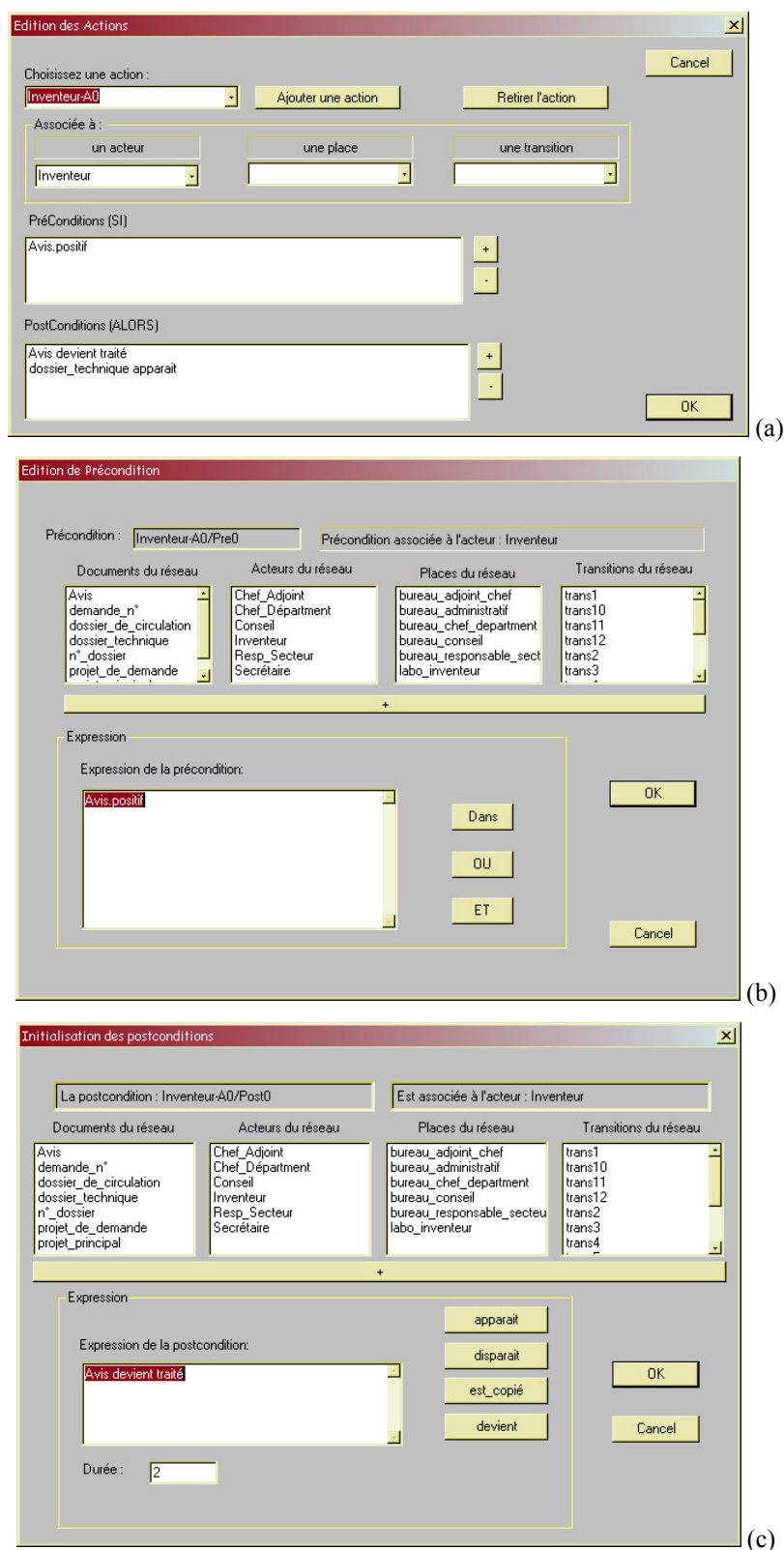


Figure IV.22. Edition d'une règle dans SIMOH

a - Edition d'une action

b - Edition d'une précondition

c - Edition d'une postcondition

IV.3.3 Conclusion sur les fonctionnalités de simulation

Le suivi de la simulation d'une procédure permet aux utilisateurs de prendre conscience des éventuels problèmes liés à la dynamique qui peuvent apparaître (tel que par exemple le blocage de la procédure suite à la surcharge de travail d'un acteur). A ce propos, à l'aide des fonctionnalités mises à sa disposition, l'utilisateur peut, à tout moment, stopper le déroulement du processus pour ajouter, supprimer ou modifier la valeur d'un élément, dans un but de recherche de solutions organisationnelles.

Lorsque les utilisateurs, principalement les responsables de la procédure simulée, ont vérifié la concordance avec le déroulement de la procédure réelle, les règles de fonctionnement de la procédure sont considérées comme validées et peuvent alors être réutilisées par les différentes composantes du SOHTCO dédiée à cette procédure.

Conclusion

Etre supportée par un atelier facile d'utilisation est très important pour l'utilisation répétée d'une méthode. L'atelier proposé dans ce chapitre est destiné à la modélisation et à la simulation de procédures administratives complexes dans le but d'obtenir les règles de fonctionnement de celles-ci. Il peut être, à ce propos, considéré comme un atelier de conception.

La souplesse d'utilisation permise par le logiciel VISIO, sur lequel se basent nos fonctionnalités de modélisation, permet de réutiliser la première partie de l'atelier afin de spécifier les procédures coopératives effectuées par les différents agents qui composent le SOHTCO.

Le chapitre suivant présente un cas d'application de la méthode MAMOSACO et de son atelier de conception sur des procédures du département de propriété industrielle de SOLVAY S.A.. Un exemple d'application est donné sur une procédure type de ce département.

Bibliographie du chapitre IV

- [**Adam 97**] Adam E., Vergison E., Kolski C., Mandiau R, Holonic User Driven Methodologies and Tools for Simulating Human Organizations, *Proceedings of the European Simulation Symposium*, (19-23 October 1997 : Passau, Allemagne), ESS'97, Ed. By Han W. & Lehmann A., pp. 57-61, 1997.
- [**Bastien 93**] Bastien J.M.C., Scapin D. L., Ergonomic Criteria for the Evaluation of Human-Computer Interface, *INRIA*, 156, 3, Juin 1993.
- [**Constantinidis 98**] Constantinidis Y., *Outils de construction du logiciel*. Paris: Hermes Science publications. 1998.
- [**Gransac 97**] Gransac J., Réalisation d'un Simulateur de Réseaux de Petri orientés Objets, Contribution à une méthode d'analyse de systèmes administratifs complexes, Rapport de DESS Technologie et Méthodes de l'Aide à la Décision, Université de Valenciennes et du Hainaut-Cambrésis, 1997.
- [**Grudin 94**] Grudin J., Eight Challenges for Developers. *Communications of the ACM*, 37, 1, 1994, pp. 93-105.
- [**Jalient 90**] *Génie Logiciel : les méthodes*. Paris : Armand Colin Editeur, 1990.
- [**Lant 94**] Lant T. K., Computer Simulations of Organizations as Experiential Learning Systems: Implications for Organization Theory. *In Computational Organization Theory*. Ed. by Carley K. M. & Prietula M.J., London : LEA publishers, pp. 195-215, 1994.

Chapitre V -

Application de la méthode MAMOSACO à des processus administratifs complexes et perspectives de recherche

INTRODUCTION	164
V.1 PRESENTATION GENERALE DU CAS D'APPLICATION	165
V.2 DESCRIPTION DE L'UTILISATION DE MAMOSACO SUR UNE PROCEDURE DE DEPOT DE BREVET	166
V.3 ANALYSE DES DONNEES RECUEILLIES ET DISCUSSION	187
V.4 PERSPECTIVES DE RECHERCHE	191
CONCLUSION	195
BIBLIOGRAPHIE DU CHAPITRE V	197

Introduction

Ce cinquième chapitre décrit une application concrète de la méthode MAMOSACO présentée précédemment. Cette application porte sur un département de propriété industrielle d'une grande compagnie, en l'occurrence SOLVAY S.A., qui possède une structure administrative complexe, c'est-à-dire que cette structure fait intervenir au moins une dizaine de rôles de niveaux différents sur plusieurs procédures. Ce type de structure peut être rapproché des structures holoniques ou en tout cas gagnerait à l'être car, selon A. Koestler, la nature holonique d'une organisation est une condition nécessaire de viabilité à long terme.

L'objectif de la méthode est la spécification et la conception d'un système intelligent distribué, ayant une architecture multi-agent holonique, pour l'aide à la coopération et à la gestion commune des documents de l'organisation. Ce système orienté holon pour l'aide au travail coopératif (SOHTCO) doit être adapté à l'organisation humaine, tout en étant composé d'un nombre de holons suffisants. Un nombre de holons peu élevé entraînerait une architecture centralisée et une impossibilité d'apporter une aide personnalisée à chaque acteur de l'organisation humaine. D'autre part, un découpage selon un trop grand nombre de holons complexifierait le système informatique et pourrait entraîner, du fait d'un trop grand nombre de communications entre les holons, une diminution de sa réactivité (dans un sens de vitesse d'exécution).

Ce chapitre décrit dans une première partie la structure du département de propriété industrielle sur lequel a été appliqué le travail présenté dans ce mémoire.

La seconde partie de ce chapitre présente un exemple d'application de MAMOSACO à l'analyse, la modélisation et la simulation d'une procédure de ce département. L'application de MAMOSACO a permis de spécifier un SOHTCO qui est également présenté dans ce chapitre.

La troisième partie présente une discussion sur base des données recueillies lors des phases d'analyse, de modélisation et de simulation du système réel et lors de la phase de spécification du SOHTCO.

Enfin, la dernière partie expose nos perspectives de recherche.

V.1 Présentation générale du cas d'application

Le travail présenté dans ce mémoire a pour origine une demande concrète d'un département de propriété industrielle d'une grande compagnie (SOLVAY S.A.). Cette demande portait sur un besoin en analyse et modélisation de ses principales procédures administratives et ce en vue de l'amélioration de celles-ci par l'utilisation d'outils intelligents.

Situé dans le centre de recherche de l'entreprise SOLVAY S.A. à Bruxelles, le département de Propriété Industrielle est fortement couplé au département de l'Information-Documentation qui a pour charge la gestion de l'information technique de l'Entreprise.

SOLVAY S.A. est une multi-nationale, fondée dès 1863, qui emploie actuellement 33.000 personnes sur un total de 46 pays et couvre quatre secteurs : le secteur chimique, le secteur plastique, le secteur transformation et le secteur pharmaceutique.

Le poids de l'histoire, ainsi que le découpage en quatre secteurs, influence le fonctionnement du département de Propriété Industrielle (P.I.). Celui-ci gère les marques et les brevets autour de trois stratégies principales [Notte 96] qu'il mène en partie avec le département de l'Information-Documentation :

- La stratégie offensive consiste à déposer des brevets et des marques afin de protéger les inventions,
- La stratégie défensive consiste à rechercher les marques et les brevets de tiers nuisant au patrimoine de l'Entreprise et à déclencher des procédures contre ces brevets ou ces marques. La défense des intérêts face aux attaques de tiers fait également partie de cette stratégie.
- La stratégie projective consiste à assurer une veille technologique afin d'aider les décideurs à anticiper sur le marché futur.

Les deux départements (PI et ID) sont composés chacun d'une trentaine de personnes pour un total d'environ 15 rôles (tels que ceux de chef du département PI-ID, chef du département PI, conseil en brevet, chargé d'étude, documentaliste-chimiste, secrétaire, dactylo, bibliothécaire ou imprimeur). Chaque rôle intervient dans un nombre variable de procédures, et possède donc des missions qui lui sont propres. Bien sûr, selon son niveau de responsabilité, un rôle possède des missions plus ou moins précises et à plus ou moins long

terme (par exemple, la mission de l'imprimeur intitulée « réaliser 5 copies du bulletin d'information » est une mission précisément définie, en revanche la mission du conseil en brevet « défendre SOLVAY en cas de litige » l'est moins).

En raison de sa structure proche du système holonique (le département peut être considéré comme une holarchie constituée de 15 couches de holons), ce double département, et donc ses procédures, est particulièrement adapté à l'application de MAMOSACO. Cependant, il n'est pas envisageable de modéliser l'ensemble du département. Ainsi, il a été décidé, par les responsables du département, de modéliser les procédures principales telles que les procédures de dépôt de brevet prioritaire, d'étude de brevetabilité, de protection des inventions, de demande de brevet transmise pour avis technique.

L'utilisation répétée de la méthode a été considérablement facilitée par l'atelier de conception, présenté dans le chapitre précédent, qui a permis non seulement d'obtenir des modèles et des règles de fonctionnement des procédures et de spécifier un système multi-agent holonique destinée à l'aide au travail coopératif (un SOHTCO) [Adam 00b], mais également d'améliorer sensiblement ces procédures (par exemple, une procédure a bénéficié d'un gain de temps de 25 jours).

Un exemple d'application de la méthode est fourni sur un extrait de la procédure de dépôt de brevet.

V.2 Description de l'utilisation de MAMOSACO sur une procédure de dépôt de brevet

La procédure de dépôt de brevet est déclenchée par un inventeur. Celui-ci a auparavant demandé une étude de brevetabilité de son invention. En cas d'avis positif, il rédige un projet de demande traité ensuite par les acteurs du département de Propriété Industrielle qui déposera la demande à l'Office national, européen ou mondial des brevets (tout dépend des revendications associées à l'invention).

Chaque utilisation de MAMOSACO nécessite de respecter l'enchaînement de ses phases. Ainsi, tout commence par l'analyse de la procédure effectuée auprès des acteurs. Une première modélisation de la procédure est ensuite effectuée, n'utilisant pas tous les modèles de MAMOSACO mais ceux permettant l'amélioration organisationnelle de la procédure par ses acteurs. L'ensemble des modèles n'est appliqué que lorsque l'organisation de la procédure, sa

structure, est stabilisée. La simulation est effectuée suite à cette seconde modélisation. Finalement, à partir des règles issues de la simulation est spécifié le SOHTCO dont les procédures coopératives sont définies par la réutilisation des fonctionnalités de modélisation de l'atelier de développement MAMOSACO. Dans le département de la Propriété Industrielle, plusieurs équipes ou personnes sont intervenues sur différentes phases de la méthode. Ainsi, l'analyse a été réalisée en partie par une équipe d'ergonomes, la modélisation sur les trois axes statiques (données, activités et traitements) a été effectuée en partie avec les acteurs de la procédure, quant à la modélisation de la dynamique et à la simulation de la procédure, ces phases font appels aux réseaux de Petri et ont été réalisées dans un but de spécification, voire de conception. La figure V.1 présente l'enchaînement des différentes phases de MAMOSACO appliquée au cas de la procédure de dépôt de brevet.

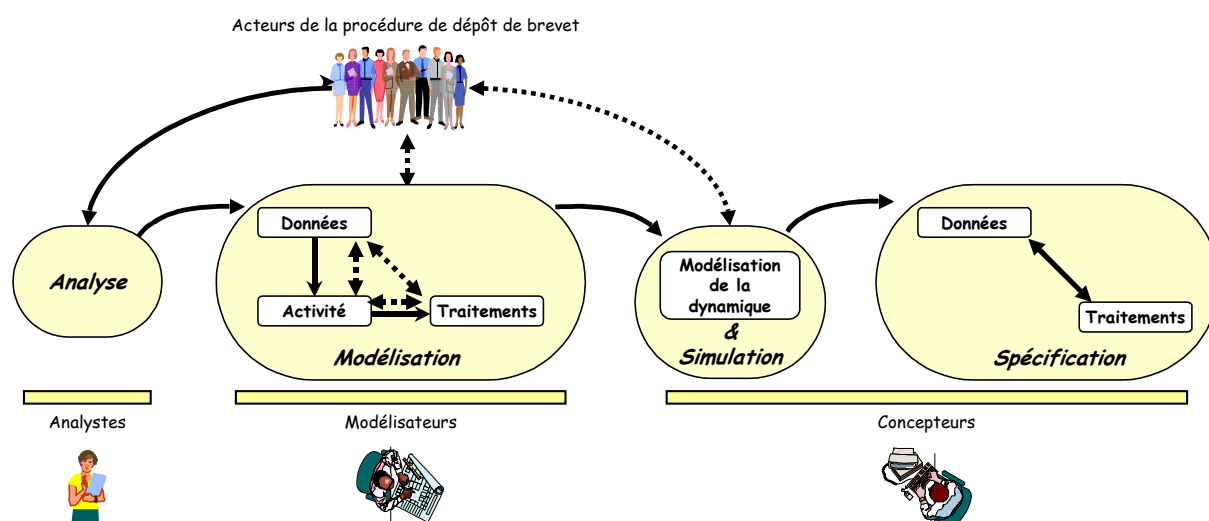


Figure V.1. Les phases de la méthode MAMOSACO appliquée au cas de la procédure de dépôt de brevet.

V.2.1 Analyse du fonctionnement de la procédure

Dans le cas du département de la propriété industrielle, les analyses orientées procédures, orientées rôles et orientées documents ont été effectuées par une équipe d'ergonomes [Notte 96], ainsi que par un étudiant de D.E.A. [Carrère 96]. Ces analyses ont été effectuées sur le terrain, auprès des acteurs du département.

En ce qui concerne l'analyse orientée rôle, le recueil des données s'est effectué non pas sur l'ensemble des acteurs du département, mais sur un groupe d'acteurs représentatifs des rôles qui composent le département.

Afin de mieux apprécier l'utilisation de l'atelier MAMOSACO, la procédure a été simplifiée (les modèles complets se trouvent en annexe C). La procédure présentée dans ce

chapitre met en jeu six rôles sur les dix initiaux. Les figures V.2 représentent des extraits de l'analyse orientée procédure (fig. V.2a) et de l'analyse orientée document (fig. V.2b) de la procédure de dépôt de brevet.

L'inventeur reçoit un avis positif de brevetabilité. Il constitue un dossier technique qu'il envoie au conseil en brevet.

Le conseil en brevet rédige alors un projet de demande de brevet quelquefois en commun avec l'inventeur. Il crée ensuite un dossier dit de circulation. Pour cela, il demande un numéro de dossier unique au secrétariat.

Le conseil en brevet place le projet dans le dossier et le transmet à l'adjoint-chef pour vérification.

L'adjoint-chef lit le dossier et le retourne en cas de modification. Sinon, il transmet ce dossier à la secrétaire.

La secrétaire copie le projet et le classe dans le dossier principal. Puis elle transmet le dossier de circulation à l'inventeur.

L'inventeur prend note des modifications apportées à la demande et transmet le dossier au responsable de secteur.

Le responsable de secteur complète le dossier de circulation et le retourne à la secrétaire.

La secrétaire enregistre le passage, afin de « tracer » le dossier, et le transmet au conseil en brevet.

Le conseil en brevet modifie et complète le texte du brevet et transmet le dossier au chef-adjoint pour vérification.

Le chef-adjoint vérifie le dossier et le retourne en cas de correction à apporter, sinon il le transmet à la secrétaire.

La secrétaire classe le dossier de circulation, et transmet le dossier principal, mis à jour, au chef de département.

Le chef de département prend connaissance de la demande et le retourne à la secrétaire.

Figure V.2a. Extrait de l'analyse orientée procédure du dépôt de brevet au sein du département de Propriété Industrielle

L'avis de brevetabilité contient la référence à l'invention et l'avis, positif ou négatif.

Le dossier technique possède la référence à l'invention, le type de domaine de l'invention, l'état de l'art et les caractéristiques de l'invention.

Le projet de demande de brevet contient la référence à l'invention, le résumé de l'invention, l'état de l'art du domaine de l'invention, et la description de l'invention.

Le dossier de circulation contient un n° de dossier interne, les noms des inventeurs, les mots clé relatifs à l'invention et inclut le projet de demande de brevet.

Le dossier principal contient un exemplaire du dossier technique et du projet de demande.

Figure V.2b. Extrait de l'analyse orientée document du dépôt de brevet au sein du département de Propriété Industrielle

Ce n'est qu'à partir de ces analyses qu'il est possible de modéliser la procédure en s'aidant pour cela de l'atelier MAMOSACO.

V.2.2 Modélisation de la procédure de dépôt de brevet

La modélisation n'est pas à effectuer par l'ergonome ou l'analyste, mais par une personne experte en gestion des processus (tel qu'un responsable de qualité par exemple). La première étape de la modélisation consiste donc, pour le modélisateur, à identifier les rôles et documents intervenant dans les différentes phases de la procédure.

V.2.2.1 **Matrice Activités-Rôle de la procédure de dépôt de brevet**

La procédure de dépôt de brevet, telle que le présente la figure V.3, peut être partagée en trois phases : la création de la demande de brevet, le traitement de cette demande, le classement de la demande. Ce découpage facilite la modélisation en ce sens qu'il permet de traiter séquentiellement des sous-procédures et non directement la procédure dans sa totalité. Le nombre de sous-procédures est laissé à l'initiative du modélisateur.

Six rôles peuvent être identifiés dans l'extrait de l'analyse, il revient au modélisateur de les classer selon le niveau de responsabilité qu'ils possèdent dans la procédure. Les rôles se composent dans notre cas en celui de secrétaire, d'inventeur, de conseil en brevet, de responsable de secteur, de chef adjoint et de chef de département.

La procédure de dépôt proposée à la figure V.1a peut être décomposée en trois parties :

- la création de la demande de brevet : dans laquelle intervient le bureau administratif, l'inventeur et le conseil en brevet;
- le traitement de cette demande par : le bureau administratif, l'inventeur et le conseil en brevet, le responsable de secteur et le chef-adjoint;
- le classement de la demande de brevet par : le bureau administratif, le chef-adjoint et le chef de département.

La grille Activités-Rôle (figure V.3) permet d'identifier le moment où la personne intervient dans la procédure. Dans le cas d'application, il s'avère que l'atelier de développement MAMOSACO permet de créer cette matrice relativement rapidement (en une dizaine de minutes en moyenne¹), au travers des fenêtres de saisies illustrées en figure V.4a et V.4b.

¹ Les temps présentés dans ce chapitre sont issus d'une moyenne observée lors des différentes formations à l'atelier MAMOSACO dispensées principalement au sein de SOLVAY. Avec de l'expérience, ces temps peuvent baisser sensiblement.

	chef de département	chef-adjoint	responsable secteur	conseil en brevet	inventeur	bureau administratif
création de la demande de brevet				X	X	X
taitement de la demande de brevet		X	X	X	X	X
classement de la demande de brevet	X	X				X

Figure V.3. Matrice Activités-Rôles de la procédure simplifiée de dépôt de brevet

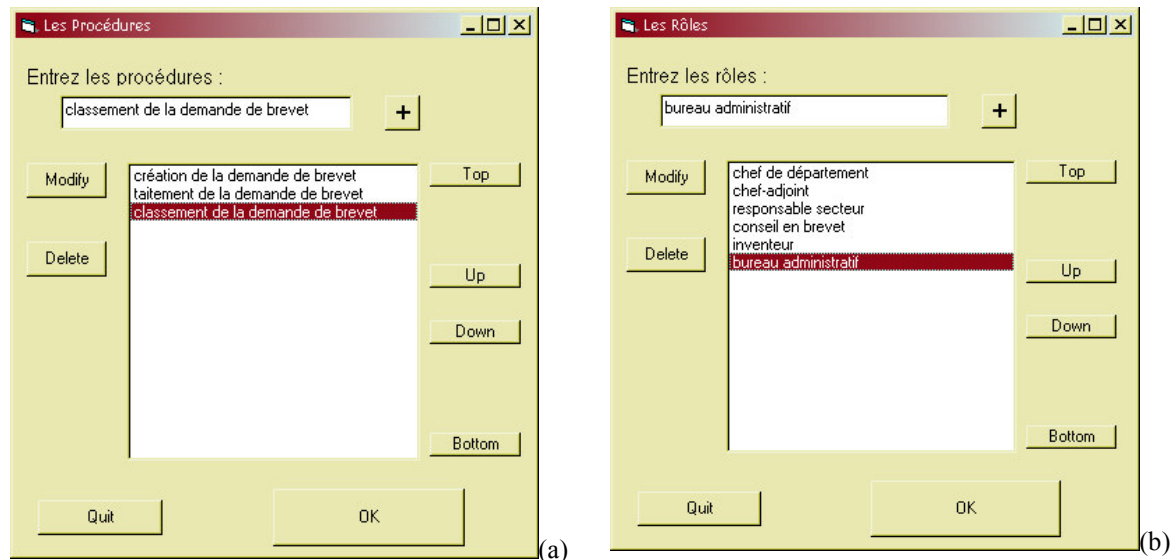


Figure V.4. Fenêtre d'édition
(a) des sous-procédures
(b) des rôles

V.2.2.2 Modèle des données de la procédure de dépôt de brevet

La seconde phase de la modélisation consiste en la modélisation des données. Celle-ci doit faire apparaître les relations entre les différents documents manipulés. A ce sujet, rappelons qu'il ne peut exister de document non lié par une relation, un lien de hiérarchie ou un lien d'agrégation à au moins un autre document.

Ce modèle doit également montrer les caractéristiques des documents, leurs différents champs, c'est-à-dire leurs propriétés. Ainsi, la figure V.5 représente le modèle des données correspondant à l'extrait de l'analyse de données présenté à la figure V.2b. En moyenne, il faut compter deux minutes par objet, soit environ quinze minutes pour le modèle présenté.

Le cas d'application montre que la modélisation objet est relativement aisée par l'atelier de conception, il s'agit tout d'abord de retranscrire les informations issues de l'analyse orientée document (rappelons que le modélisateur est aidé pour cela de fenêtres de saisie). Dans le cas

présenté, nous avons ajouté une information à l'objet associé au document technique signalant que la procédure de dépôt de brevet débute par le traitement de ce document.

La seconde étape consiste à placer les liens entre les objets. L'analyse a signalé que le dossier principal contient un exemplaire du dossier technique et du projet de demande de brevet. Un exemplaire de ce projet de demande se trouve également dans le dossier de circulation qui contient, outre des attributs, un numéro de dossier. L'avis de brevetabilité possède une référence à l'invention sur laquelle il porte et une nature indiquant si l'avis est positif ou négatif. Cet avis est lié par une association (par la référence à l'invention) au dossier technique. Une donnée supplémentaire apparaît ici dans l'analyse orientée rôle, il s'agit de la demande de numéro. Cette demande est associée au numéro et au dossier de circulation (par le champ auteur du dossier). Il n'apparaît pas ici de relation d'héritage, cette relation n'a été que très peu utilisée lors des modélisation des procédures du département.

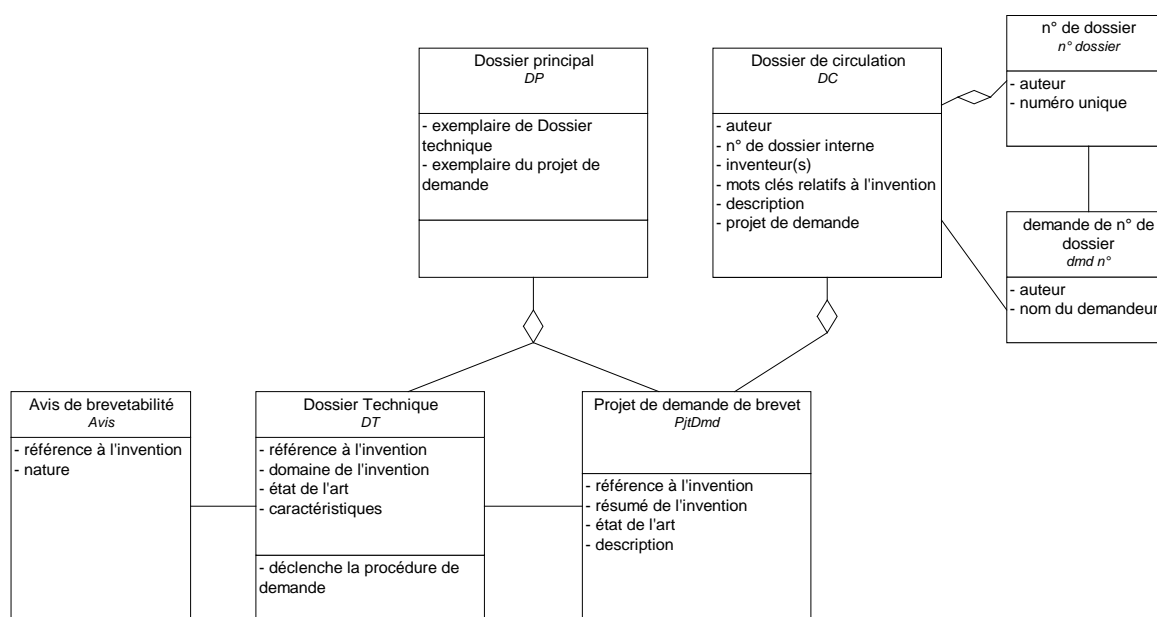


Figure V.5. Modèle des données de la procédure simplifiée de dépôt de brevet

En ce qui concerne l'atelier MAMOSACO, ces deux premières phases sont obligatoires car elles permettent de stocker le nom des rôles ainsi que leurs relations de responsabilité, le nom des sous-procédures ainsi que le nom et les abréviations des documents et données utilisées dans la procédure.

V.2.2.3 Modèle des activités et simplifications de la procédure dépôt de brevet

Généralement, en ce qui concerne la simplification et/ou l'amélioration des procédures, le modèle des activités est suffisant. C'est également le cas pour la procédure de dépôt de brevet.

La figure V.6 représente le modèle des activités de la procédure. Cette phase est relativement longue car le modélisateur doit en effet traduire par des boîtes d'activité, la description de la procédure réalisée lors de l'analyse. Il faut compter en moyenne 45 minutes pour la réalisation du modèle de la procédure totale (qui nécessite l'utilisation de 28 actigrammes répartis sur 10 rôles). Dans le cas de la procédure simplifiée, vingt minutes suffisent en moyenne à la réalisation du modèle.

La légende se crée automatiquement en faisant appel au modèle des données, il est donc important que ce modèle soit initialement le plus complet possible. La modélisation des activités consiste à interpréter les informations issues de l'analyse orientée procédure.

La première activité de la procédure est réalisée par l'inventeur qui constitue un dossier technique à partir d'un avis de brevetabilité positif. Une première boîte doit donc être placée sur la ligne correspondant à l'inventeur. Cette activité consomme l'avis en entrée et produit le dossier technique en sortie qui est ensuite envoyé au conseil en brevet. La principale difficulté rencontrée pour cette modélisation réside dans l'interprétation de l'analyse et notamment en ce qui concerne les notions temporelles. Par exemple, l'analyse indique que le conseil en brevet doit créer un dossier de circulation, mais que « pour cela, il demande un numéro de dossier unique au secrétariat ». Il y a donc ici un échange de données avec le secrétariat avant la création du dossier de circulation. La rédaction de l'analyse influe fortement sur la modélisation. Ainsi il est important lors de l'analyse d'utiliser le même terme pour désigner le même rôle, et non, pour des questions de rédaction, de faire l'usage de synonymes (tels que le adjoint-chef, puis le responsable de l'administration, puis le responsable des relations département-chercheurs par exemple, pour un seul et même rôle). Les retours posent également un problème lors de la modélisation, mais il s'agit plus d'un problème de lisibilité, c'est pourquoi nous proposons que ces retours (tel que « l'adoint-chef lit le dossier et le retourne en cas de modifications ») utilisent une notation pointillée (figure V.6).

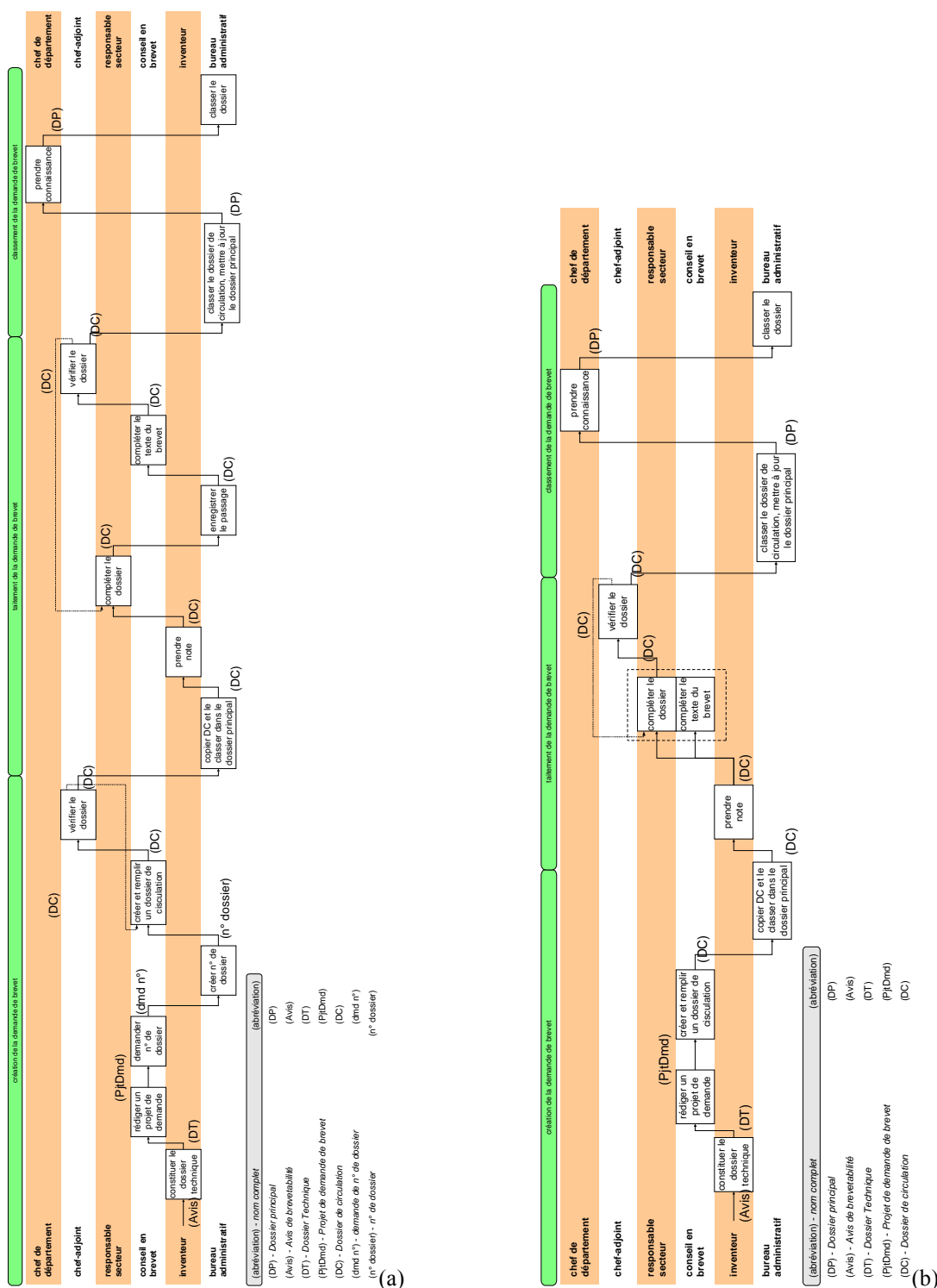


Figure V.6. Modèle de flux de données de la procédure de dépôt de brevet
 (a). Avant simplification
 (b). Après simplification

Les simplifications possibles ne sont pas évidentes dans cette procédure simplifiée. Le but de la phase d'amélioration est d'obtenir une organisation plus holonique de la procédure, c'est-à-dire dont les membres soient à la fois autonomes et coopérants. Une solution consiste à

diminuer la présence de la hiérarchie, ici de l'adjoint chef, de renforcer la coopération, ici au niveau du conseil en brevet et du responsable secteur et d'augmenter l'autonomie du conseil en brevet pour ce qui est de la création du dossier de circulation. La nouvelle procédure possède alors la structure présentée à la figure V.6. Cette simplification est réalisée dans un but pédagogique et reflète les améliorations généralement effectuées sur les procédures modélisées par MAMOSACO.

Bien entendu, les solutions doivent être approuvées par les acteurs. La démarche que nous avons suivie pour la simplification des procédures s'est déroulée en trois étapes d'une durée totale de deux heures :

- 1) présentation aux acteurs concernés du modèle de flux de données. Il y a alors une prise de conscience des acteurs de l'étendue de la procédure. En effet, peu de personnes connaissent le déroulement total des procédures, ce qui entraîne souvent des actes dupliqués tel que l'encodage ou le classement. Lors de cette première réunion, des solutions sont proposées par les acteurs.
- 2) présentation des modèles des solutions proposées. La meilleure solution ou celle convenant à la majorité des personnes est alors choisie.
- 3) validation par les acteurs de la solution choisie à l'étape précédente.

Cette démarche a permis d'obtenir un gain de temps de 15% en moyenne sur les procédures nécessitant une amélioration organisationnelle (donc sur ces procédures d'une durée variant de quelques jours à quelques semaines, le gain peut atteindre vingt jours, ce qui a été le cas pour une procédure du département). En outre, pour l'ensemble des procédures, le modèle des activités, par sa mise à plat de leur fonctionnement, sert de support lors de questions relatives aux procédures.

Par la suite, dans le cadre de l'exemple d'application, nous continuons les présentations des modèles sur base de la procédure simplifiée initiale.

V.2.2.4 Modèle de traitements de la procédure de dépôt de brevet

Le modèle des traitements est plus "précis" que le précédent modèle, en ce sens qu'il permet de représenter les conditions dans lesquelles une donnée circule entre les acteurs. Ce modèle est réalisé suite au modèle des activités. Cependant du fait de son niveau de détail plus élevé, le temps de création de ce modèle se monte également à 45 minutes en moyenne

pour la totalité de la procédure. Le modèle simplifié présenté à la figure V.8 nécessite, quant à lui, une vingtaine de minutes. La principale difficulté de ce modèle, par rapport au précédent consiste en l'identification des conditions de traitements des données. Ainsi, l'inventeur rédige un dossier technique lorsqu'il reçoit un avis positif de brevetabilité. Si l'avis est négatif, aucune action n'est effectuée par l'inventeur. Cette condition est symbolisée par un losange au début du modèle. Le modèle des traitements permet également de représenter les lieux de collaboration, c'est-à-dire les actions communes à plusieurs rôles. Ainsi il est indiqué dans l'analyse que "le conseil en brevet rédige alors un projet de demande de brevet quelquefois en commun avec l'inventeur", cette collaboration est symbolisée par un cadre regroupant les deux rôles (figure V.7).

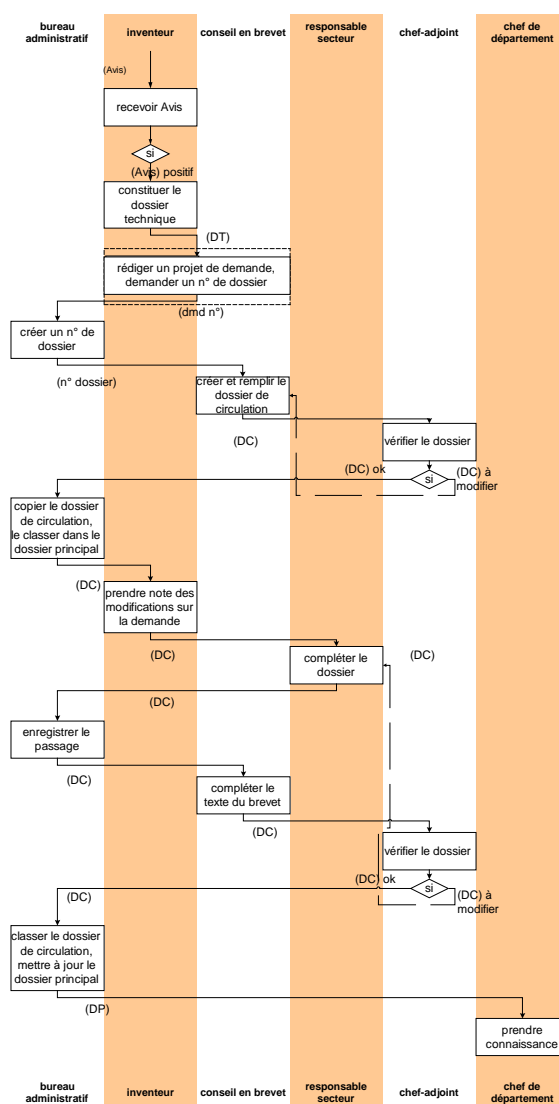


Figure V.7. Modèle de traitements de la procédure simplifiée de dépôt de brevet

Une question revient souvent à ce stade de la part des premières expériences des modélisateurs, à savoir l'intérêt de posséder les deux modèles (activités et traitements). En

fait, le modèle des activités représente tous les flux de données quelles que soient leurs conditions de passage, il offre une lecture aisée (de gauche à droite) des échanges qui se déroulent au sein d'une procédure. Cependant, il ne permet pas de représenter les différents chemins, les différentes variations d'une procédure. Le modèle des traitements permet cette représentation, au prix certes d'un temps de modélisation plus long et d'une représentation un peu plus complexe, se révélant fort utile d'après les acteurs impliqués.

V.2.2.5 Modèle dynamique de la procédure de dépôt de brevet

Le modèle dynamique est le plus détaillé des quatre modèles, il implique de décrire sous forme de règles les activités composant la procédure. La modélisation d'une procédure complexe dans son ensemble nécessite en moyenne deux heures de travail de la part du modélisateur, hors vérification de la cohérence.

La figure V.8 présente un extrait de la procédure de dépôt de brevet. Les circulations de données sont décrites au niveau des transitions (comme pourraient l'être les mouvements des acteurs si besoin). L'événement déclenchant la circulation, c'est-à-dire la condition nécessaire à la circulation de la donnée, est également indiqué au niveau des transitions. Ainsi, lorsque l'avis est donné, il parvient à l'inventeur. Les activités associées aux rôles, ainsi que leurs préconditions, sont situées au niveau des places, par exemple, lorsque l'avis est positif, l'inventeur doit constituer le DT (Dossier Technique). Lorsque ce DT est prêt, il est transmis au conseil en brevet. Ce dernier, à la réception du DT prêt peut créer un « pjt dmd » (projet de demande) ou demander des informations complémentaires à l'inventeur. Après avoir créé le projet de demande, le conseil doit demander un numéro de dossier au secrétariat. Pour des raisons de lisibilité, la place correspondant au rôle a été dupliquée (figure V.9).

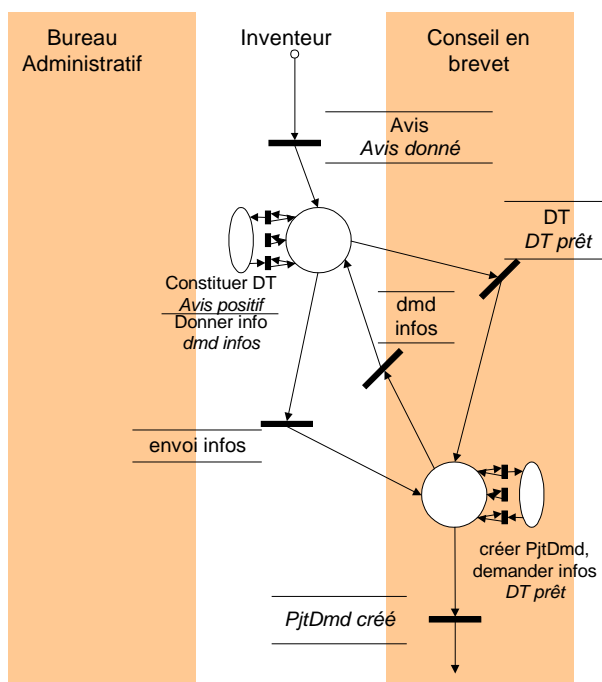


Figure V.8. Extrait du modèle dynamique de la procédure simplifiée de dépôt de brevet

Le modèle dynamique entier de la procédure simplifiée de dépôt de brevet est représenté sur la figure V.9. L'extrait montre la complexité de la modélisation de la dynamique. Cette tâche s'est révélée fastidieuse pour les modélisateurs disons "classiques" (tels que les chefs de projets et les responsables qualités). En fait, cette tâche ressort plutôt du travail des concepteurs plus habitués à manipuler des formalismes permettant un niveau de détail aussi fin. Toutefois, pour faciliter la définition des règles et des préconditions associées aux flux de données et aux activités effectuées par les rôles, une aide a été développée sous la forme d'un simulateur.

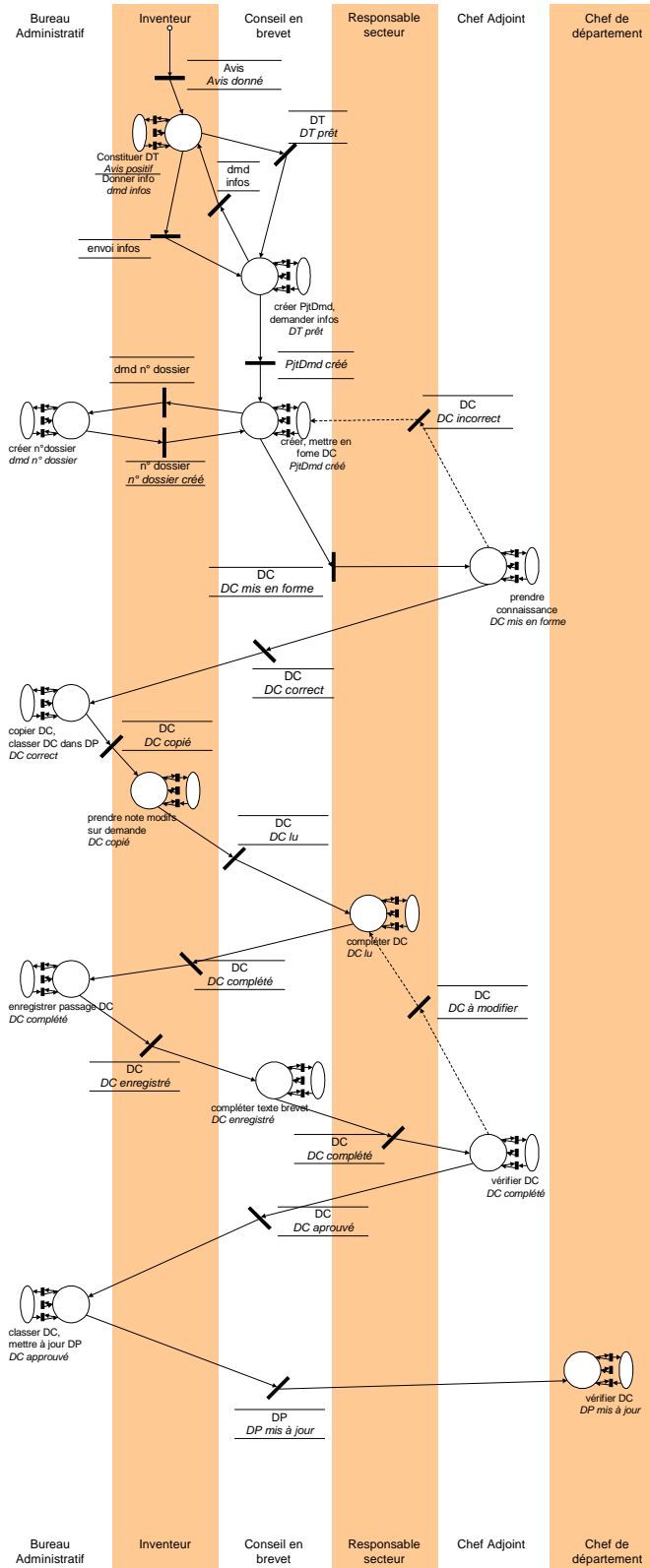


Figure V.9. Modèle dynamique de la procédure simplifiée de dépôt de brevet

Cependant, cette aide ne remplace pas la modélisation de la dynamique, celle-ci reste nécessaire et consiste en un travail de préparation à l'utilisation du simulateur. La création du modèle dynamique peut être négligée selon le cas pour passer à l'étape de simulation. Par

exemple, une procédure « courte », c'est-à-dire ne faisant pas intervenir un grand nombre d'acteurs (≤ 5) ou ne contenant pas un grand nombre d'activités (≤ 10) peut être directement simulée. Par contre, les procédures complexes², comme le dépôt de brevet par exemple, nécessite la modélisation de la dynamique (qui permet de mettre à plat le fonctionnement et de le « dérouler » dans le temps) avant de passer à l'étape de simulation.

V.2.3 Simulation de la procédure de dépôt de brevet prioritaire

L'utilisation du simulateur SIMOH nécessite de définir dans l'ordre : les documents, les acteurs, les espaces de travail et les transitions entre ces espaces (une transition correspond alors à un passage de document d'une place A vers une place B ; si un retour est nécessaire, une seconde transition est créée).

Au total, 31 objets ont été créés (sept objets document, six objets acteur, six objets place, douze objets transition) et onze règles ont été définies pour la procédure de dépôt de brevet.

La figure V.10 représente une copie d'écran du simulateur de l'atelier MAMOSACO. La structure du réseau montre que le bureau administratif est le point critique de la procédure : toute interruption de ses activités (par surcharge de travail ou manque de personnel par exemple) entraîne un retard pour l'ensemble de la procédure. Au début de la procédure, chaque acteur se trouve dans son bureau, l'inventeur dispose de l'avis de brevetabilité. Chaque élément est distingué par une couleur unique. A tout moment, le contenu d'une place peut être demandé ainsi que les caractéristiques des jetons présents dans la place.

² La complexité des procédures n'a pas été quantifiée. Toutefois, suite à plusieurs applications de la méthode et de son atelier, il nous est apparu que la durée de l'étape de modélisation s'accroît rapidement lorsque la procédure implique plus de cinq acteurs réalisant plus d'une dizaine d'activités.

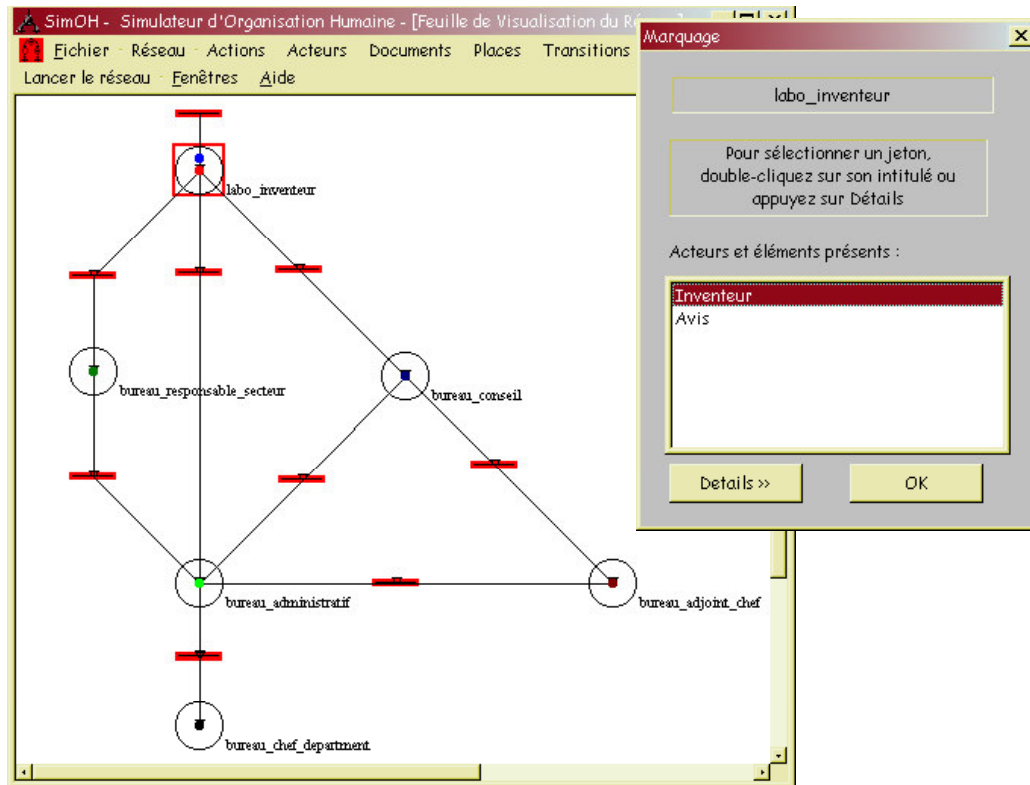


Figure V.10. Copie d'écran de la simulation de la procédure dépôt de brevet

L'ensemble du réseau, défini à l'aide des fenêtres de saisie présentées au chapitre IV, est sauvegardé dans un fichier texte. La figure V.11 représente un extrait de ce fichier pour la définition des activités de l'inventeur dans le cadre du dépôt de brevet (le fichier entier s'étend sur 780 lignes de texte). L'inventeur possède un degré de responsabilité d'une valeur égale à deux (ce degré correspond aux modèles, le premier degré étant porté par le bureau administratif et le dernier par le chef de département). L'inventeur est libre au début de la procédure. Un champ 'qualité' a été ajouté au niveau des jetons afin de permettre aux utilisateurs de placer leurs commentaires ; ici, l'inventeur est défini comme responsable de son bureau. Deux actions sont associées à l'inventeur :

- la première, 'Inventeur-A0', se déclenche lorsque l'inventeur a à sa disposition un avis de nature positive. L'avis est alors traité, en deux temps³, puis un dossier technique est créé par l'inventeur, en deux temps également. A l'issue de cette action, trois éléments se trouvent donc dans la place représentant le bureau de l'inventeur : l'inventeur, l'avis traité et le nouveau dossier technique.
- la seconde, 'Inventeur-A1', est exécutée lorsque l'inventeur reçoit le dossier de circulation (l'attribut copié indique simplement que le dossier vient d'être photocopié par un acteur

³ Dans le simulateur, les activités possèdent une durée mesurée en unité de temps. Dans le cas de la procédure de dépôt de brevet, cette unité prend la valeur d'une heure.

avant d'être envoyé à l'inventeur, cet attribut permet de distinguer les éléments au niveau des règles de passages associées aux transitions). L'inventeur consulte alors le dossier en deux temps.

<pre>[Acteur] nom = Inventeur degré de responsabilité = 2 etat = libre qualite = responsable_du_bureau [Actions associees] [Action] nom = Inventeur-A0 [Preconditions] Avis.positif [Fin Preconditions] [Postconditions] [Postcondition] expression = Avis devient traité duree = 2 ecoulement = 0 acteur = [Fin Postcondition] [Postcondition] expression = dossier_technique apparait duree = 2 ecoulement = 0 acteur = [Fin Postcondition] [Fin Postconditions] [Fin Action]</pre>	<pre>[Action] nom = Inventeur-A1 [Preconditions] dossier_de_circulation.copié [Fin Preconditions] [Postconditions] [Postcondition] expression = dossier_de_circulation devient consulté duree = 2 ecoulement = 0 acteur = [Fin Postcondition] [Fin Postconditions] [Fin Action] [Fin Acteur]</pre>
---	--

Figure V.11. Extrait du fichier texte définissant la procédure et son fonctionnement

Le cas d'application montre que la principale difficulté pour l'initialisation du simulateur réside surtout en la définition des règles, plus précisément des attributs associés aux éléments (documents et acteurs) qui circulent de place en place. En effet, le passage d'un élément par une transition est lié à son état (libre, copié par exemple). Il importe donc que cet état soit unique pour éviter une re-exécution des règles. Par exemple, si le jeton relatif à l'avis de brevetabilité reste à l'état positif et ne passe pas à l'état traité, la règle "Inventeur-A0" (Fig V.11) serait exécutée à chaque balayage du réseau.

L'avantage du simulateur est qu'il permet, outre la visualisation de la dynamique de la procédure, de définir rapidement les règles de fonctionnement de la procédure. L'adéquation de la simulation avec le déroulement de la procédure réelle permet de valider ces règles. Rappelons que ces règles sont réparties au niveau des sous-SOHTCO dédiés, chacun, à un acteur de la procédure. Le sous-SOHTCO associé à l'inventeur contient donc les règles décrites à la figure V.12 et les exécute afin de le guider (par exemple, si l'inventeur reçoit un

avis, le SOHTCO l'avertit de l'arrivée et lui propose de créer un dossier technique). La partie suivante propose une spécification d'un SOHTCO adapté au fonctionnement du département.

V.2.4 Spécification du SOHTCO

Dans le cas de la procédure de dépôt de brevet, outre le suivi des opérations (qui consiste à guider les acteurs en suivant les règles de fonctionnement), deux objectifs supplémentaires ont été proposés par ces acteurs :

- le premier objectif consiste en l'assistance à la transmission d'un document : un acteur demande au sous-SOHTCO qui lui est associé d'envoyer un document à un autre acteur de la procédure, le sous-SOHTCO doit alors vérifier la présence et la disponibilité du document puis la présence de l'autre acteur ; il doit ensuite envoyer le document au sous-SOHTCO de cet acteur qui notifiera l'arrivée du document.
- Le second objectif consiste en la recherche d'un document de la procédure : un acteur demande un document au sous-SOHTCO présent sur son poste, le sous-SOHTCO recherche le document en local, puis dans le voisinage et finalement, le cas échéant dans l'ensemble de la procédure.

Dans les procédures du département de Propriété Industrielle, le nombre de données circulant entre les acteurs est peu important (de l'ordre de trois documents pour une procédure), la base des sous-SOHTCO peut donc n'être constituée que de quatre agents holoniques (un agent responsable de la réception des données, un agent responsable de l'envoi de données, un agent gestionnaire de données et un agent interface). La figure V.12 présente la structure générale du SOHTCO (pour des raisons de lisibilité, tous les acteurs ne sont pas représentés). Dans notre cas, chaque holon (représenté par les petites sphères foncées) est atomique, c'est-à-dire qu'il n'est pas composé d'autres holons.

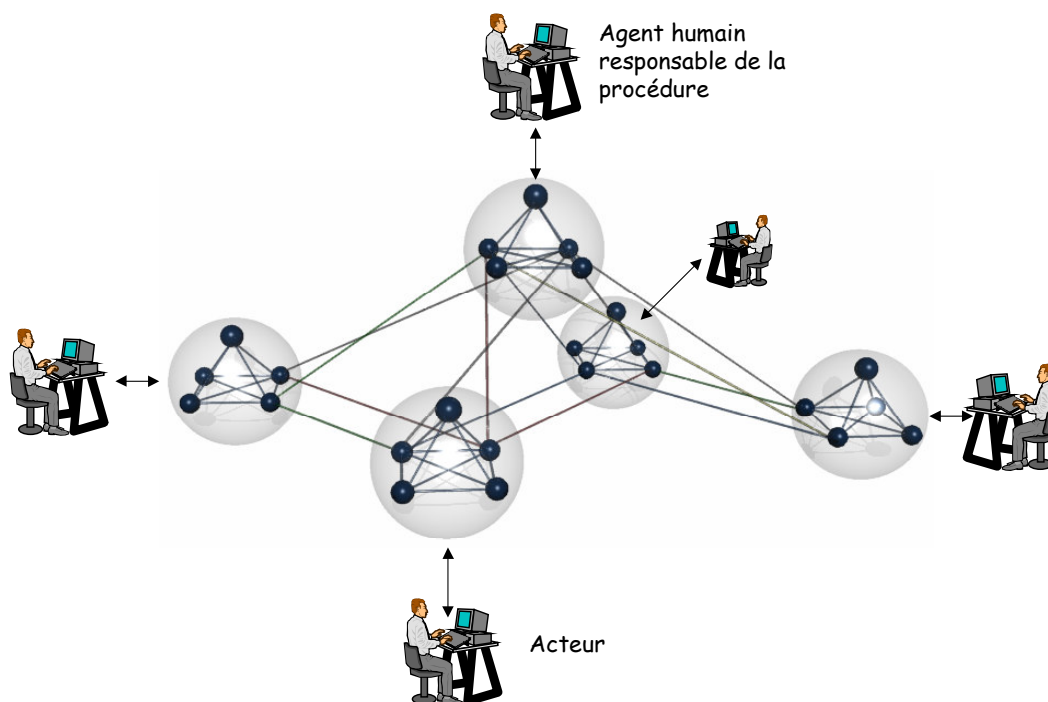


Figure V.12. Extrait de l'architecture du SOHTCO dédié à la procédure de dépôt de brevet

Les spécifications individuelles des agents holoniques restent celles présentée au chapitre III. Par contre les spécifications des fonctions coopératives (telles que l'envoi d'un document assisté par le SOHTCO, la recherche d'un document, l'auto-protection du SOHTCO, l'initialisation) sont spécifiques au fonctionnement du département, car les acteurs interviennent dans ces procédures.

Dans le cas du département de Propriété Industrielle, les acteurs possèdent en général une double expérience dans les domaines juridiques et chimiques. Il n'existe qu'un seul expert informatique au sein du département, c'est donc lui qui sera le gestionnaire du SOHTCO. Les éventuels problèmes survenus pendant l'exécution de la procédure sont signalés aux acteurs et au responsable humain de la procédure qui signale ensuite ce problème au gestionnaire du SOHTCO. Les figures V.13 et V.14 représentent la procédure d'initialisation du SOHTCO, chaque sous-partie est suivie d'une explication du fonctionnement.

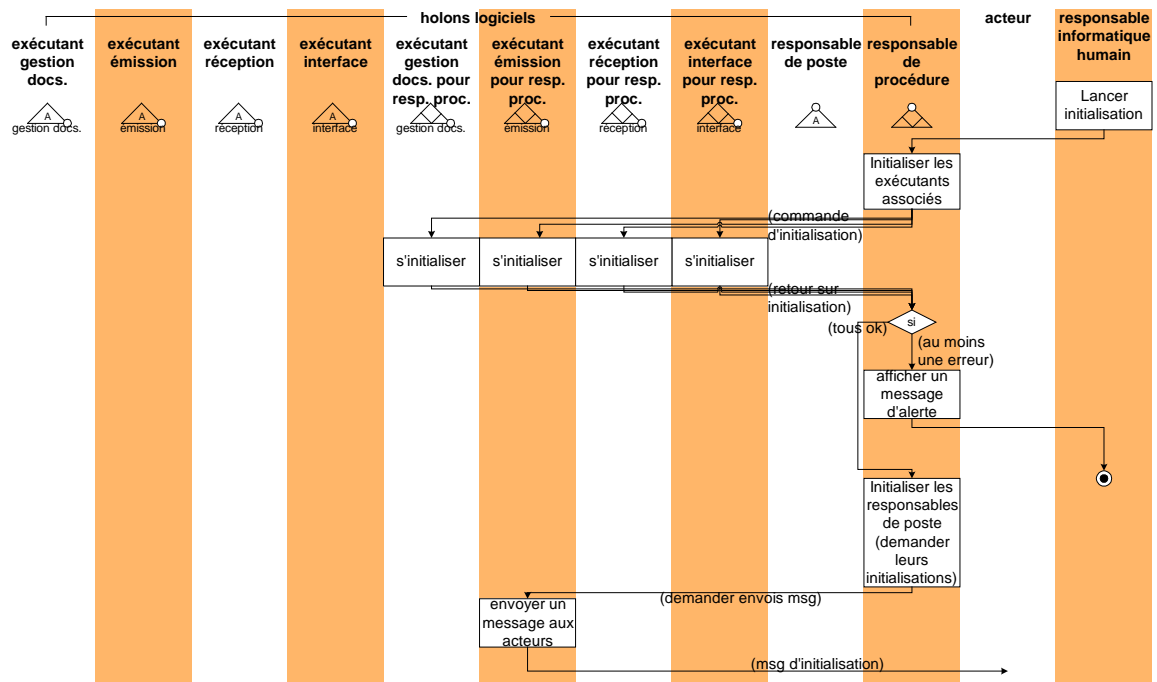


Figure V.13. Spécification de la procédure d'initialisation du SOHTCO (1/2)

L'initialisation du SOHTCO est effectuée par le responsable humain de la procédure, qui « lance », exécute le programme correspondant au responsable holonique de la procédure. Ce dernier va alors lancer à son tour les holons exécutants qui lui sont associés. En cas de problème à ce niveau, un message est alors affiché directement par le holon responsable à destination de l'acteur. Si les holons exécutants fonctionnent correctement, le holon responsable de procédure demande au holon responsable d'émission d'envoyer un message à tous les acteurs intervenant dans la procédure de dépôt de brevet.

La seconde partie de la procédure d'initialisation est représentée par la figure V.15. Nous traitons ici le cas de l'acteur A recevant la demande d'initialisation. Celui-ci exécute alors le logiciel correspondant au responsable de poste. Ce dernier exécute alors les holons exécutants dont il est responsable. Si un problème survient (impossibilité d'exécuter un holon, ou problème de liaison), un message est affiché par le responsable de poste. Dans le cas contraire, le responsable de poste demande au holon responsable d'émission d'envoyer un message au responsable humain de la procédure afin de l'avertir de la bonne initialisation du sous-SOHTCO.

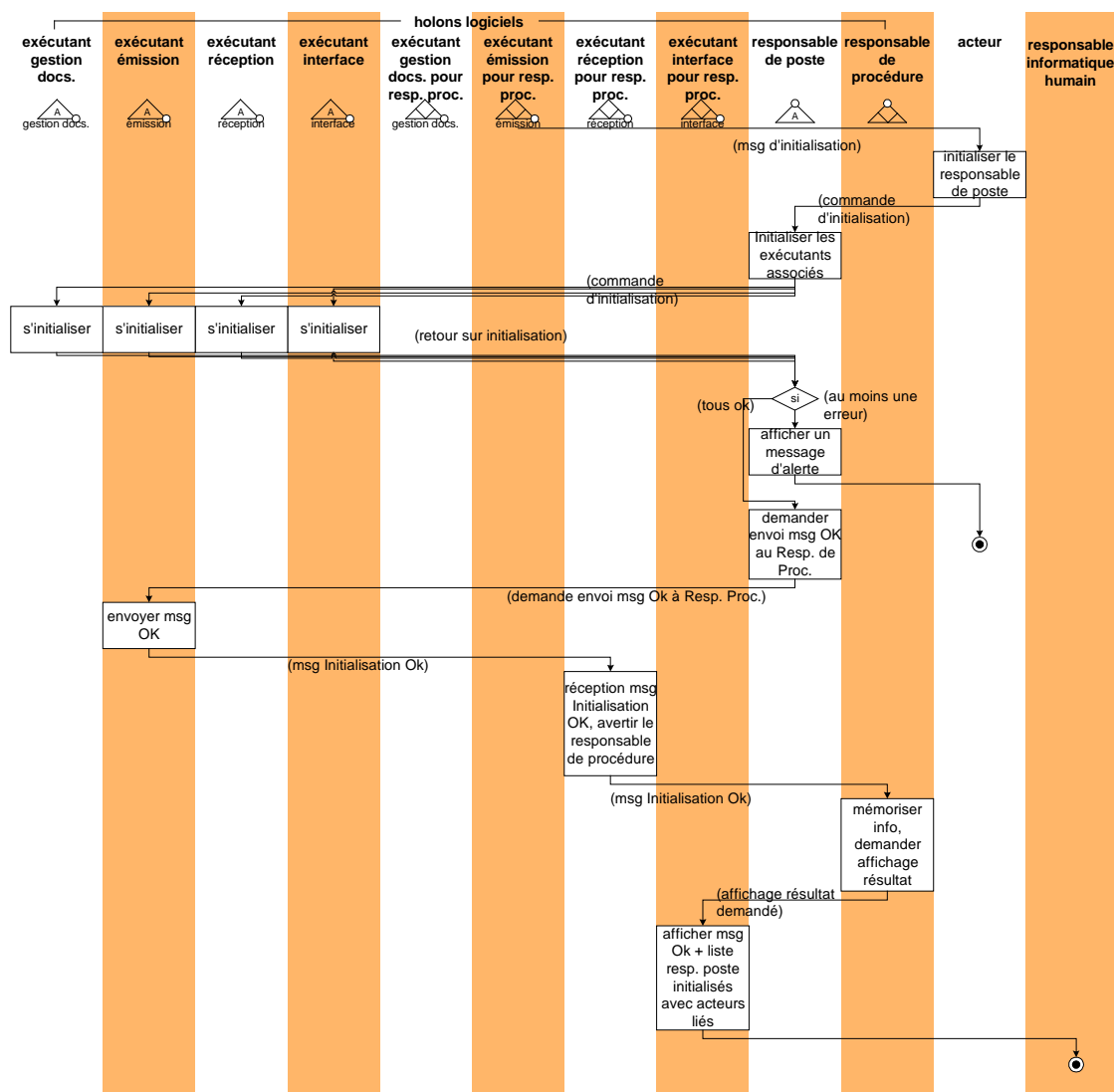


Figure V.14. Spécification de la procédure d'initialisation du SOHTCO (2/2)

V.2.5 Éléments de développement du SOHTCO

Afin de rester cohérent avec les développements informatiques du centre de recherche de l'entreprise SOLVAY S.A., la réalisation des SOHTCO a été effectuée grâce à l'atelier de développement Visual Basic 5. Les dernières versions de cet atelier permettent une programmation objet (bien que limitée car n'autorisant pas l'héritage) et surtout un ensemble de bibliothèques facilitant la programmation de fonctionnalités de messagerie.

Plusieurs classes ont ainsi été développées : la classe C_HolonResp (dont les holons responsables de poste sont des instances), la classe C_HolonExec (correspondant aux holons exécutants), la classe C_Acteur (qui correspond à la représentation qu'un holon a d'un acteur, c'est-à-dire le nom, l'adresse e-mail, le rôle, son état), la classe C_Procédure (correspondant à la représentation de la procédure, sous la forme indiquée en figure V.10, que possède un holon, notamment les responsable de poste et le responsable de procédure), la classe

C_RepHolon (la représentation d'un autre holon pour un holon donné, c'est-à-dire son nom, son adresse IP (internet), le nom de l'acteur associé), la classe C_Poste (correspondant à la connaissance que possède le holon sur le poste de travail, c'est-à-dire l'adresse IP, la quantité de mémoire et d'espace disque disponible, le nom des fichiers utiles à la procédure), et la classe KQML (contenant l'ensemble des définitions et fonctions nécessaires à la communication inter-holons).

Si l'implémentation des fonctionnalités de communication inter-poste est facilitée par les bibliothèques Visual Basic, la communication inter-holon au sein d'un même poste pose plus de problèmes et nécessite la création d'une classe particulière de gestion d'événements. Ainsi, lorsqu'un holon envoie un message à un autre holon appartenant au même sous-SOHTCO, un événement est envoyé au gestionnaire qui le traite et le retourne à l'agent bénéficiaire.

Ce traitement est possible par l'utilisation du formalisme KQML (pour Knowledge and Query Manipulation Language) [Finin 94] qui propose un ensemble de termes et une grammaire, c'est-à-dire des actes de langage, relativement complets et simple d'utilisation. Par exemple, la figure V.15 représente le dernier message KQML que le holon responsable du poste A (RespPosteA) envoie au responsable d'émission du poste (ExecEmissionA), lui demandant (par la performative 'Tell') d'envoyer un message de bon fonctionnement au responsable humain de la procédure ('send msg ok to RespHumain').

```
[Tell
  :sender RespPosteA
  :receiver ExecEmissionA
  :content send msg ok to RespHumain]
```

Figure V.15. Exemple de communication KQML dans le SOHTCO

L'atelier Visual Basic ne permet pas réellement de concevoir les fonctionnalités d'inférence qui permettent au simulateur de traiter les différents niveaux de règles, ou alors au prix d'une profonde refonte du code. C'est pourquoi la gestion des règles n'a pas été portée sur cet atelier, mais reste implémentée en Visual C++.

Un prototype de SOHTCO, constitué de deux sous-SOHTCO placés sur deux postes différents, a donc été développé en Visual Basic. Ceci nous a permis de tester l'architecture holonique du système, notamment en ce qui concerne les communications. Il en résulte que les communications intra-poste entre les holons d'un même sous-SOHTCO ne sont pas pénalisants, en ce sens que leurs temps de traitements ne sont pas perceptibles pour

l'utilisateur qui communique au maximum une dizaine de documents en une journée. En ce qui concerne les connaissances déclaratives sur la procédure (les règles de fonctionnement), celles-ci ont été extraites du fichier de règles et codées "en dur" dans les deux holons responsables de postes du prototype, le Visual Basic ne permettant pas une réutilisation efficace du code créé en Visual C++ lors du codage du simulateur.

V.3 Analyse des données recueillies et discussion

L'application de MAMOSACO a été bien perçue dans le département de la Propriété Industrielle, et ceci grâce à notre démarche participative. En effet, les acteurs ont apprécié le fait d'échanger leurs idées autour des modèles lors de la phase d'amélioration organisationnelle des procédures. La clarté des modèles, c'est-à-dire leur lisibilité, leur accessibilité, y a joué un rôle essentiel ; elle a en fait été améliorée au fil des utilisations de la méthode. En effet, les premiers modèles proposés aux acteurs n'étaient pas structurés par lignes ou par colonnes comme le sont les actuels modèles d'activités, de traitements et le modèle dynamique. La figure V.16 est un exemple des premiers modèles présentés aux acteurs ; bien que lisible pour une procédure non complexe (ici, la gestion du courrier entrant au sein du département de propriété industrielle), cette représentation devient vite illisible pour des procédures de plus grande envergure. C'est donc par des interactions permanentes concepteurs-utilisateurs que la modélisation a évolué.

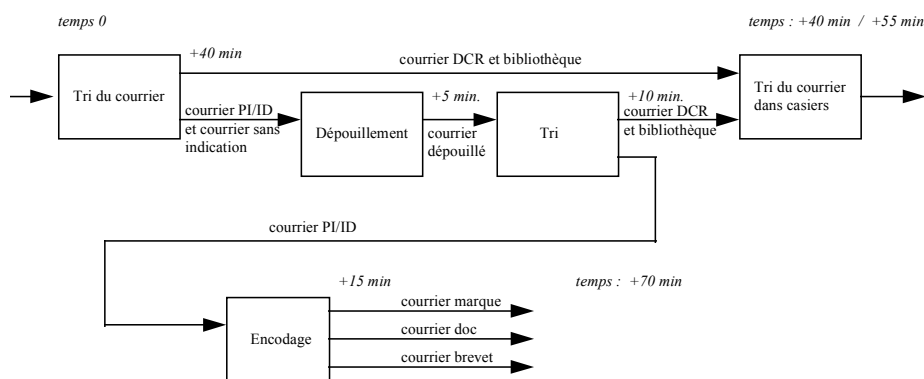


Figure V.16. Exemple des premiers modèles utilisés (cas d'une procédure de gestion de courrier entrant)

Le modèle dynamique est lui aussi issu des premières applications de la méthode MAMOSACO. En effet, celle-ci ne comportait initialement que trois modèles statiques⁴, la représentation de la dynamique est rapidement apparue nécessaire suite à la modélisation de la

⁴ En effet, MAMOSACO résulte initialement de l'intégration des méthodes OMT (remplacée ensuite par UML), SADT et CISAD qui ne proposent pas de quatrième représentation dynamique sous la forme de réseaux de Petri de haut niveau.

procédure de gestion du courrier entrant. Le décalage entre le modèle des activités et la réalité a été signalé par l'acteur de cette procédure : si les durées de chaque phase, signalées aux côtés des actigrammes, sont correctes, le total de ces durées est inférieur à la durée réelle de la procédure. Cette différence, cet écart de temps (de l'ordre de 30 à 45 minutes), trouvait sa raison dans les interruptions de travail (tels que la gestion des fax, les appels téléphoniques, les demandes de renseignements directes), que seul un modèle dynamique permet de représenter.

Une démarche participative a donc été menée non seulement à l'étape de modélisation, mais également à l'étape de spécification de la méthode. Ceci a permis d'obtenir, de façon relativement rapide (car utilisant l'intégration de méthodes existantes), une méthode adaptée au cas d'application et aux acteurs de l'organisation étudiée.

En ce qui concerne les utilisateurs de l'atelier de conception, et notamment les acteurs responsables de la qualité des procédures, il a été reproché le fait de ne pouvoir faire figurer sur les différents modèles la durée des activités ou des traitements. Cette durée n'est utilisée qu'à l'étape de simulation, est n'est pas nécessaire à la spécification de SOHTCO. Néanmoins, cette information pourrait être utilisée pour faciliter l'étape d'amélioration organisationnelle des procédures. Si les lecteurs des modèles apprécient le modèle des données issu de UML pour sa lisibilité, les modélisateurs sont plus réservés quant à l'instantiation des diagrammes objets. En effet, la formalisation des données sous forme d'objets nécessite une bonne analyse préalable orientée documents qui est souvent omise par les utilisateurs de MAMOSACO.

Il ressort de l'utilisation de MAMOSACO au sein du département de Propriété Industrielle les modélisations complètes de quatre procédures jugées critiques par la direction du département. L'une d'entre elles a bénéficié d'une amélioration considérable permettant de réduire de 20 jours son temps de traitement. Ces modèles entrent dans le cadre de la capitalisation des connaissances de l'entreprise en ce sens qu'ils représentent une trace écrite, une mémoire, des procédures pouvant être réutilisées lors de réunions ou de formations de nouveaux acteurs par exemple. De ces quatre procédures, la procédure de dépôt de brevet s'est avérée la moins complexe (en terme du nombre d'activités entreprises par les acteurs), elle a donc été choisie pour vérifier l'adéquation de la structure d'un SOHTCO avec ce type de procédure.

Un SOHTCO a donc été spécifié suite à l'analyse, la modélisation et la simulation de cette procédure. Ce SOHTCO est adapté au fonctionnement d'une organisation administrative complexe dans lequel le nombre de données échangées dans les procédures est limité (une dizaine en moyenne) mais dont le temps de traitement est relativement long (suivant l'importance du brevet, le dépôt de brevet varie de un à trois mois, ce temps dépend surtout de l'Office national, européen ou mondial des brevets). La spécification du SOHTCO nécessite une petite adaptation au type de fonctionnement de l'organisation, notamment en ce qui concerne l'acteur portant la responsabilité du fonctionnement du SOHTCO, celui-ci pouvant être le responsable de la procédure ou un responsable informatique de l'organisation. Dans ce dernier cas, les deux personnes doivent être averties en cas de problèmes. L'implémentation des agents constituant le SOHTCO a été facilitée par leur structure générique qui permet, à partir d'une classe principale de dériver tous les éléments du SOHTCO. Toutefois, bien que le langage Visual Basic ait permis un prototypage rapide des fonctionnalités de communications, il convient d'utiliser un autre langage évolué (tel que le C++ ou le Java par exemple) pour implémenter toutes les fonctionnalités du SOHTCO.

Cette première application nous permet de compléter les différents niveaux de réutilisabilité mis en évidence dans le chapitre III. Ainsi le niveau structurel et individuel s'enrichit des objectifs possibles de recherche d'information et d'envoi d'informations assistés par les SOHTCO. Le niveau coopératif bénéficie maintenant de cinq procédures coopératives (initialisation, recherche de document, envoi d'un document, protection du SOHTCO et suivi des actions) pour le cas où les sous-SOHTCO ne sont constitués que de 5 agents. Ce cas est le plus simple, il pourra donc servir de base pour les prochaines spécifications de SOHTCO. En effet, si un autre cas d'application nécessite l'utilisation de plus de 4 agents exécutants, il suffira d'augmenter ces procédures. Enfin, dans le niveau environnement (de programmation), se trouvent dorénavant les classes et procédures Visual Basic permettant d'implémenter les holons. La figure V.17 présente les apports en ce qui concerne les trois niveaux de la base réutilisable du modèle ∇ (nabla) adapté à la conception de SOHTCO.

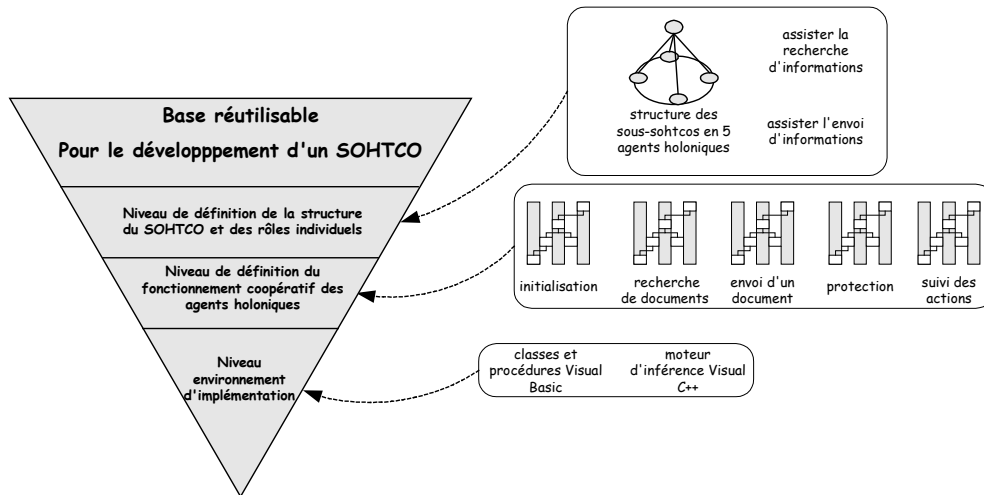


Figure V.17. Apports à la base réutilisable du modèle nabla

V.4 Perspectives de recherche

L'application de la méthode MAMOSACO relative à des cas concrets de processus administratifs complexes a permis de mettre en évidence les points forts de cette méthode ainsi que les besoins non couverts et nécessitant par conséquent de nouvelles réflexions.

V.4.1 Perspectives associées à MAMOSACO

La démarche participative proposée dans MAMOSACO a été bien accueillie par les acteurs du département et a conduit ces acteurs à mettre en évidence divers besoins tels que la possibilité de représenter la durée des activités et des traitements. Cette demande émane surtout des personnes utilisant MAMOSACO dans un but d'amélioration de la qualité des procédures [Adam 00c]. Des modifications des modèles, notamment du modèle des activités, sont en cours. Actuellement, deux diagrammes Pert sont proposés soit pour indiquer les débuts et fins prévus ou réels des activités, soit pour indiquer les débuts et fins au plus tôt et au plus tard des procédures. Les figures V.18a et V.18b représentent ces diagrammes Pert destinés à remplacer, dans le cadre d'une démarche de qualité, les actigrammes du modèle des activités de MAMOSACO.



Figure V.18. Diagrammes Pert envisagés pour la représentation d'une activité
 (a) en indiquant les dates de début et de fin prévue et réelle
 (b) en indiquant les dates de début et de fin au plus tôt et au plus tard

D'autres demandes de la part des utilisateurs de l'atelier de conception, mettent en évidence le besoin d'une vérification automatique de la cohérence entre les différents modèles. En effet, celle-ci est actuellement effectuée totalement par les modélisateurs. Ceci provient d'un choix délibéré de notre part afin d'impliquer totalement l'utilisateur dans la modélisation et d'éviter une modélisation assistée par ordinateur qui banaliserait la création de modèles. Toutefois, il semble qu'une vérification sommaire de la cohérence entre les modèles (tels que la vérification du sens des flux entre les modèles d'activité et de traitements par exemple) serait bien perçue de la part des modélisateurs et permettrait de faciliter la gestion de la cohérence sans pour autant l'automatiser.

De même, il n'existe pas de liaisons automatiques entre le modèle dynamique et le simulateur ; ceci est principalement dû à la différence d'atelier de développement, les fonctionnalités de modélisation étant basées sur le logiciel Visio et le langage Visual Basic,

les fonctionnalités de simulation étant conçues en Visual C++. Il n'est pas envisageable de générer automatiquement toutes les règles à partir du modèle dynamique, car les règles utilisées dans le simulateur sont plus précises. Par contre, il est envisageable de générer la structure du réseau (les places, les transitions, les acteurs et les documents) directement à partir des fonctionnalités de modélisation de l'atelier MAMOSACO.

MAMOSACO n'a pas été seulement utilisée pour la modélisation de procédures administratives complexes. Elle nous a également permis de modéliser des procédures composées d'un nombre moins important d'activités et faisant intervenir un nombre plus restreint d'acteurs. Ces modélisations ont été effectuées soit dans un but d'amélioration organisationnelle (en tentant de suivre le modèle holonique), soit dans un but de détection des points critiques en vue de l'apport de solutions informatiques. MAMOSACO a été conçue pour la modélisation d'organisations d'architecture holonique s'échangeant des informations discrètement (par opposition aux flux de données continus). Il serait intéressant, à ce titre, de tester MAMOSACO sur d'autres organisations holoniques que les organisations administratives complexes (telles que certaines chaînes de montage par exemple).

V.4.2 Perspectives associées aux SOHTCO

L'application de MAMOSACO à un processus administratif particulier, le dépôt de brevet, a permis d'obtenir des règles de fonctionnement du processus qui peuvent ensuite être réutilisées par les différents éléments d'un SOHTCO. En effet, un SOHTCO, dont les sous-SOHTCO sont constitués de 5 agents holoniques a été proposé, et ses procédures coopératives ont été spécifiées. La réalisation des agents holoniques le composant a été facilité par la structure récursive et donc reproductible du système. Cependant l'atelier de développement Visual Basic n'a pas permis de réaliser toutes les fonctionnalités, notamment, celle d'inférence sur les règles issues de la simulation. Il est donc souhaitable et envisageable de porter l'architecture holonique sur d'autres langages évolués tels que Java par exemple ; un travail a d'ailleurs été entamé à ce sujet [Broudoux 00]. Dans ce contexte, la création d'une plate-forme de développement de SOHTCO en java (éventuellement en utilisant d'autres plates-formes existantes dont les structures se rapprochent des SOHTCO) serait également envisageable.

En ce qui concerne les fonctionnalités de notre SOHTCO, plusieurs améliorations possibles sont apparues. En effet, le cas d'étude utilise un flux constant de documents.

Toutefois, d'autres cas d'application pourraient posséder des flux très variables (entraînant la gestion de non plus deux à trois documents mais d'une cinquantaine de documents par exemple). Une adaptativité du SOHTCO est alors nécessaire et plusieurs agents de bas niveaux doivent être créés, ce qui entraîne alors la création dynamique d'un responsable pour chaque type d'agents générés à bas niveaux. Il est également possible de permettre l'adaptativité du SOHTCO par migration des tâches associées aux agents de bas niveau (un agent d'émission d'informations se transformerait en agent responsable de réception en cas d'afflux soudain de messages). Une telle adaptativité qui nécessite une gestion et une distribution dynamique des tâches est donc encore à étudier.

La notion de stabilité est très importante dans les systèmes holoniques. Toutefois une trop grande stabilité de l'ensemble du système holonique risquerait paradoxalement de déstabiliser le système en le rendant moins adaptable aux fluctuations de l'environnement (il est quelquefois nécessaire qu'une partie d'une organisation soit temporairement déstabilisée, par des contraintes de production par exemple, afin de permettre aux systèmes de s'adapter au nouvel environnement). Un système de gestion de la stabilité répartie au niveau de chaque agent holonique est donc à proposer pour permettre d'obtenir des SOHTCOs à la fois stables et adaptatifs.

Les SOHTCOs permettent de distribuer l'aide à la gestion d'information dans une organisation administrative complexe. Les sous-SOHTCO, situés sur chaque PC utilisateur, possèdent chacun une interface afin de communiquer avec les acteurs. Cette interface est représentée par un agent de bas niveau contrôlé par l'agent responsable de poste. Cette notion de séparation entre la représentation des éléments d'une interface et du contrôle de ces éléments a déjà été proposée (par les modèles PAC [Coutaz 90] et MVC [Goldberg 84] par exemple). Il serait intéressant de confronter le SOHTCO (qui peut être vu comme une interface multi-utilisateurs dont les éléments sont répartis dans un réseau d'entreprise), avec les différentes architectures d'interface existantes.

Enfin, il sera intéressant d'étudier l'utilisation de MAMOSACO et des SOHTCO à d'autres domaines en relation avec le traitement de l'information, notamment le domaine de la veille technologique. En effet, une veille technologique est généralement effectuée par un groupe de personnes recherchant des informations et les collectant dans des bases de données afin de les distribuer. Il s'agit donc d'un travail coopératif effectué par un groupe de personnes pouvant être distribuées, physiquement éloignées, et nécessitant une cohérence dans la gestion

de l'ensemble de l'information recueillie. Un SOHTCO, proposant une gestion de la cohérence au niveau global et une aide répartie au niveau des acteurs, semble donc être adapté à cette problématique. Une étude est actuellement en cours à ce propos dans le cadre d'une convention de recherche entre SOLVAY S.A. et l'Université de Valenciennes et du Hainaut-Cambrésis⁵.

⁵ Adam E., Conception et mise en œuvre d'un réseau d'agents intelligents pour la veille « marchés ». Convention d'étude entre SOLVAY S.A. et VALUTEC S.A., rapport de recherche, Novembre 2000.

Conclusion

Le cas d'étude présenté dans ce chapitre a permis de décrire une utilisation de la méthode MAMOSACO, ainsi qu'une spécification d'un SOHTCO adapté, conduisant à l'implémentation des agents holoniques le constituant.

Les premiers retours sur MAMOSACO sont très positifs, autant du point de vue de l'atelier le supportant que des modèles qui en découlent. L'application de MAMOSACO a mis en évidence des points critiques dans les procédures et a mené, pour certaines, à une amélioration considérable grâce à une démarche participative très bien perçue. En ce qui concerne l'atelier basé sur le logiciel Visio, les aides et fenêtres de saisie mises au point en Visual Basic ont été bien accueillies de la part des modélisateurs qui n'ont pas eu de difficultés à modéliser quelques procédures sans posséder de formation poussée dans le domaine de la gestion des processus et du Génie Logiciel. L'utilisation du simulateur pour produire les règles de fonctionnement des procédures est plus destinée au concepteur du système multi-agent SOHTCO. En effet, cette phase requiert la pratique sinon des réseaux de Petri, du moins de la formalisation de processus sous la forme de règles.

En ce qui concerne la spécification, le cas d'étude a permis de spécifier la structure d'un SOHTCO adapté, ainsi que son fonctionnement coopératif. Cette spécification a montré la nécessité de prendre en considération l'organisation humaine dans laquelle devra fonctionner le système, notamment pour ce qui concerne les notions de maintenance (associée à la fonction conservative des agents holoniques) à court terme (à qui devra être donné l'alerte en cas de dysfonctionnement). L'implémentation d'un prototype de SOHTCO en Visual Basic a permis de valider l'architecture en ce qui concerne les communications, ce qui paraissent être le point crucial des SOHTCO au vu des règles de communication holarchiques.

Nous pensons que différents points peuvent être améliorés, tant en ce qui concerne la méthode qu'en ce qui concerne l'atelier de modélisation, le simulateur et les SOHTCOs. Différentes perspectives ont d'ailleurs été proposées à ce sujet à la fin de ce chapitre. Toutes ne sont pas situées au même niveau. Ainsi, les perspectives à court terme concernent l'amélioration de l'atelier de spécification et particulièrement la vérification des liens entre les différents modèles ou la création d'un lien entre la modélisation et le simulateur SIMOH. Les perspectives à long terme concernent, entre autres, l'implémentation totale d'un SOHTCO sur un autre cas réel d'application que le cas présenté, la création d'une plate-forme de

développement de SOHTCO, l'ajout de fonctionnalités d'adaptation à l'environnement du SOHTCO, de fonctionnalités de négociation entre les agents holoniques.

Bibliographie du chapitre V

- [**Adam 00b**] Adam E., Mandiau R., Kolski C., HOMASCOW : a Holonic Multi-Agent System for Co-operative Work. *Proceedings of the first First International Workshop on Industrial Applications of Holonic and Multi-Agent Systems in conjunction with the Eleventh International Conference on Database and Expert Systems Applications*, (6-8 September 2000, Greenwich, London, UK), Holomas'2000.
- [**Adam 00c**] Adam V., Le processus de dépôt de brevet prioritaire à SOLVAY S.A., NOH, Bruxelles. Mémoire de DESS "Démarche Qualité et Ingénierie de la Décision". Université d'Aix-Marseille, Faculté d'Économie Appliquée, 2000.
- [**Broudoux 00**] Broudoux L., Payen L., Réalisation d'un Système Orienté Holon pour l'aide Au Travail Coopératif. Rapport de projet IUP G.E.I.I. Automatisation Intégrée et Systèmes Homme-Machine. Université de Valenciennes et du Hainaut-Cambrésis, Juin 2000.
- [**Carrère 96**] Carrère P., Production, gestion et diffusion électronique de documents, Ré-ingénierie des processus d'activités d'un département de Propriété Industrielle, Rapport de DEA, Université Technologique de Compiègne & SOLVAY S.A, 1996.
- [**Coutaz 90**] Coutaz J., *Interfaces homme-ordinateur : conception et réalisation*. Paris : Bordas, 1990.
- [**Finin 94**] Finin T., Mckay R.F.D., McEntire R., KQML as an Agent Communication Language, *Proceedings of the Third International Conference on Information and Knowledge Management*, (Novembre 1994 ; Gaithersburg, USA), CIKM'94, ACM Press, 1994.
- [**Goldberg 84**] Goldberg A., *Smalltalk-80, the interactive programming environment*. Addison-Wesley, 1984.
- [**Notte 96**] Notte D., Le Département de Propriété Industrielle : Etude du domaine, rapport interne SOLVAY S.A.& Ergodin, 1996.

Conclusion Générale

Conclusion générale

L'étude d'une organisation humaine pour y apporter une aide intelligente à la gestion de documents ne doit pas se limiter à la modélisation des flux de données. Il est nécessaire de prendre en compte l'aspect organisationnel du système. Si la structure générale du système peut être représentée sous forme de réseau, de graphe, le comportement de chaque partie du système reste à définir, afin de pouvoir apporter l'aide adéquate. Cette aide devant être distribuée au niveau de chaque acteur du système (de l'organisation humaine), il devient essentiel qu'elle soit basée sur les recherches entreprises en I.A.D., ou plus précisément qu'elle utilise les concepts développés dans le domaine des systèmes multi-agent. Une organisation orientée agent doit alors être conçue afin d'assister les acteurs de l'organisation humaine.

Le travail présenté a montré le caractère pluridisciplinaire de la recherche et de la spécification d'une organisation agent adaptée à une organisation humaine. En effet, il est nécessaire, afin de mieux appréhender la complexité des organisations, de s'inspirer des résultats de disciplines telle que la sociologie. Puis, afin de représenter, de mettre à plat, le fonctionnement de l'organisation, les outils et méthodes du génie logiciel sont indispensables. Enfin, comme nous l'avons signalé plus haut, pour suivre une démarche cohérente dans la mise en place de l'organisation multi-agent, il convient d'intégrer au plus tôt les concepts développés dans ce domaine de l'I.A.D. Cette pluridisciplinarité de la démarche apparaît également sur le terrain, lors de la mise en œuvre de la méthode proposée. Ainsi il est conseillé que trois équipes collaborent à la conception du système multi-agent : une équipe d'ergonomes, réalisant l'analyse du travail effectuée par les acteurs au sein de l'organisation humaine ; une équipe de modélisateurs, réalisant les différents modèles correspondant aux analyses ; et une équipe de concepteurs, chargée de la mise en œuvre du système intelligent.

L'étude et la recherche du modèle organisationnel général applicable à la plupart des systèmes complexes ont fait l'objet du premier chapitre. Elles ont mené au choix du modèle holonique, définissant tous systèmes complexes en tant que hiérarchie flexible, c'est-à-dire en tant que structure pyramidale dont les membres possèdent des niveaux de responsabilités et une relative autonomie. Il a été montré dans ce chapitre que les règles organisationnelles définissant ce type de système peuvent être intégrées dans les organisations multi-agent. Le système d'assistance peut donc ainsi posséder une structure proche du système étudié (l'organisation humaine).

Outre l'importance d'utiliser un modèle sociologique facilitant la compréhension et l'amélioration du fonctionnement de l'organisation humaine, ce travail a également montré l'importance de l'utilisation d'un cadre de comparaison pour choisir et/ou construire la méthode adéquate. En effet, sur base d'une confrontation rigoureuse de différentes méthodes, il est possible d'intégrer les parties les plus pertinentes des méthodes testées dans une méthode adaptée au cas d'étude. Il est important de signaler qu'il n'est pas obligatoire de suivre une seule et même méthode tout au long d'un projet, mais qu'il est possible de coupler, de façon pertinente, différentes méthodes afin de répondre au mieux aux besoins. Le second chapitre s'est intéressé à la recherche d'une méthode permettant la mise à plat du fonctionnement d'organisation complexe de type holonique. Cette étude s'est donc effectuée au travers d'un benchmark, d'un cadre de comparaison, défini sur cinq dimensions. De cette étude, il a été montré que, parmi les méthodes testées, aucune ne répondait à tous les critères de sélection. Par contre, les conclusions sur cette comparaison ont permis de poser les premiers éléments d'une méthode adaptée aux organisations administratives complexes.

L'intégration des parties les plus pertinentes des méthodes testées a été montrée dans le troisième chapitre. Cette intégration a permis de concevoir la méthode MAMOSACO, Méthode Adaptable de Modélisation de Systèmes Administratifs COMplexes. La modélisation et la simulation du fonctionnement des organisations administratives complexes permettent d'obtenir trois niveaux de règles constituant autant de niveaux de règles d'assistance. L'organisation multi-agent holonique utilisant ces règles a également été présentée dans ce chapitre. Ce SOHTCO (Système Orienté Holon pour l'aide au Travail COopératif) est spécifié en deux phases : une phase de spécification des trois rôles joués par les agents holoniques ; une phase de spécification du fonctionnement coopératif réutilisant les modèles de MAMOSACO. Le caractère récursif du comportement du système multi-agent holonique, qui se répète au niveau de chacune de ses parties, a facilité cette étape de spécification.

Le quatrième chapitre a présenté l'atelier de modélisation MAMOSACO, atelier basé sur le logiciel VISIO et nécessaire à l'application de la méthode. Dans ce chapitre a également été décrit le SIMulateur d'Organisation Humaine SIMOH, qui a facilité la rédaction des règles de fonctionnement des procédures du système étudié.

Le cinquième chapitre a détaillé une application de MAMOSACO sur une procédure réelle d'un département de Propriété Industrielle de l'entreprise SOLVAY S.A. Les retours

concernant la méthode ont été très positifs (certaines procédures modélisées ont bénéficié d'améliorations conséquentes diminuant leurs temps de traitement). D'autre part, les résultats concernant la spécification d'un SOHTCO et la réalisation d'un prototype sont très encourageants (le comportement général d'un SOHTCO étant récursif, c'est-à-dire reproductible sur chacune de ses parties, sa spécification est facilitée, de plus, son adéquation avec l'organisation humaine à laquelle il s'applique permet d'intégrer les acteurs humains dans cette spécification). Des applications et des perspectives de recherches, tant en ce qui concerne la méthode MAMOSACO (par exemple l'amélioration de l'atelier de conception, l'augmentation des modèles de MAMOSACO), qu'en ce qui concerne les agents holoniques SOHTCO (par exemple l'ajout de capacités d'adaptativité, la spécification de fonctionnements généraux pour d'autres domaines de gestion de l'information), sont proposés dans ce dernier chapitre. Il est à noter que certaines applications dans d'autres domaines sont actuellement en cours.

Références Bibliographiques

-
- [**Abed 94**] Abed M., Angue J.C., A new method for conception, realisation and evaluation of man-machine interfaces. *Proceedings IEEE Systems, Man and Cybernetics conference*, (2-5 Octobre 1994 ; San Antonio, USA).
- [**Adam 95**] Adam E., Modèle d'interface réactive dans les systèmes tutoriels de découverte guidée. Mémoire de D.E.A., Université Paul Sabatier, Toulouse, Juin 1995.
- [**Adam 97**] Adam E., Vergison E., Kolski C., Mandiau R., Holonic User Driven Methodologies and Tools for Simulating Human Organizations, *Proceedings of the European Simulation Symposium*, (19-23 Octobre 1997 : Passau, Allemagne), ESS'97, Ed. By Han W. & Lehmann A., pp. 57-61, 1997.
- [**Adam 98a**] Adam E., Mandiau R., Vergison E., Parameterized Petri nets for modelling and simulating human organisations in a workflow context. *Proceedings of the Workflow Management Workshop (WfM) in the 19th International Conference on Application and Theory of Petri Net*, (juin 1998 : Lisbonne, Portugal), ICAPTN'98.
- [**Adam 98b**] Adam E., Mandiau R., Vergison E., Méthode adaptable basée sur la modélisation de processus pour l'analyse et l'optimisation de systèmes coopératifs dans l'Entreprise. *Actes du Colloque International Ergonomie et Informatique Avancée*, (Novembre 1998 : Biarritz), Ergo'IA 98, Biarritz : ESTIA/ILS, 1998.
- [**Adam 99a**] Adam E., Specifications of intelligent human-machine interfaces for helping cooperation in human-organizations. *Ergonomics and user interfaces, volume 1*. Ed. by Bullinger H.J. & Ziegler J. London: Lawrence Erlbaum Associates. pp. 311-315. 1999.
- [**Adam 99b**] Adam E., Mandiau R., Kolski C., Approche holonique de modélisation d'une organisation orientée workflow : SOHTCO. *Ingénierie des Systèmes multi-agents*. Ed. par Gleizes M.P. et Marcenac P., Paris: HERMES Science publications, pp. 121-134, 1999.
- [**Adam 00a**] Adam E., Kolski C., Etude comparative de méthodes du Génie Logiciel en vue du développement de processus administratifs complexes. *Revue Génie Logiciel*, 49, 2000.
- [**Adam 00b**] Adam E., Mandiau R., Kolski C., HOMASCOW : a Holonic Multi-Agent System for Co-operative Work. *Proceedings of the first First International Workshop on Industrial Applications of Holonic and Multi-Agent Systems in conjunction with the Eleventh International Conference on Database and Expert Systems Applications*, (6-8 September 2000, Greenwich, London, UK), Holomas'2000.
- [**Adam 00c**] Adam V., Le processus de dépôt de brevet prioritaire à SOLVAY S.A., NOH, Bruxelles. Mémoire de DESS "Démarche Qualité et Ingénierie de la Décision". Université d'Aix-Marseille, Faculté d'Économie Appliquée, 2000.
- [**Agimont 96**] Agimont G., Le Strugeon E., Mandiau R., Libert G., Parametrized petri nets for organizational simulation and systems design. *Proceedings of the Second International Conference on the Design of Cooperative Systems*, (12-14 June 1996 ; Juan-les-Pins), COOP'96, Ed. by COOP Group, INRIA Press.

-
- [**Alberto 93**] Alberto T., Combemale P., *Comprendre l'entreprise. Théorie, gestion relations sociales*. Paris : Nathan, 1993.
- [**Alliot 95**] Alliot J.M., Schiex T., *Intelligence Artificielle et Informatique Théorique*. Toulouse : Edition Cépaduès, 1995.
- [**Aloisi 95**] D'Aloisi D., Giannini V., The Info Agent : an interface for supported users in intelligent retrieval. *Proceedings of the Network Information Retrieval - Italia Conference*, (21-23 November 1995 ; Milan, Italia), NIR-IT'95.
- [**Alvares 98**] Alvares L.O., Menezes P.B., Demazeau Y., Problem decomposition: an essential step for multi-agent systems. *Proceedings of the 10th International Conference on Systems Research, Informatics and Cybernetics* (August 1998 ; Baden-Baden, Allemagne), ICSRIC'98.
- [**Arafa 99**] Arafa Y., Charlton P., Fehin P., Mamdani A., Personal Service Assistants with personality. *Human Computer Interaction – Communication, Cooperation and Application Design*, Ed. by Bullinger H.J. & Ziegler J., London : LEA, pp. 147- 151, 1999.
- [**Arai 97**] Arai T., Aiyama Y., Sasaki Y., Holonic Storage: an assembly and storage cell by manipulation using environment. *proceedings of the 29th CIRP International Seminar on Manufacturing System - New Manufacturing Era*, (11-13 May 1997 ; Osaka, Japan), pp.221-226.
- [**Bailly 87**] Bailly C., Challine J.F., Gloess P.Y., Ferri H.C., Marchesin B., *Les langages orientés objet: Concepts, Languages et Applications*. Cepadues editions, 1987.
- [**Balbo 94**] Balbo S., Evaluation ergonomique des interfaces utilisateur : un pas vers l'automatisation. Thèse de doctorat, Université de Grenoble I, Septembre 1994
- [**Bannon 91**] Bannon L.J., Schmidt K., CSCW: Four characters in Search of a Context. *Studies in CSCW, Theory, Practice and Design*, Ed. by Bowers J.M. & Benford S.D., Nottingham, 1991.
- [**Barthet 88**] Barthet M.F., *Logiciels Interactifs et ergonomie. Modèle et méthodes de conception*. Paris : Dunod Informatique, 1988.
- [**Barthet 95**] Barthet M.F., The DIANE Method and its connection with MERISE Method. *Proceedings IEA World Conference "Ergonomic design, interfaces, products, Information"*, Rio de Janeiro, Brazil, pp. 106-110, 16-20 October 1995.
- [**Bastide 00**] Bastide R., Spécification comportementale par réseaux de Petri : application aux systèmes distribués à objets et aux systèmes interactifs. Habilitation à Diriger des Recherches, Université de Toulouse I, 18 janvier 2000.
- [**Bastien 93**] Bastien J.M.C., Scapin D. L., Ergonomic Criteria for the Evaluation of Human-Computer Interface, *INRIA*, 156, 3, Juin 1993.
- [**Benaissa 93**] Benaissa M., Une démarche de conception, réalisation et évaluation d'IHM : application au projet ferroviaire ASTREE. Thèse de doctorat. Université de Valenciennes et du Hainaut-Cambrésis, Décembre 1993.

- [Bensaid 97]** Bensaid N., Mathieu Ph., A Hybrid and Hierarchical Multi-Agent Architecture Model. *Proceedings of the Practical Application of Intelligent Agents and Multi-Agent Technology*, (21-23 April 1997 ; London, U.K.), PAAM'97.
- [Bensaid 99]** Bensaid N., Contribution à la réalisation d'une modèle d'architecture multi-agent hiérarchique. Thèse de doctorat, Université des Sciences et Technologies de Lille, France, Mai 1999.
- [Berge 70]** Berge C., *Graphe et hyper-graphes*. Paris : Dunod, 1970.
- [Boehm 88]** Boehm B.W., A spiral model of Software Development and enhancement. *Computer*, May 1988.
- [Boissier 94]** Boissier O., Demazeau Y., An Architecture for Social and Individual Control and its Application to Computer Vision. *Proceedings of the Modelling Autonomous Agents in a Multi-Agent World conference*, (2-4 August 1994 ; Odense, Denmark), MAAMAW.
- [Bonabeau 99]** Bonabeau E., Dorigo E., Theraulaz G., L'intelligence en essaim. Ingénierie des Systèmes Multi-Agent (Actes des Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents), (Novembre 1999 ; Saint Gilles), JFIADSMA'99, Ed. par Marie-Pierre Gleizes, Paris : Hermes, 1999.
- [Booch 94]** Booch G., *Conception orientée objets et applications. 2ème édition*. USA : Addison-Wesley, 1994.
- [Borusan 93]** Borusan A., Colored Petri net based modeling of FMS. *Proceedings of IEEE Internationale Conference on Systems, Man, and Cybernetics, Vol 1*, (17-20 Octobre 1993 : Le Touquet), SMC'93, pp. 54-59.
- [Bourjot 99]** Bourjot C., Chevrier V., Bernard A., Krafft B., Coordination par le biais de l'environnement, une approche biologique. *Ingénierie des Systèmes Multi-Agent (Actes des Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents)*, (Novembre 1999 ; Saint Gilles), JFIADSMA'99, Ed. par Marie-Pierre Gleizes, Paris : Hermes, 1999.
- [Boy 91]** Boy G., *Intelligent Assistant Systems*. New York: Academic Press, 1991.
- [Briffaut 98]** Briffaut X., Guichard N., Pierre D., MERCURE : une plate-forme multi-agents cognitifs communiquant par KQML, appliquée à la communication d'entreprise. *Proceedings of the complex systems, intelligent systems & interfaces conference*, (25-27 Mai 1998 ; Nîmes), NÎMES'98.
- [Brooks 91]** Brooks R. A., Intelligence Without Reason. *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, (Août 1991 ; Sydney, Australie), IJCAI'91, Ed. by Myopoulos J.& Reiter R., pp. 569-595.
- [Broudoux 00]** Broudoux L., Payen L., Réalisation d'un Système Orienté Holon pour l'aide Au Travail Coopératif. Rapport de projet IUP G.E.I.I. Automatisation Intégrée et Systèmes Homme-Machine. Université de Valenciennes et du Hainaut-Cambrésis, Juin 2000.

-
- [Carrère 96] Carrère P., Production, gestion et diffusion électronique de documents, Ré-ingénierie des processus d'activités d'un département de Propriété Industrielle, Rapport de DEA, Université Technologique de Compiègne & SOLVAY S.A, 1996.
- [Chaib-Draa 92] Chaib-Draa B., Moulin B., Mandiau R., Millot P., Trends in Distributed Artificial Intelligence. *AI Review*, 1992, n°4, pp.35-66.
- [Chaib-draa 96] Chaib-draa B., Interaction between agents in routine, familiar and unfamiliar situations. *International Journal of Intelligent and Cooperative Information Systems*, 1996, 1, 5, pp. 7-20.
- [Chang 97] Chang E. J., Dillon T. S., Les méthodes du génie logiciel pour la conception des interfaces utilisateurs, un panorama critique d'approche. *Génie Logiciel*, n°43, 1997, pp. 2-23.
- [Coad 91] Coad P., Yourdon E., *Object-oriented design*. Prentice Hall, 1991.
- [Collinot 98] Collinot A., Drogoul A., Approche orientée agent pour la conception d'organisations: application à la robotique collective. *Revue d'intelligence artificielle*, 12, 1, 1998, pp. 125-147.
- [Constantinidis 98] Constantinidis Y., *Outils de construction du logiciel*. Paris: Hermes Science publications. 1998.
- [Coutaz 90] Coutaz J., *Interfaces homme-ordinateur : conception et réalisation*. Paris : Bordas, 1990.
- [De Michelis 94] De Michelis G., From the analysis of cooperation within work-processes to the design of CSCW Systems. *Proceedings of the 15th Interdisciplinary Workshop on Informatics and Psychology: Interdisciplinary approaches to system analysis and design*, (24-26 May 1994, Schaerding, Austria).
- [DeMarco 79] DeMarco T., *Structured analysis and system specification*. USA : Yourdon Press, 1979.
- [Demazeau 95] Demazeau Y., From Interactions To Collective Behaviour in Agent-Based Systems. *Proceedings of the first European Conference on Cognitive Science*, Saint-Malo, France, Avril 1995.
- [Diaper 89] Diaper D., *Task analysis for human-computer interaction*. Chichester, Angleterre : Ellis Horwood Limited, 1989.
- [Dignum 99] Dignum F., Are Information Agent just an extension of information systems or a new paradigme ? *Proceedings of Autonomous Agents Conference*, (1-5 May 1999 ; Seattle, USA), Agents'99.
- [Dumas 90] Dumas P., Charbonnel G., *La méthode OSSAD, pour maîtriser les technologies de l'information. Tome 1 : principes*. Paris : Les éditions d'organisation, 1990.
- [Ermine 95] Ermine J-L., MKSM, méthode de gestion des connaissances. Rapport de recherche CEA DIST/SMTI, France, 1995.

-
- [**Etzioni 94**] Etzioni O., Weld D., An Softbot-based interface to the Internet. *Communication of the ACM*, 1994, 37, 7, pp. 72-76.
- [**Ferber 95**] Ferber J., Les systèmes multi-agents, vers une intelligence collective. IIA, Paris : InterEditions, 1995
- [**Ferber 97**] Ferber J., Gutknecht O., Aladin: a meta-model for the analysis and design of organization in multi-agent systems. Rapport de recherche LIRMM 97189, Laboratoire d'Informatique, de Mécanique et de Micro-électronique de Montpellier, Décembre 1997.
- [**Finin 94**] Finin T., McKay R.F.D., McEntire R., KQML as an Agent Communication Language, *Proceedings of the Third International Conference on Information and Knowledge Management*, (Novembre 1994 ; Gaithersburg, USA), CIKM'94, ACM Press, 1994.
- [**Flores-Mendez 99**] Flores-Mendez R.A., Towards the Standardization of Multi-Agent System Architectures: An Overview. In ACM Crossroads. *Special Issue on Intelligent Agents*, Association for Computer Machinery, 5.4, pp.18-24, 1999.
- [**Gasser 89**] Gasser L., Research Notes in Artificial Intelligence. *Distributed Artificial Intelligence, volume I.* Ed. by Huns. Londres : Pitman, 1989.
- [**Gasser 92**] Gasser L., An overview of DAI. *Distributed Artificial Intelligence: Theory and Praxis*. Ed. by Avouris & Gasser, pp. 9-30. Dordrecht, Netherlands : Kluwer Academic Publishers, 1992.
- [**Gerber 99**] Gerber C., Siekmann J., Vierke G., Holonic Multi-Agent Systems. *Rapport de Recherche RR-99-03*, DFKI (Deutsche Forschungszentrum für Künstliche Intelligenz) GmbH, Mars 1999.
- [**Goldberg 84**] Goldberg A., Smalltalk-80, the interactive programming environment. Addison-Wesley, 1984.
- [**Goodman 86**] Goodman GO, Abel M.J., Collaboration research in SCL. *Proceedings of the Conference on Computer-Supported Cooperative Work*, (3-5 Décembre 1986 ; Austin, USA), CSCW'86, Ed. by Peterson D.
- [**Gracanion 94**] Gracanion D., Srinivasan P., Valavanis K.P., *Parameterized Petri nets and their application to planning and coordination in intelligent systems*. IEEE Transactions on Systems, Man and Cybernetics, 24, pp. 1483-1497, 1994.
- [**Gransac 97**] Gransac J., Réalisation d'un Simulateur de Réseaux de Petri orientés Objets, Contribution à une méthode d'analyse de systèmes administratifs complexes, Rapport de DESS Technologie et Méthodes de l'Aide à la Décision, Université de Valenciennes et du Hainaut-Cambrésis, 1997.
- [**Grislin 96**] Grislin M. E., Kolski C., Evaluation des interfaces homme-machine lors du développement de système interactif. *Technique et Science Informatiques (TSI)*, 15, 3, 1996, pp. 265-296.

-
- [Grudin 94] Grudin J., Eight Challenges for Developers. *Communications of the ACM*, 37, 1, 1994, pp. 93-105.
- [Handl 99] Handl D., Hoffmann H-J., Workflow agents in the document-centred communication in MALL2000 systems. *Proceedings of Autonomous Agents Conference*, (1-5 Mai 1999 ; Seattle, USA), Agents'99.
- [Helander 97] *Handbook of Human-Computer Interaction*. Ed. by Helander M., Landauer T.K., Prabhu P., Netherlands : Elsevier Science Publishers, 1997.
- [Henderson-Sellers 99] Henderson-Sellers B., Firesmith D.G., Comparing OPEN and UML: the two third generation OO development approaches. *Information and software technology*, 41, 1999, pp. 139-156.
- [Hofstadter 85] Hofstadter D., *Gödel, Escher, Bach, Les brins d'une guirlande éternelle*. Paris : InterEdition, 1985.
- [IGL 89] I.G.L. Technology, *SADT, un langage pour communiquer*. Paris : Eyrolles, 1989.
- [Iivari 98] Iivari J., Maansaari J., The usage of systems development methods: are we stuck to old practices ? *Information and software technology*, 40, 1998, pp. 501-510.
- [Ito 98] Ito N., Nakagawa K., Hotta T., Du X., Ishii N., EAMMO : an environmental agent model for multiple objects. *Information and software technology*, 40, 1998, pp. 397-404.
- [Jackson 95] Jackson M., *Software Requirements & Specifications : A Lexicon of Practice, Principles and Prejudices*. USA : Addison-Wesley, ACM Press Books, 1995.
- [Jacobson 93] Jacobson I., Christenson M., Jonsson P., Overgaard G., *Le génie logiciel orienté objet, une approche fondée sur les cas d'utilisation*. USA : Addison-Wesley, ACM Press Books, 1993.
- [Jambon 96] Jambon F., Erreurs et interruptions du point de vue de l'ingénierie de l'interaction homme-machine, Thèse de doctorat, Université de Grenoble, France, Décembre 1996.
- [Jaulent 90] *Génie Logiciel : les méthodes*. Paris : Armand Colin Editeur, 1990.
- [Kitano 98] Kitano H., Asada M., Kuniyoshi Y., Noda I., Osawa E., Matsubara H., RoboCup: A Challenge Problem for AI and Robotics. *RoboCup-97: Robot Soccer World Cup I*, Ed. Springer-Verlag, pp.1-19, 1998.
- [Koda 96] Koda T., Agents with Faces: A Study on the Effects of Personification of Software Agents. M.S. Thesis, Massachusetts Institute of Technology, 1996.
- [Koestler 69] Koestler A., *The Ghost in the Machine*. London : Arkana Books, 1969.
- [Kolski 97] Kolski C., *Interfaces homme-Machine, application aux systèmes industriels complexes (2^{ème} édition revue et étendue)*. Paris : Editions Hermes, 1997.

-
- [**Kolski 98**] Kolski C., A "call for answers" around the proposition of an HCI-enriched model. *ACM SIGSOFT Software Engineering Notes*, 23, 3, 1998, pp. 93-96.
- [**Kolski 98b**] Kolski C., Le Strugeon E., A review of "intelligent" human-machine interfaces in the light of the ARCH Model. *International Journal of Human-Computer Interaction*, 10, 3, 1998, pp. 193-231.
- [**Kuflik 00**] Kuflik T., Shoval P., User Profile Generation for Intelligent Information Agents - Research in Progress. *Proceedings of the Conference on Advanced Information Systems Engineering*, (5-6 juin 2000 ; Stockolm, Suède), CaiSE'00.
- [**Laird 97**] Laird J. E., Pearson D. J., Huffman S. B., Knowledge-directed adaptation in multi-level agents. *Journal of Intelligent Information systems*. 9, 3, pp. 261-275, 1997.
- [**Lant 94**] Lant T. K., Computer Simulations of Organizations as Experiential Learning Systems: Implications for Organization Theory. In *Computational Organization Theory*. Ed. by Carley K. M. & Prietula M.J., London : LEA publishers, pp. 195-215, 1994.
- [**Lashkari 94**] Lashkari Y., Metral M., Maes P., Collaborative Interface Agents. *Proceedings of the Twelfth National Conference on Artificial Intelligence*. (1-4 August 1994 ; Seattle, USA), AAAI '94, AAAI Press.
- [**Laurel 97**] Laurel B., Interface Agents : Metaphors with Character. *Software Agent*, Ed. by J. M. Bradshaw, AAAI Press, 1997.
- [**Le Moigne 77**] Le Moigne J.L., *La théorie du système général - théorie de la modélisation*. Vendôme : Presses Universitaires de France, 1977.
- [**Le Strugeon 95**] Le Strugeon E., Une méthodologie d'auto-adaptation d'un système multi-agents cognitifs. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, France, 17 janvier 1995.
- [**Lieberman 97**] Lieberman H., Autonomous Interface Agents. *Proceedings of the Conference on Human Factors in Computing Systems*, (22-27 Mars 1997 ; Atlanta, USA), CHI'97, Ed. by Pemberton S., New-York, USA : ACM Press, pp. 67-74.
- [**Lim 94**] Lim K.Y., Long J.B., The MUSE method for usability engineering. *Cambridge series on Human-Computer Interaction*, Cambridge, USA : Cambridge University Press, 1994.
- [**Mandiau 93**] Mandiau R., Contribution à la modélisation des univers multi-agents : génération d'un plan partagé. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, 22 février 1993.
- [**Mandiau 99**] Mandiau R., Le Strugeon E., Agimont G., Study of the influence of organizational structure on the efficiency of a multi-agent system. *Networking and Information Systems Journal*, 2, 1999, pp.153-179.
- [**Marshak 95**] Marshak R. T., Workflow : Applying Automation to Group Processes. *Groupware, Technology and Applications*, Ed. by Coleman D. & Khanna R., Prentice Hall, pp. 71-95, 1995.

- [**Medina-Mora 92**] Medina-Mora R., Winograd T., Flores R., Fores F., The action Workflow Approach to Workflow Management Technology. *Proceedings of the Computer Supported Cooperative Work conference (CSCW'92)*, (31 October - 4 November 1992 ; Toronto, Canada), USA : ACM Press, pp. 281-288, 1992.
- [**Minar 98**] Minar N., Kramer K., Maes P., Cooperating Mobile Agents for Dynamic Network Routing. In *Software Agents for Future Communications Systems*, Ed. by Hayzelden A., Bigham J., Londres : Springer-Verlag, 1998.
- [**Montmollin 86**] De Montmollin M., *L'intelligence de la tâche, éléments d'ergonomie cognitive*. Berne : Peter Lang.
- [**Morejon 94**] Morejon J., *MERISE, vers une modélisation orientée objet*. Paris : Les Editions d'Organisation, 1994.
- [**Muller 97**] Muller P.A., *Modélisation objet avec UML*. Paris : Eyrolles, 1997.
- [**Nielsen 93**] Nielsen J., *Usability engineering*. Academic Press, 1993.
- [**Notte 96**] Notte D., Le Département de Propriété Industrielle : Etude du domaine, rapport interne SOLVAY S.A.& Ergodin, 1996.
- [**Nurcan 96**] Nurcan S., Analyse et conception de systèmes d'information coopératifs. *Technique et Science Informatiques*, 15, 9, 1996, pp. 1287-1315.
- [**Nwana 96**] Nwana Hyacin S., Software Agent : an overview. *Knowledge Engineering Review*, 11, 3, 1996, pp. 205-244.
- [**O'Brien 98**] O'Brien P.D., Wiegand M.E., Agent based process management : applying intelligent agents to workflow. *Knowledge Engineering Review*, 13, 2, 1998.
- [**Occello 98**] Occello M., Demazeau Y., Baeijs C., Designing Organized Agents for Cooperation with real Time Constraints. *Proceedings of the first international workshop on collective robotics*, (4-5 July 1998 ; Paris), CRW'98, ed. by Drogoul A., Tambe M. & Fukuda T., New York, USA : Springer Verlag., pp. 25-37.
- [**Odgers 99**] Odgers B.R., Thompson S.G., Shepherdson J.W., Cui Z., Judge D.W., O'Brien P. D., Technologies for Intelligent Workflows: Experiences and Lessons. *Proceedings of Agent-Based Systems in the Business Context*, (18-22 juillet 1999 ; Orlando, Californie), AAAI 1999, AAAI Press, pp. 63-67.
- [**Odubiyi 97**] Odubiyi J.B., Kocur D.J., Weinstein S. M., Wakim N., Srivastava S., Gokey C., Graham J., SAIRE - A scalable agent-based information retrieval engine. *Proceedings of the Autonomous Agent conference*, (5-8 Février 1997 ;), Agents'97, Ed. by Müller J., New York : ACM Press, p. 292-299.
- [**Paide 99**] Paide R. F., When are methods complementary. *Information and software technology*, 41, 1999, pp. 157-162.
- [**Palanque 95**] Palanque P., Bastide R., Spécifications formelles pour l'ingénierie des interfaces homme-machine. *Technique et Science Informatiques*, 14, 4, 1995, pp. 473-500.

-
- [Pascot 93]** Pascot D., Bernadas C., L'Essence des Méthodes : Etude Comparative de Six Méthodes de Conception de Systèmes d'Information Informatisés. *Actes du congrès INFORSID'93 «Systèmes d'information, systèmes à base de connaissances»*, Lille, 11-14 Mai 1993.
- [Perrin 97]** Perrin P., La guerre des standards des méthodes orientées objet n'aura pas lieu. *Ingénierie des Systèmes d'Information*, 5, 5, 1997, pp. 533-552.
- [Quintero 98]** Quintero Carillo Ramos A. C., Ucros M., Multi-Agent System Coordination Model applied to the administrative Process of Systems Auditory in an Organization. *Proceedings of the 4th International Conference on information Systems Analysis and Synthesis*, (12-16 July 1998 ; Orlando, USA), ISAS'98.
- [Rhodes 00]** Rhodes J., Just-In-Time Information Retrieval. Ph.D. Thesis, MIT Media Lab, May 2000.
- [Rich 83]** Rich E., *Artificial Intelligence*. New York : Mac Graw Hill, 1983.
- [Rich 97]** Rich C., Candace L., Sidner (Lotus Development Corporation) *Proceedings of the ninth ACM Symposium on User Interface Software and Technology*, (November 1997 ; Seattle, USA), pp. 21-30.
- [Ross 77]** Ross D.T., Structured analysis (SA) : a language for communicating ideas. *IEEE Transactions on Software Engineering*, SE-3, 1, Janvier 1997.
- [Rumbaugh 91]** Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorenzen W., *Object-oriented modeling and design*. Prentice Hall, 1991.
- [Rumbaugh 98]** Rumbaugh J., Jacobson I., Booch G., *Unified Modeling Reference Manual*. Addison Wesley, 1998.
- [Rus 97]** Rus D., Gray R., Kotz D., Transportable Information Agent. *Journal of Intelligent Information systems*. 9, 3, pp. 215-238, 1997.
- [Schael 97]** Schael T., Théorie et pratique du workflow. Des processus métiers renouvelés. New-York : Springer Edition, 1997.
- [Schneider-Hufschmidt 93]** *Adaptive User Interfaces*. Ed. by Schneider-Hufschmidt M., Kühme T., Malinkowski U., Netherlands : Elsevier Science Publishers, 1993.
- [Simon 95]** Simon H. A., Explaining the ineffable : AI on the Topics of Intuition, Insight and Inspiration. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, (20-25 Auhust 1995 ; Montréal, Canada), IJCAI'95, Ed. by Dean T., Morgan Kaufman Publishers, pp. 939-948.
- [Soulié 98]** Soulié JC., Marsenac P., Calderoni S., Courdier R., GEAMAS V2.0 : an Object Oriented Platform for Complex systems Simulations. *Proceedings of Technology of Object-Oriented Languages and Systems conférence*, (28 Août 1998 ; Santa-Barbara, USA), TOOLS'98 , IEEE Computer Society.

- [**Tarby 93**] Tarby J.C., Gestion automatique du dialogue homme-machine à partir des spécifications conceptuelles. Thèse de doctorat, Université de Toulouse, septembre 1993.
- [**Tardieu 85**] Tardieu H., Rochfeld O., Colleti R., Panet G., Vahee G., *La méthode Merise, démarche et pratiques (tome 2)*. Editions d'Organisation : Paris, 1985.
- [**Tardieu 91**] Tardieu H., Rochfeld O., Colleti R., *La méthode Merise, principes et outils (tome 1), 2ème édition*. Paris : Editions d'Organisation, 1991.
- [**Thimbleby 94**] Thimbleby H., Marsh S., Jones S., Cockburn A., Trust in CSCW. *CSCW : The multimedia and networking paradigm*, UNICOM, ed. by Scrivener S.A.R., Aldershot, England: Avebury Technical, pp. 253-272, 1994.
- [**Tranvouez 99**] Tranvouez E., Espinasse B., Protocoles de coopération pour le réordonnement d'atelier. Ingénierie des Systèmes Multi-Agent (Actes des Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents), (Novembre 1999 ; Saint Gilles), JFIADSMA'99, Ed. par Marie-Pierre Gleizes, Paris : Hermes, 1999.
- [**Turing 50**] Turing A. M., Computing Machinery and Intelligence. *Journal of the Mind Association*. LIX, 236, pp. 433-460, Oxford , UK :Oxford University Press, 1950.
- [**Van Brussel 96**] Van Brussel H., Bongaerts L., Wyns J., Valckenaers P., Van Ginderachter T., A Conceptual Framework for Holonic Manufacturing Systems: Identification of Manufacturing Holons. *Journal of Manufacturing Systems*, 18, 1, 1999, pp. 35-52.
- [**van der Aalst 98**] van der Aalst W.M.P., The application of Petri nets to workflow management. *Journal of Circuits, Systems and Computers*, 8, 1, 1998, pp. 21-66.
- [**Van Mulken 99**] Van Mulken S., André E., Müller J., An empirical Study on the Trustworthiness of Life-Like Interface Agents. *Human Computer Interaction – Communication, Cooperation and Application Design, Vol 2*. Ed. by Bullinger H-J. & Ziegler J., Londres : LEA., pp. 152-156, 1999.
- [**Vergison 98**] Vergison E., Adam E., Problèmes de supervision dans les systèmes industriels complexes. *Actes des Journées Automatique et Homme du Club EEA*. (Septembre 1998 ; Valenciennes).
- [**Voorhoeve 96**] Voorhoeve M., van der Aalst W.M.P., Conservative Adaption of Workflow. *Proceedings of the International Conference on Practical Aspects of Knowledge Management, Workshop on Adaptive Workflow*, (October 1996 : Basel, Switzerland), *PAKM'96*, Ed. by Wolf M. & Reimer U., pp. 1-15, 1996.
- [**WFMC 99**] Terminology and Glossary. *Workflow Management Coalition*. Document numéro WFMC-TC-1011, 3.0, February 1999.
- [**White 97**] White J. E., Mobile Agents. In *Software Agent*. Ed. by J. M. Bradshaw, AAAI Press, Menlo Park, USA, 1997.

- [Wielinga 93]** Wielinga B., Van de Velde W., Schreiber G., Akkermans H., Towards an unification of knowledge modelling approaches. *Second generation expert systems*, Ed. by David J.M., Krivine J.P., Simmons R., pp. 299-335, Londres : Springer-Verlag, 1993.
- [Williges 87]** Williges R.C., Williges B.H., Elkerton J., Software Interface Design. Ed. by Salvendy G., *Handbook of Human factors*, pp. 1416-1449, New York, Wiley, 1987.
- [Wilson 90]** *Evaluation of human works :a practical ergonomics methodology*. Ed. by Wilson J., Corlett N., London, UK : Taylor and Francis, 1990.
- [Wilson 94]** Wilson P., Introducing CSCW, What it is and why we need it. *CSCW : The multimedia and networking paradigm*, UNICOM, ed. by S.A.R. Scrivener, Aldershot, England: Avebury Technical, pp. 1-18, 1994.
- [Wyns 99]** Wyns J., Reference architecture for holonic manufacturing systems - the key to support evolution and reconfiguration. Ph.D. Thesis, K.U. Leuven (Université Catholique de Louvain), Belgique 1999.
- [Young 95]** Young M. J., Human Error and the holon cognitive architecture. *Proceedings of the 6th Symposium on analysis, design and evaluation of Man-Machine systems*, Massachusetts Institute of Technology, Cambridge, USA : MIT-Press, 1995.
- [Yu 99]** Yu L., A Conceptual Framework For Agent Oriented and Role Based Workflow Modeling. *Proceedings of the Conference on Advanced Information Systems Engineering*, (14-15 june 1999; Heidelberg, Germany), CaiSE'99.

Index des figures et des tableaux

Index des figures et des tableaux

CHAPITRE I. MODÈLES D'ORGANISATIONS MULTI-AGENTS POUR LES SYSTÈMES D'AIDE AU TRAVAIL COOPÉRATIF EN ENTREPRISE

FIGURE I.1. UN EXEMPLE DE RÉSEAU COOPÉRATIF	18
FIGURE I.2. RÉSEAU REPRÉSENTANT L'ACTIVITÉ D'UN SYSTÈME ADMINISTRATIF	20
FIGURE I.3. REPRÉSENTATION D'UN SYSTÈME À L'AIDE DE PLUSIEURS RÉSEAUX	21
FIGURE I.4. MÉCANISME DE NÉGOCIATION	22
FIGURE I.5. EXEMPLE DE STRUCTURE FONCTIONNELLE	24
FIGURE I.6. EXEMPLE DE STRUCTURE DIVISIONNELLE	25
FIGURE I.7. EXEMPLE DE STRUCTURE MATRICIELLE	26
FIGURE I.8. POSITIONNEMENT D'UN AGENT DANS LE MONDE	30
FIGURE I.9. DEGRÉS D'AUTONOMIE, DE COOPÉRATION ET D'ADAPTATIVITÉ DES PRINCIPAUX AGENTS COGNITIFS	34
FIGURE I.10. OMA À STRUCTURE HIÉRARCHIQUE	36
FIGURE I.11. OMA À STRUCTURE DE MARCHÉ POURSUIVANT DEUX OBJECTIFS	36
FIGURE I.12. OMA À STRUCTURE DE COMMUNAUTÉ COMPOSÉE DE 5 AGENTS DE MÊME NIVEAU	37
FIGURE I.13. OMA À STRUCTURE DE SOCIÉTÉ COMPOSÉE DE 3 ENTITÉS	37
FIGURE I.14. EXEMPLES DE SYSTÈMES HOLONIQUES	41
FIGURE I.15. HIÉRARCHIES IMBRIQUÉES	43
FIGURE I.16. DIFFÉRENTES STRATÉGIES RÉPONDANT À LA RÈGLE "ALLER DE BRUXELLES À BIARRITZ"	44
FIGURE I.17. FILTRAGE DE L'INFORMATION DANS UN SYSTÈME HOLONIQUE	45
FIGURE I.18. AUTRE EXEMPLE DE FILTRAGE DE L'INFORMATION DANS UN SYSTÈME HOLONIQUE	46
FIGURE I.19. ORGANISATION HOLONIQUE SOUS FORME DE MULTI-HIÉRARCHIE	47
FIGURE I.20. EQUIVALENCE ENTRE DEUX ENVIRONNEMENTS MULTI-AGENT	49
TABLEAU I.1. GRILLE D'ANALYSE FONCTIONNELLE [FERBER 95]	31

CHAPITRE II. ETUDE COMPARATIVE DE METHODES DU GENIE LOGICIEL DANS UN BUT DE MODELISATION DES ORGANISATIONS MULTI-AGENTS

FIGURE II-1. EXEMPLE DE PERCEPTION D'ORGANISATION PAR UNE MÉTHODE	64
FIGURE II.2. CONFRONTATION DES 7 MÉTHODES À L'AIDE DE DIAGRAMMES	84
TABLEAU II.1. ANALYSE DE LA METHODE MERISE SELON LES CINQ DIMENSIONS	74
TABLEAU II.2. ANALYSE DE LA METHODE OMT SELON LES CINQ DIMENSIONS	76
TABLEAU II.3. ANALYSE DE LA METHODE UML SELON LA DIMENSION REPRESENTATION	77
TABLEAU II.4. ANALYSE DE LA METHODE SADT SELON LES CINQ DIMENSIONS	78
TABLEAU II.5. ANALYSE DE LA METHODE OSSAD SELON LES CINQ DIMENSIONS	80
TABLEAU II.6. ANALYSE DE LA METHODE CISAD SELON LES DIMENSIONS REPRESENTATION ET COOPERATION	82
TABLEAU II.7. ANALYSE DE LA METHODE MKSM SELON LES CINQ DIMENSIONS	83

CHAPITRE III. MAMOSACO : UNE MÉTHODE ADAPTABLE DE MODÉLISATION DE SYSTÈMES ADMINISTRATIFS COMPLEXES

FIGURE III.1. MODÈLE ∇ POUR LE DÉVELOPPEMENT DE MODULES D'AIDE ADAPTÉ DE [KOLSKI 1997]	95
FIGURE III.2. EXEMPLE D'ANALYSE ORIENTÉE PROCESSUS	97
FIGURE III.3. EXEMPLE D'ANALYSE ORIENTÉE RÔLE	98
FIGURE III.4. EXEMPLE D'ANALYSE ORIENTÉE DOCUMENT	98
FIGURE III.5. EXEMPLE DE REPRÉSENTATION DE DOCUMENTS PAR LE MODÈLE DE DONNÉES DE UML	100
FIGURE III.6. EXTRAIT D'UN MODÈLE DE FLUX DE DONNÉES	101
FIGURE III.7. EXTRAIT D'UN MODÈLE DES TRAITEMENTS	102
FIGURE III.8. RÉSEAU DE PETRI OÙ UNE PLACE REPRÉSENTE UNE ACTIVITÉ	103
FIGURE III.9. RÉSEAU COLORÉ OÙ UN ACTEUR EST SYMBOLISÉ PAR UNE PLACE	104
FIGURE III.10. EXEMPLE DE WORKFLOW NET	105
FIGURE III.11. REPRÉSENTATION DÉTAILLÉE AU MAXIMUM	108

FIGURE III.12. REPRÉSENTATION MOYENNEMENT ABSTRAITE DU PROBLÈME.....	109
FIGURE III.13. REPRÉSENTATION DU PROBLÈME AU DEGRÉ D'ABSTRACTION MAXIMUM.....	109
FIGURE III.14. REPRÉSENTATION D'UN ESPACE DE TRAVAIL PAR RÉSEAU DE PETRI PARAMÉTRÉ.....	112
FIGURE III.15. EXEMPLE DE TRAITEMENT D'UNE INTERRUPTION.....	112
FIGURE III.16. EXTRAIT D'UN MODÈLE DYNAMIQUE.....	113
FIGURE III.17. EXEMPLE DE SIMPLIFICATION ORGANISATIONNELLE PAR AUGMENTATION DE L'AUTONOMIE	116
FIGURE III.18. EXEMPLE DE SIMPLIFICATION ORGANISATIONNELLE PAR LA COOPÉRATION.....	117
FIGURE III.19. ENCHAÎNEMENT DES PHASES DANS LA MÉTHODE MAMOSACO [ADAM 98b].....	117
FIGURE III.20. L'ARCHITECTURE D'UN SOHTCO.....	121
FIGURE III.21. L'ARCHITECTURE D'UN SOHTCO	122
FIGURE III.22. EXTRAIT D'UNE MODÉLISATION OBJET DU SOHTCO.....	128
FIGURE III.23. EXEMPLE DE TRANSMISSION D'UN DOCUMENT ASSISTÉ PAR LE SOHTCO.....	130
FIGURE III.24. MODÈLE ∇ ET PRINCIPE DE RÉUTILISABILITÉ POUR LE DÉVELOPPEMENT DE SOHTCO.....	132
TABLEAU III.1. DEFINITION D'UN RESEAU DE PETRI PARAMETRE (EXTRAIT DE [AGIMONT 96]).....	106
TABLEAU III.2. EXEMPLES DE REGLES D'ASSISTANCE.....	120
TABLEAU III.3. GRILLE DE CONCEPTION ADAPTEE DE LA GRILLE D'ANALYSE DE FERBER [FERBER 95].....	124
TABLEAU III.4. LA DEFINITION DE L'AGENT RESPONSABLE DE PROCEDURE.....	125
TABLEAU III.5. LA DEFINITION DE L'AGENT RESPONSABLE DE POSTE.....	125
TABLEAU III.6. LA DEFINITION DE L'AGENT EXECUTANT.....	126

CHAPITRE IV. CONCEPTION D'UN ATELIER DE GÉNIE LOGICIEL SUPPORTANT LA MÉTHODE MAMOSACO
--

FIGURE IV.1. LES COMPOSANTS PRINCIPAUX D'UN ATELIER DE CONCEPTION, INSPIRÉ DE [JAULENT 90].....	140
FIGURE IV.2. ARCHITECTURE GÉNÉRALE D'UN ATELIER DE DÉVELOPPEMENT.....	141
FIGURE IV.3. LES SIX PAGES DE L'ATELIER MAMOSACO.....	143
FIGURE IV.4. EXEMPLE DE TABLEAU DE DÉFINITION DE DIAGRAMME SOUS VISIO.....	144
FIGURE IV.5. COUCHE LOGICIELLE FACILITANT L'ÉDITION D'UNE DONNÉE DANS VISIO.....	145
FIGURE IV.6. EXEMPLE DE MODÈLE OBJET D'UN DOCUMENT.....	145
FIGURE IV.7. BOÎTE À OUTILS DE L'ATELIER DE MAMOSACO	146
FIGURE IV.8. EXEMPLE DE MATRICE ACTIVITÉS/RÔLES POUR UNE PROCÉDURE ADMINISTRATIVE COMPLEXE (CAS D'UNE PROCÉDURE DE DEMANDE DE BREVET).....	147
FIGURE IV.9. EXEMPLE DE REPRÉSENTATION DES DONNÉES PAR LE SYMBOLISME DE UML.....	148
FIGURE IV.10. OUTILS RELATIFS AU MODÈLE DES DONNÉES	149
FIGURE IV.11. OUTILS NÉCESSAIRES À LA MODÉLISATION DES ACTIVITÉS	149
FIGURE IV.12. EXEMPLE DE MODÈLE D'ACTIVITÉ	150
FIGURE IV.13. OUTILS DISPONIBLES POUR LA MODÉLISATION DES TRAITEMENTS	150
FIGURE IV.14. EXEMPLE DE MODÈLE DE TRAITEMENT (CAS D'UNE PROCÉDURE DE DÉPÔT DE BREVET)	151
FIGURE IV.15. EXEMPLE DE MODÈLE DYNAMIQUE (CAS D'UNE PROCÉDURE D'ÉTUDE DE BREVETABILITÉ).....	152
FIGURE IV.16. INTERFACE UTILISATEUR POUR L'IMPRESSION DES MODÈLES SOUS VISIO.....	153
FIGURE IV.17. INTERFACE SIMOH POUR UNE PROCÉDURE DE DÉPÔT DE BREVET.....	154
FIGURE IV.18. MODÈLE OBJET UML DE LA STRUCTURE INTERNE DU SIMULATEUR	155
FIGURE IV.19. DÉFINITION D'ACTEURS DANS SIMOH	157
FIGURE IV.20. FENÊTRE D'ÉDITION DES PLACES	157
FIGURE IV.21. EXEMPLE DE DÉFINITION DES TRANSITIONS DANS SIMOH	158
FIGURE IV.22. ÉDITION D'UNE RÈGLE DANS SIMOH.....	159

CHAPITRE V. APPLICATION DE LA MÉTHODE MAMOSACO À DES PROCESSUS ADMINISTRATIFS COMPLEXES ET PERSPECTIVES DE RECHERCHE

FIGURE V.1. LES PHASES DE LA MÉTHODE MAMOSACO APPLIQUÉE AU CAS DE LA PROCÉDURE DE DÉPÔT DE BREVET.	167
FIGURE V.2. EXTRAIT DE L'ANALYSE ORIENTÉE PROCÉDURE DU DÉPÔT DE BREVET AU SEIN DU DÉPARTEMENT DE PROPRIÉTÉ INDUSTRIELLE	168
FIGURE V.3. MATRICE ACTIVITÉS-RÔLES DE LA PROCÉDURE SIMPLIFIÉE DE DÉPÔT DE BREVET.....	170
FIGURE V.4. FENÊTRE D'ÉDITION	170
FIGURE V.5. MODÈLE DES DONNÉES DE LA PROCÉDURE SIMPLIFIÉE DE DÉPÔT DE BREVET	171
FIGURE V.6. MODÈLE DE FLUX DE DONNÉES DE LA PROCÉDURE DE DÉPÔT DE BREVET	173
FIGURE V.7. MODÈLE DE TRAITEMENTS DE LA PROCÉDURE SIMPLIFIÉE DE DÉPÔT DE BREVET	175
FIGURE V.8. EXTRAIT DU MODÈLE DYNAMIQUE DE LA PROCÉDURE SIMPLIFIÉE DE DÉPÔT DE BREVET	177
FIGURE V.9. MODÈLE DE TRAITEMENTS DE LA PROCÉDURE SIMPLIFIÉE DE DÉPÔT DE BREVET	178
FIGURE V.10. COPIE D'ÉCRAN DE LA SIMULATION DE LA PROCÉDURE DÉPÔT DE BREVET.....	180
FIGURE V.11. EXTRAIT DU FICHIER TEXTE DÉFINISSANT LA PROCÉDURE ET SON FONCTIONNEMENT....	181
FIGURE V.12. EXTRAIT DE L'ARCHITECTURE DU SOHTCO DÉDIÉ À LA PROCÉDURE DE DÉPÔT DE BREVET	183
FIGURE V.13. SPÉCIFICATION DE LA PROCÉDURE D'INITIALISATION DU SOHTCO (1/2).....	184
FIGURE V.14. SPÉCIFICATION DE LA PROCÉDURE D'INITIALISATION DU SOHTCO (2/2).....	185
FIGURE V.15. EXEMPLE DE COMMUNICATION KQML DANS LE SOHTCO.....	186
FIGURE V.16. EXEMPLE DES PREMIERS MODÈLES UTILISÉS (CAS D'UNE PROCÉDURE DE GESTION DE COURRIER ENTRANT).....	187
FIGURE V.17. APPORTS À LA BASE RÉUTILISABLE DU MODÈLE NABLA	190
FIGURE V.18. DIAGRAMMES PERT ENVISAGÉS POUR LA REPRÉSENTATION D'UNE ACTIVITÉ.....	191