



HAL
open science

Conception générique d'un outil de configuration de “ e-TP ”

Saher Arnous

► **To cite this version:**

Saher Arnous. Conception générique d'un outil de configuration de “ e-TP ”. Environnements Informatiques pour l'Apprentissage Humain. INSA de Lyon, 2014. Français. NNT : 2014ISAL0093 . tel-01153218

HAL Id: tel-01153218

<https://theses.hal.science/tel-01153218>

Submitted on 19 May 2015

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.

Thèse
**Conception générique d'un outil de configuration de
« e-TP »**

Présentée devant
L'Institut National des Sciences Appliquées de Lyon (INSA)

Pour obtenir
Le grade de docteur

Ecole Doctorale
Informatique et Mathématiques (InfoMaths)

Par
Saher ARNOUS

Jury :

M. Jean-Pierre PECUCHET : Professeur à l'INSA de Rouen, Rapporteur
M. Pascal LEROUX : Professeur à l'université du Maine, Rapporteur
M. Patrick PREVOT : Professeur à l'INSA de Lyon
M. Arnaud LELEVE : Maître de conférences à l'INSA de Lyon
M. Jean-Yves RAMEL : Professeur à l'Ecole Polytechnique de l'Université de TOURS
M. Khalid KOUISS : Maître de conférences à l'IFMA de Clermont-Ferrand

Laboratoire de recherche : Laboratoire d'InfoRmatique en Image et Systèmes d'information
(LIRIS) - UMR 5205

Résumé

Renforcés par les avancées technologiques des « Sciences et Technologies de l'Information et de la Communication (STIC) », les Travaux Pratiques électroniques « e-TP » sont devenus un mode d'enseignement incontournable surtout dans les disciplines techniques et scientifiques. Plusieurs modes d'e-TP ont émergé : TP virtuel, TéléTP, TP présentiel assisté par ordinateur, TP en réalité augmentée... ces derniers modes nécessitant l'usage de dispositifs matériels (maquette pédagogique, appareils de mesure, robots, etc.). Dans la majorité des cas, avant ou même pendant une session d'e-TP, il est nécessaire de (re)configurer ces dispositifs selon des besoins pédagogiques. Cette reconfiguration nécessite, pour des systèmes complexes, comme les Systèmes Automatisés de Production (SAP), des compétences que l'instructeur ne possède pas systématiquement. Ce qui impose la présence d'un technicien ou limite le nombre d'instructeurs susceptibles d'utiliser la plate-forme pédagogique.

Ce travail de recherche a pour objectif de faciliter la (re)configuration de systèmes complexes, particulièrement les SAP, dans le cadre d'e-TPs.

Une première enquête auprès des utilisateurs de l'« AIP-Priméca-RAO » (situé à l'INSA de Lyon) a révélé les besoins et contraintes liés à une plate-forme de ce type. Il s'est avéré qu'au-delà de la (re)configuration, l'absence d'outil commun de gestion des ressources pédagogiques faisait perdre un temps précieux aux utilisateurs. Ce constat a nourri la conception d'un outil informatique gérant une chaîne éditoriale dont le but est de simplifier la création, l'édition, l'assemblage, l'organisation et la réutilisation des différentes ressources à exploiter dans une session d'e-TP. Cet outil a également pour objectif d'améliorer l'autonomie de l'instructeur lors de la préparation de sessions d'e-TP tout en réduisant le temps requis pour configurer cette session. Cela a impliqué d'automatiser le processus de reconfiguration du SAP support d'e-TP, et de publication des scénarios pédagogiques sur un système de gestion d'apprentissage « LMS » (*Learning Management System*).

Un prototype a été développé et testé sur des e-TPs réels afin de valider cette conception.

Cet outil pourrait, ultérieurement, être rendu plus générique afin de servir des e-TPs dans d'autres disciplines.

Abstract

Powered by the technological advances of the “Information and communication sciences and technologies”, the Electronic Laboratory for Practical Training “ELab” (also known as ELab hands-on training) has become an inescapable teaching mode especially in the technical and scientific disciplines. Thus, several ELab modes have emerged: virtual ELab, remote ELab, Local Elab, augmented reality ELab, etc. The latter require the use of hardware devices (educational mock ups, measuring instruments, robots, etc.). In most cases, before or during an ELab session, these devices need to be reconfigured according to teaching purposes. For complex systems, like Automated Production Systems (APS), this reconfiguration process requires technical skills which the instructor does not have systematically. This imposes that a technician should be available, failing which the usage of the pedagogical platform will be limited to a few skilled instructors.

Accordingly, this research aims at facilitating the reconfiguration process of complex systems (particularly the APS) featuring ELabs.

A first survey designated to the users of « AIP-Priméca-RAO », located at the *INSA de Lyon*, has revealed the needs and constraints related to such a platform. It has been highlighted that beyond the (re)configuration process, a waste of precious time was detected. It has been established that it was due to the absence of a common tool for pedagogical resource management. This observation fed the design of a software tool managing an editorial chain aiming at simplifying creation, edition, assembling, organization, and the reutilization of different resources that can be exploited in an ELab session. This tool is also intended to improve the autonomy of the instructor during the preparation of an ELab session, by reducing the required time to configure this session. This implies to automate the reconfiguration process of an APS supporting the ELab and the publishing of the pedagogical learning scenarios on a Learning Management System (LMS).

In order to validate this design, a prototype has been developed and tested on real Elab cases.

Subsequently, this tool could be rendered more generic so that it can serve Elabs in different disciplines.

Table des matières

Résumé.....	2
Abstract	3
Table des matières.....	4
Table des figures	8
Table des tableaux.....	11
Introduction générale.....	12
1. Contexte général : l'enseignement supérieur et les avancées technologiques	17
Introduction	17
1.1 Émergence des STIC et leur intégration dans les universités	18
1.2 La formation ouverte et à distance	22
1.3 La pédagogie du « e-Learning ».....	23
1.4 Variétés pédagogiques et situationnelles du e-learning	25
1.5 Les outils du e-learning	27
1.5.1 Les outils	28
1.5.2 Les ressources et composantes pédagogiques	31
1.6 Effets des nouvelles technologies sur les modes d'enseignement	33
1.6.1 Les modes d'enseignement.....	35
1.6.1.1 Le cours	35
1.6.1.2 Les Travaux Dirigés (TD)	36
1.6.1.3 Les Travaux Pratiques (TP).....	36
1.6.1.4 Le projet	36
1.6.1.5 Le stage	37
1.6.2 Impact des TIC sur l'évolution des modes d'enseignement.....	38
1.6.2.1 Évolution des cours	39
1.6.2.2 Les TD.....	40
1.6.2.3 Les TP	42
1.6.2.4 Les projets	44
1.6.2.5 Les stages	46
1.6.2.6 Emergence d'un nouveau mode : les Learning Games	47
1.7 Conclusion.....	50
2. Contexte de la recherche : les Travaux Pratiques.....	52
Introduction	52
2.1 Objectifs généraux des TP.....	52
2.2 Les TP dans l'enseignement supérieur	55
2.3 Les acteurs principaux dans un environnement de TP.....	57
2.4 Comment intégrer les TP dans une démarche e-learning globale	58
2.5 Evolution de l'environnement TP vers e-TP	60
2.5.1 Plate-forme de TP virtuels (VLab).....	61
2.5.2 Plate-forme de téléTP (RLab)	63
2.5.3 Plate-forme de TP hybrides (HLab)	65
2.5.4 Synthèse sur l'e-TP	67
2.5.5 Les TP dans la plate-forme de TP Electronique (ELab)	67
2.6 Notre terrain d'expérimentation	69
2.6.1 L'AIP-Primeca-RAO	69
2.6.2 Le transfert libre	70
2.6.2.1 Exemple d'utilisation : le projet MES.....	72
2.7 Problématiques liées à la plate-forme « transfert libre »	73
2.7.1 Problématique 1 : la variété des modes de fonctionnement.....	73
2.7.2 Problématique 2 : la sécurité (matérielle, logicielle et humaine).....	74

2.7.3	Problématique 3 : la chaîne éditoriale	76
2.7.3.1	Gestion des ressources pédagogiques	76
2.7.3.2	Les différents intervenants de TP	77
2.7.3.3	Les façons d'utiliser un SAP dans un TP	78
2.7.3.4	Interdisciplinarité	79
2.7.4	Problématique 4 : de la reconfiguration d'un SAP	80
2.7.5	Besoins identifiés en automatique	81
2.8	Objectifs de ce travail de recherche	84
3.	Le cadre de modélisation du système	87
	Introduction	87
3.1	Travaux à l'origine de notre recherche	87
3.2	Choix méthodologiques et techniques lors de la conception	89
3.3	Modélisation et conception du système	89
3.3.1	Modélisation des exigences : modèle CIM	90
3.3.1.1	Synthèse des besoins exprimés par les utilisateurs directs :	90
3.3.1.2	Analyse de la situation existante en fonction des besoins	91
3.3.1.3	Extraction des exigences du système	95
3.3.1.4	Notre engagement de réalisation	100
3.3.1.5	Cas d'utilisation types	101
3.3.1.6	Le configurateur dans son environnement	103
3.3.1.7	Les données du configurateur en interface avec l'environnement	104
3.3.1.8	Objectifs qualité du système	106
3.3.1.9	Bilan sur le modèle CIM	106
3.3.2	Définition structurelle du système : modèle PIM	107
3.3.2.1	Types de données internes	107
3.3.2.2	Généricité de configuration et réutilisation des ressources	108
3.3.2.3	Flux de données internes du configurateur	110
3.3.2.4	Architecture du configurateur	113
3.3.2.5	Structure de la bibliothèque de ressources pédagogiques	115
3.3.2.5.1	Entrées/Sorties et interconnexion des éléments de la bibliothèque	116
3.3.2.6	Structure du générateur de configuration	118
3.3.2.7	Structure de la bibliothèque de configurations	122
3.3.2.8	Structure de l'installateur de configuration	126
3.4	Vérification de conformité du modèle en fonction de la spécification	130
3.5	Conclusion sur la modélisation	133
4.	Implémentation et validation expérimentale	134
	Introduction	134
4.1	Plate-forme d'implémentation	134
4.2	Réalisation d'un prototype	137
4.2.1	Fonctions principales	137
4.2.2	La structuration d'une configuration	138
4.2.3	Catégorisation et indexation des ressources	140
4.2.3.1	Perspective « composant »	141
4.2.3.2	Perspective « système réel »	141
4.2.3.3	Perspective des objectifs pédagogiques	141
4.2.4	Réutilisabilité des ressources	143
4.3	Utilisation du configurateur	144
4.3.1	Préparation des ressources	144
4.3.2	Phase de génération de configuration	148
4.3.3	Phase d'exploitation	150

4.4	Application sur des environnements de TP en automatique	151
4.4.1	Cas du « Magasin Vertical »	151
4.4.1.1	Description du magasin vertical	152
4.4.1.2	Expérimentation	154
4.4.1.3	La préparation des ressources.....	154
4.4.1.4	Génération d'une configuration	159
4.4.1.5	Installation et utilisation d'une configuration	160
4.4.1.6	Gains et limites.....	161
4.4.2	Cas du « Transfert libre »	162
4.4.2.1	Description du transfert libre.....	162
4.4.2.2	Objectifs pédagogiques du projet MES et son déroulement	165
4.4.2.3	Scénario pédagogique témoin: production autonome d'assemblages	165
4.4.2.3.1	Besoins pour le déroulement d'une session de projet	166
4.4.2.3.2	Conception/réalisation des ressources.....	167
4.4.2.3.3	La préparation du TP.....	168
4.4.2.4	Utilisation du configurateur lors de la phase de préparation.....	168
4.4.2.5	Utilisation du configurateur pour générer une configuration spécifique	171
4.4.2.6	Utilisation du configurateur lors de la phase d'exploitation	172
4.4.2.7	Gains et limites.....	173
4.4.3	Réutilisation de configuration du « Transfert libre » dans un autre contexte que le projet MES : TP S.A.P. en GMC	174
4.4.3.1	Scénario pédagogique : gestion d'un poste de travail	174
4.4.3.2	Utilisation du configurateur dans la phase de préparation	175
4.4.3.3	Utilisation du configurateur pour générer la configuration spécifique	176
4.4.3.4	Utilisation du configurateur dans la phase d'exploitation.....	176
4.4.3.5	Évaluation.....	177
4.4.4	Synthèse	177
4.5	Application sur d'autres environnements de TP : en informatique.....	179
4.5.1	Sujet du TP.....	179
4.5.2	Préparation des ressources pour la première session	181
4.5.3	Génération de configuration et son exploitation dans la session de TP	182
4.5.4	Deuxième session	182
4.5.5	Évaluation des retours des utilisateurs	184
4.6	Synthèse sur l'utilisation du configurateur.....	185
4.6.1	Analyse technique	185
4.6.2	Analyse des objectifs remplis.....	187
4.6.3	Limites et points à améliorer	191
4.7	Conclusion.....	192
	Conclusion générale	193
	Annexe A : L'approche objet.....	198
	1. Approche orientée objet en informatique	199
	2. Application dans le domaine de l'automatisme.....	200
	2.1. POO dans les environnements de programmation en automatisme	201
	2.2. Langages POO en informatique au service de l'automatisme	204
	Annexe B : Langages et méthodologies de conception orientée objet.....	207
	1. L'arrivée d'un langage de modélisation unifié (UML)	208
	2. Le langage de modélisation des systèmes SysML	209
	3. Méthodologies de Conception des systèmes	212
	3.1. Le processus d'ingénierie système	213
	3.1.1. Exemples de processus d'ingénierie système.....	217

- SIMILAR [Bahill-1998].....	217
- ESDL [Berger-2002].....	217
- HARMONY [Estefan-2008].....	218
3.2. L'approche de modélisation MDA.....	218
Annexe C : Sujet sur le magasin vertical	221
A) Utilisation du clavier.....	226
B) Mémorisation des informations.....	226
C) Le grafcet.....	226
Annexe D : TP sur le Transfert libre en GMC	228
Annexe E : Questionnaire sur les retours d'utilisation du configurateur	238
Glossaire.....	240
Références	243

Table des figures

Figure 1.1 : Évolution des termes qualifiant l'utilisation des TIC dans l'éducation.....	19
Figure 1.2 : Modes de formation assistée par TIC.	22
Figure 1.3 : La typologie du « e-Learning ».	26
Figure 1.4 : Evolution des technologies dans le e-learning.....	28
Figure 1.5 : Le triangle pédagogique.	34
Figure 1.6 : Le triangle pédagogique avec le rôle central du concepteur.....	35
Figure 1.7 : Liste d'activités d'utilisateur affichée dans la vue de Time Machine de TrAVis « Tdelille ».	42
Figure 1.8 : Les 7 étapes de conception de Serious Games.	48
Figure 1.9 : Les 6 facettes du jeu sérieux.....	49
Figure 1.10 : Boucle « perception, cognition, action » en communication avec le monde virtuel.	50
Figure 1.11 : Le continuum réel / virtuel.	50
Figure 2.1 : Construction du Savoir-faire.....	54
Figure 2.2 : Catégorisation des connaissances dans l'éducation.	56
Figure 2.3 : Les différents acteurs et leurs interactions dans un TP en présentiel.	57
Figure 2.4 : Les plate-forme de TP virtuels.	62
Figure 2.5 : La plate-forme de téléTP ([Gravier-2007] p.41).	64
Figure 2.6 : La plate-forme de TP hybride.	66
Figure 2.7 : ELabs : TP assistés par des TIC.	68
Figure 2.8 : Structure du système de transfert libre.	70
Figure 2.9 : Pièces de différentes couleurs à poser sur une palette.....	71
Figure 2.10 : Structure de poste.....	72
Figure 2.11 : Enquête: domaines d'enseignement des contributeurs.....	82
Figure 2.12 : Enquête : temps pour configurer une session de TP.....	83
Figure 2.13 : Enquête : instructeurs ayant besoin de l'aide d'un technicien.	83
Figure 2.14 : Enquête : réutilisation et gestion de ressources.	84
Figure 2.15 : Enquête : archivage et gestion des configurations via une bibliothèque de configuration.	84
Figure 3.1: Analyse causale de la situation existante illustrant les freins qui motivent ces travaux.....	92
Figure 3.2 : Schéma comportemental du système en fonction des besoins des utilisateurs. ...	93
Figure 3.3 : Les cas d'utilisation standards pour configurer un e-TP, extraits de l'analyse de situation existante.....	94
Figure 3.4 : Diagramme d'exigences du configurateur affichant les exigences principales du système.	96
Figure 3.5 : Sous exigences de « Création/Édition des ressources ».	97
Figure 3.6 : Liste des sous exigences de EX2 « Génération de configurations ».	98
Figure 3.7 : Liste des sous exigences de EX3 « Exploitation des ressources ».	100
Figure 3.8 : Diagramme d'exigences du configurateur que le prototype satisfera.....	101
Figure 3.9 : Cas d'utilisation types - Phase de création des configurations.	102
Figure 3.10 : Cas d'utilisation types - Phase d'utilisation des configurations.	102
Figure 3.11 : Le domaine du configurateur, ses utilisateurs et son environnement.	104
Figure 3.12 : Décomposition des critères de recherche.	105
Figure 3.13 : Décomposition des types de ressources brutes.....	105
Figure 3.14 : Ressources produites.	106
Figure 3.15 : Les entrées/sorties du configurateur.	106
Figure 3.16 : Structure du « Composant ».	108
Figure 3.17 : Structure de composant (configuration générique).....	109

Figure 3.18 : Génération de versions d'un composant pour en générer plusieurs configurations.....	110
Figure 3.19 : Flux d'activités au sein du configurateur.....	112
Figure 3.20 : Première décomposition fonctionnelle du configurateur.....	113
Figure 3.21 : Composants de l'environnement d'e-TP.....	114
Figure 3.22 : Allocation des activités aux partitions de structure du configurateur.....	115
Figure 3.23 : Structure de la bibliothèque des ressources pédagogiques.....	116
Figure 3.24 : Entrées et sorties des sous-blocs de la bibliothèque des ressources pédagogiques.....	116
Figure 3.25 : Connexions entre les blocs du « Bibliothèque des ressources pédagogiques ».....	118
Figure 3.26 : Structure du générateur de configuration.....	119
Figure 3.27 : Entrées et sorties des sous-blocs du générateur de configuration.....	120
Figure 3.28 : Connectivité entre les blocs du « générateur de configuration ».....	121
Figure 3.29 : Diagramme d'activité illustrant l'assemblage des ressources génériques par l'assembleur.....	122
Figure 3.30 : Diagramme d'activité illustrant la génération de configuration par le générateur de configuration.....	122
Figure 3.31 : Structure de la bibliothèque de configurations.....	124
Figure 3.32 : Entrées et sorties des sous-blocs de la bibliothèque de configuration.....	124
Figure 3.33 : Connexions entre les blocs de la « bibliothèque de configuration ».....	126
Figure 3.34 : Structure de l'installateur de configuration.....	127
Figure 3.35 : Entrées et sorties des sous-blocs du générateur de configuration.....	128
Figure 3.36 : Connexions entre les blocs de l' « Installateur de configuration ».....	129
Figure 3.37 : Logique de fonctionnement de l'installateur de configuration.....	129
Figure 3.38 : Étapes de développement du configurateur.....	130
Figure 4.1 : "Technology stack" de Drupal.....	135
Figure 4.2 : Vue du noyau de Drupal.....	136
Figure 4.3 : Passage de configuration générique à configuration spécifique.....	140
Figure 4.4 : Taxonomies.....	142
Figure 4.5 : Exemple des termes correspondant à des thèmes pédagogiques.....	143
Figure 4.6 : Page de création des catégories.....	145
Figure 4.7 : Page de création d'un élément de programme.....	146
Figure 4.8 : Sélection des catégories correspondant au composant avant sa création.....	147
Figure 4.9 : Page de création d'un composant.....	147
Figure 4.10 : Diagramme de séquence illustrant le processus de création de configuration.....	148
Figure 4.11 : Écran affichant les résultats de recherche lancée par l'utilisateur.....	149
Figure 4.12 : Contenu du fichier ZIP téléchargé contenant les ressources à compiler.....	150
Figure 4.13 : Liste de configurations enregistrées dans la bibliothèque de configuration.....	151
Figure 4.14 : Le magasin vertical.....	152
Figure 4.15 : Le magasin vertical et le bras rotatif avec les pièces métalliques.....	153
Figure 4.16 : Machine à état illustrant le fonctionnement du magasin rotatif.....	153
Figure 4.17 : Grafcet maître (pour le magasin vertical).....	155
Figure 4.18 : Représentation schématique du système de transfert libre.....	163
Figure 4.19 : (A) Le robot ADEPT (poste 4), (B) Le robot STAUBLI (poste 6).....	164
Figure 4.20 : Schéma d'un poste de travail.....	164
Figure 4.21 : Exigences pour l'application de dessin 2D.....	180
Figure 4.22 : Classe représentante les opérations sur les matrice (pour la 1ère session).....	182
Figure 4.23 : Extension de classe des opérations basiques pour introduire des nouvelles fonctionnalités.....	183

Figure 4.24 : Analyse de niveau de satisfaction des utilisateur par le configurateur	190
Figure A.1 : L'architecture de solution orientée objet.....	205
Figure B.1 : Les trois étapes générales de la modélisation orientée objet.....	207
Figure B.2 : Historique de méthodes orientées objet (AxelScheithauer)	208
Figure B.3 : SysML en tant que profil UML.....	210
Figure B.4 : Les 9 types de diagrammes SysML.	212
Figure B.5 : Définition du système.	212
Figure B.6 : La démarche de l'ingénierie système.....	214
Figure B.7 : Processus d'ingénierie système ANSI/EIA 632.....	215
Figure B.8 : Le processus SIMILAR.	217
Figure B.9 : Le processus ESDL.	217
Figure B.10 : Le processus HARMONY.	218
Figure B.11 : Transformation de modèle selon le paradigme MDA.....	220
Figure D.1 : Représentation schématique de l'AIP, avec les 6 automates et postes de travail respectifs, ainsi que le robot Staubli RX90.	229
Figure D.2 : Réseaux de communication à l'AIP.	230
Figure D.3 : Schéma d'un poste de travail.....	231
Figure D.4 : Configuration d'un automate sur Unity Pro.	232
Figure D.5 : Interface graphique de Unity Pro lors de l'ouverture du fichier de configuration.	233

Table des tableaux

Tableau 1.1 : Les dimensions du e-learning.....	21
Tableau 1.2 : Les différentes formes du e-learning.....	27
Tableau 1.3 : Critères pour l'évaluation de l'utilisabilité des outils e-learning et les lignes directrices.....	31
Tableau 1.4 : Composantes pédagogiques utilisées dans un environnement du e-learning.....	33
Tableau 1.5 : Avis paradoxaux sur l'impact des TIC sur la performance des apprenants dans l'enseignement supérieur.....	38
Tableau 1.6 : Apports des TIC pour améliorer les cours.....	40
Tableau 1.7 : Apports des TIC pour améliorer les TP.....	44
Tableau 1.8 : Apports des TIC pour améliorer la pédagogie du projet.....	45
Tableau 1.9 : Apports des TIC pour améliorer la pédagogie du stage.....	47
Tableau 2.1 : Principales mutations dans l'utilisation des ressources dans un cadre de TP en e-learning.....	60
Tableau 2.2 : Analyse SWOT pour la plate-forme de TP Virtuel (VLab).....	63
Tableau 2.3 : L'analyse SWOT pour la plate-forme de téléTP (RLab).....	65
Tableau 2.4 : L'analyse SWOT pour la plate-forme de TP Hybride (HLab).....	67
Tableau 2.5 : Aspects manquants dans les ELabs existants.....	67
Tableau 2.6 : Comparaison entre les différents types de plate-formes de TP.....	69
Tableau 2.7 : Structure du tableau des risques selon la méthode AMDEC.....	76
Tableau 2.8 : Les cas d'utilisation d'un SAP.....	78
Tableau 3.1 : Exigences principales à remplir par le système de configurateur.....	96
Tableau 3.2 : Liste des sous exigences de EX1 « Création/Édition des ressources ».....	97
Tableau 3.3 : Liste des sous exigences de EX2 « Génération de configurations ».....	98
Tableau 3.4 : Liste des sous exigences de EX3 « Exploitation des ressources ».....	99
Tableau 3.5 : Les acteurs principaux du système de configuration.....	103
Tableau 3.6 : Les champs de métadonnées.....	108
Tableau 3.7 : Ports associés aux sous blocs de la « Bibliothèque des ressources pédagogiques ».....	117
Tableau 3.8 : Ports associés aux sous blocs du « Générateur de configuration ».....	120
Tableau 3.9 : Ports associés aux sous blocs du « Bibliothèque de configuration ».....	125
Tableau 3.10 : Ports associés aux sous blocs de l' « Installateur de configuration ».....	128
Tableau 3.11 : Matrice de traçabilité des exigences.....	132
Tableau 4.1 : Bilan des temps d'utilisation du prototype de configurateur.....	178
Tableau 4.2 : Estimation de timing sur l'utilisation du configurateur d'une version plus mature.....	179
Tableau 4.3 : Résultats de l'évaluation de performance du configurateur par les utilisateurs.....	189
Tableau C.1 : Références des modules.....	222
Tableau C.2 : Module TSX DMZ 28DR (position 1) : 16 entrées binaires 24VCC.....	223
Tableau C.3 : Module TSX DMZ 28DR (position 2) : 16 sorties binaires à relais.....	223
Tableau C.4 : Module TSX DEZ 12D2 (position 3) : 12 entrées binaires 24VCC.....	223
Tableau C.5 : Module TSX AEZ 801 position 5) : 8 entrées analogiques +/- 10V.....	223
Tableau C.6 : Module TSX ASZ 401 (position 6) : 4 sorties analogiques +/- 10V.....	223
Tableau C.7 : Codage du clavier.....	225

Introduction générale

Les Technologies de l'Information et de la Communication (TIC) ont, à travers la dernière décennie, impacté notre vie quotidienne tant personnelle que professionnelle. Cette révolution numérique concerne de multiples domaines tel que le social, le médical, l'organisationnel, la production..., et l'éducation. Dans le domaine pédagogique, les Technologies de l'Information et de la Communication pour l'Enseignement (TICE) ont ouvert la voie à de nouvelles modalités (télé-Cours, télé-TD, télé-Projets, Learning-Games, télé-TP, ...).

La réussite de cette exploitation des TICE en enseignement, quelles que soient les disciplines, dépend de la viabilité des solutions proposées et de leur aptitude à évoluer pour fournir des services améliorant la qualité de l'enseignement ou de l'apprentissage. Quel que soit le secteur d'enseignement où les TICE sont utilisés, la façon d'enseigner en a été profondément affectée : décloisonnement de l'acte pédagogique, formation participative, implication contextuelle, variété pédagogique, adaptation aux formateurs et aux publics d'apprenants, simulation, vérification et certification de compétences...

Après avoir présenté le contexte scientifique de notre recherche, nous abordons la problématique, les objectifs de nos travaux et l'approche méthodologique adoptée. Enfin, nous présentons l'organisation de ce rapport.

Contexte de recherche

Les travaux de recherche présentés dans cette thèse ont été menés au sein de la composante INSA de Lyon du laboratoire LIRIS¹ (Laboratoire d'InfoRmatique en Image et Systèmes d'information). Le LIRIS est une unité mixte de recherche (UMR 5205) dont les tutelles sont le CNRS, l'INSA de Lyon, l'Université Claude Bernard Lyon 1, l'Université Lumière Lyon 2 et l'Ecole Centrale de Lyon. Il développe une activité de recherche et un savoir-faire au service de la société, en liaison étroite avec les disciplines de l'Ingénierie, des Sciences Humaines et Sociales, des Sciences de l'Environnement et des Sciences de la Vie.

Nous nous intéressons aux Environnements Informatiques pour l'Apprentissage Humain (**EIAH**) et plus particulièrement à ceux liés aux enseignements par l'expérimentation,

¹ Voir : <https://liris.cnrs.fr/>

communément appelé « travaux pratiques ». Ce mode pédagogique s'est révélé indispensable dans les formations scientifiques et techniques [Blanch-2010], en présentiel [Candelas et al.-2006, Corter et al.-2007] comme en distanciel [Domínguez et al.-2011, Logar et al.-2011]. Il a fait l'objet de recherches visant à fournir aux apprenants, dans un premier temps, un environnement informatique dédié à un besoin pédagogique spécifique. Ce n'est que dans une deuxième phase que l'environnement s'est détaché de l'objectif pédagogique pour supporter des contenus pédagogiques prônant l'expérimentation. Cet environnement n'est pas seulement dédié à l'apprenant mais également à l'auteur du contenu pédagogique, à l'instructeur et aussi, dans le cadre des travaux pratiques, à la personne responsable de la mise à disposition du matériel pédagogique, afin que l'ensemble de la chaîne éditoriale pédagogique soit homogène.

À l'INSA de Lyon, l'Atelier Inter Etablissement de Productique de Rhône Alpes (AIP-Priméca RAO) accueille des installations industrielles (systèmes automatisés, systèmes robotisés, ...), sur lesquelles se déroulent des enseignements sous forme de travaux pratiques et de projets. Les installations sont partagées entre plusieurs établissements et exploitées dans plusieurs disciplines scientifiques et techniques (automatique industrielle², productique, ...).

Nous avons profité de ce terrain expérimental pour étudier comment un EIAH pouvait répondre aux attentes des différents utilisateurs de l'AIP. La section suivante révèle les difficultés rencontrées par les utilisateurs de ces installations automatisées.

Éléments de problématique

Le point de départ de cette recherche est le croisement de deux thèses déjà soutenues : la première s'inscrit dans le domaine des **EIAH**. Dans cette thèse [Benmohamed-2007], un modèle générique de téléTP en sciences de l'ingénierie a été proposé, accompagné d'une méthodologie de mise à distance de dispositifs technologiques. L'ensemble a été élaboré dans un souci de généricité et de réutilisabilité des ressources liées à tout téléTP. La deuxième thèse s'inscrit dans le domaine de conception des Systèmes Automatisés. Dans cette thèse [Chiron-2007] propose une méthodologie pour concevoir des systèmes automatisés. Elle s'appuie sur la génération automatique des codes automates compatibles avec la norme **IEC-61131-3**.

² Par la suite, nous utiliserons l'anglicisme automation pour désigner l'automatique industrielle

L'AIP-Priméca-RAO mutualise des ressources matérielles et logicielles lourdes, pour répondre aux exigences d'une grande variété de champs de formation dans des disciplines techniques ou scientifiques (automatismes industriels, logistique, maintenance, ...).

Nous avons observé que le processus de (re)configuration d'un système automatisé, pour préparer une session de TP, est une opération de plusieurs heures rigoureuse : téléchargement de programmes dans les automates programmables industriels (API) et dans les robots industriels, réglages mécaniques, remplissage des stocks, redémarrage des robots et des systèmes automatisés liés, etc. Cette procédure **varie selon le TP, ses objectifs pédagogiques, le niveau des apprenants et la discipline enseignée**. Elle exige un niveau élevé d'expertise couvrant l'ensemble de l'installation, expertise que ne possède pas nécessairement chaque instructeur (par exemple : un instructeur en Génie Industriel n'est pas spécialiste en automatisme). La personne disposant de l'expertise nécessaire est en général un technicien, qui prend en charge cette procédure dans le cas où l'instructeur n'est pas autonome. L'absence ou l'indisponibilité temporaire de ce technicien peut bloquer la plateforme et limiter son exploitation en termes de temps ou de fonctionnalités.

Le partenariat entre plusieurs établissements et le partage de ressources lourdes dans l'atelier « AIP RAO Priméca » imposent un calendrier serré. Les configurations de chaque matériel doivent être régulièrement modifiées entre deux sessions, et leur usage intense impose aussi de réduire le temps nécessaire pour les configurer. En plus de la charge de travail requise pour configurer le matériel d'un système, nous avons observé un problème de collaboration entre les instructeurs. Actuellement, chaque instructeur développe ses propres programmes sans avoir une vue globale concernant qui développe quoi, ou qui utilise le système pour quel usage, et l'échange d'expériences s'avère difficile. L'échange et la gestion des ressources pédagogiques, comme les documentations et les scénarios pédagogiques, est aussi limité à cause de cette désorganisation.

Nous avons résumé la problématique de recherche en trois éléments :

- **autonomie de l'instructeur** : lorsqu'il n'est pas spécialiste en automation, l'instructeur manque d'autonomie lors de la reconfiguration d'un système automatisé ;
- **temps de configuration** : la configuration d'un système exige un temps trop important du fait de la non coordination des emplois du temps de chaque partenaire ;

- **réutilisation des ressources de-TP** : la réutilisation des ressources développées par d'autres collègues n'est pas facilitée ni encouragée. Les différents intervenants travaillent plutôt de manière solitaire et ont tendance à ne réutiliser que leurs propres ressources.

Objectifs de recherche

Notre objectif est de proposer un EIAH pour gérer le processus de (re)configuration de Systèmes Automatisés de Production (SAP) dans un contexte de Travaux Pratiques. Pour améliorer les conditions d'exploitation d'un SAP quelconque au sein d'un atelier tel que l'AIP-Priméca RAO, il ne suffit pas d'automatiser le processus de configuration de ce SAP, mais il faut également élaborer une **chaîne éditoriale** aidant les auteurs de contenu pédagogique à éditer des contenus à jour et cohérents avec les ressources et matériels utilisés dans leurs TP. La mise à disposition d'un tel outil informatique vers les auteurs et instructeurs de TP facilitera le partage et la réutilisation des ressources exploitées sur un SAP donné. Il s'ensuivra une réduction du temps de préparation des ressources pédagogiques et du temps de configuration du SAP.

Notre recherche se caractérise donc par la proposition d'un EIAH améliorant le processus de (re)configuration d'un SAP dans un contexte pédagogique, pour atteindre trois objectifs :

- **diminuer le temps de configuration** d'un SAP d'au moins 50 % ;
- **améliorer, d'au moins 50 %, la réutilisation des ressources pédagogiques** afin de diminuer le temps de création de deux configurations de TP proches ;
- **améliorer, d'au moins 50%, l'autonomie de l'instructeur** d'une session en réduisant sa dépendance à un technicien.

Plan de la thèse

Le premier chapitre présente l'environnement scientifique de la recherche : le e-Learning. Nous abordons dans un premier temps l'émergence des TIC et leur intégration dans les universités, et la formation ouverte et à distance. Puis, nous présentons la variété pédagogique et situationnelle du e-learning ainsi que ses outils. Nous évoquons ensuite les effets des

nouvelles technologies sur les différents types d'enseignement ainsi que l'évolution de ces types avec les TICE.

Le deuxième chapitre présente le contexte spécifique de cette thèse : l'environnement de TP étendu via les TICE. Nous situons les TP dans une démarche globale de e-learning en montrant l'évolution de l'environnement de TP vers l'e-TP. Puis, nous détaillons l'émergence des différents types de TP supportés par les TIC. Ensuite, nous discutons les problématiques repérées dans le contexte d'e-TP en se rapprochant de la problématique de notre recherche. Nous terminons ce chapitre en précisant, à la lumière de ces éléments, l'objectif de cette recherche.

Le troisième chapitre présente l'approche scientifique suivie afin de modéliser notre solution nommée « Configurateur ». Nous avons pris soin de développer un modèle indépendant de toute technologie ou plate-forme d'implémentation cible.

Le quatrième chapitre décrit la mise en œuvre du modèle de configurateur précédent. Nous avons développé une application web de chaîne éditoriale pour la configuration d'un SAP lié à un contexte pédagogique. Les expérimentations présentées ont pour objectif d'évaluer la concordance de notre configurateur aux objectifs énoncés. Nous terminons ce chapitre par une analyse des retours utilisateurs de notre système.

Enfin, pour conclure ce rapport, nous effectuons un bilan de ce travail et nous ouvrons des perspectives de recherche.

1. Contexte général : l'enseignement supérieur et les avancées technologiques

Introduction

Ce chapitre rappelle les missions et rôles des établissements de l'enseignement supérieur dans la construction des compétences personnelles et collectives. Nous évoquerons également en quoi les nouvelles technologies ont bouleversé les modes d'apprentissage et de construction cognitive.

Les universités et les Grandes Écoles d'Ingénieurs et de Commerce Françaises sont des établissements fédérant la production, la conservation, la transmission et la maintenance de différents domaines de la connaissance. Elles sont le plus souvent ancrées dans une recherche scientifique amont portant sur le domaine enseigné (fond) et quelques fois (trop rarement) sur la façon d'enseigner. Cette recherche scientifique aboutit à la construction de modules pédagogiques instrumentés destinés à des apprenants en formation initiale ou professionnelle. Dans les deux cas il s'agit de mettre en place un processus d'apprentissage permettant à un individu (ou un groupe d'individus) d'acquérir le savoir, le savoir-faire et le savoir-être nécessaires à l'exercice d'un métier ou d'une activité professionnelle. Selon Guy Rocher³ « l'université est une institution et, dans la mesure où elle est vivante, il se développe entre elle et la société de multiples rapports, variables en étendue et en intensité selon les époques ». Les universités jouent donc un rôle charnière qui influence tous les secteurs de la société, comme l'indique Brennan [Brennan et al.-2010] qui identifie que l'enseignement supérieur est sujet à différents changements externes (conduits par les changements sociaux et économiques) et internes (conduits par la manière de produire et organiser la connaissance).

Réciproquement l'enseignement supérieur a un impact sur le secteur économique, lieu où s'exerceront les compétences des personnes formées. Une étude, réalisée dans [Stevens-2003] sur le lien entre éducation et croissance économique, analyse le rôle joué par de bonnes pratiques technologiques, et affirme que l'éducation est indispensable en tant que moyen de transfert et d'essaimage. Cette étude établit un lien entre l'augmentation du niveau de vie de ces deux derniers siècles et le rôle du système éducatif au service du développement et de la

³ **Guy Rocher** (20 avril 1924) : sociologue, professeur en sciences sociales au département de sociologie de l'Université de Montréal et chercheur au Centre de recherche en droit public. Cette définition est publiée dans le livre « *L'Éducation 25 ans plus tard! et après?* » en 1990.

dissuasion de connaissance. Une nouvelle économie a ainsi émergé : l'économie du savoir. Les retombées industrielles sont multiples. Nous en distinguons trois niveaux [Chartrand-2005] :

1. les industries utilisant un savoir technologique très développé. Elles exercent dans des activités de pointe (innovation) et utilisent une main d'œuvre hautement spécialisée ;
2. les industries utilisant un savoir-faire couramment maîtrisé. Elles correspondent généralement à des productions de masse et à des secteurs matures (automobiles, produits électriques, métaux primaires, textiles, papier, plastiques, aliments, boissons, etc.) ;
3. les industries ne nécessitant pas d'équipements coûteux et faisant appel à des composantes artistiques ou de service. Ce sont généralement des activités à fort coefficient de main-d'œuvre (bois, meuble, commerce de détail, vêtement, etc.).

Sous l'influence de l'évolution sociale et sociétale (banalisation de l'accès aux études, travail des femmes, essor économique post 2^{ème} guerre mondiale) les établissements d'enseignement supérieur ont dû accueillir au cours de la 2^{ème} moitié du 20^{ème} siècle un nombre croissant d'apprenants français puis étrangers (Erasmus, francophones héritiers de l'empire colonial français, internationalisation du monde éducatif « Supérieur »...). Cela conduit aujourd'hui à redonner à l'Enseignement Supérieur un rôle central en réponse à une demande exigeante sans précédent : **former au bon endroit, au plus vite, en apportant la connaissance juste utile et au moindre coût !** –Remarquons que l'Enseignement Supérieur empêtré dans ses archaïsmes est loin de satisfaire encore cette « **démarche qualité** » – Lors de la conférence mondiale sur l'enseignement supérieur en 1998 l'UNESCO souligne le rôle vital que l'enseignement supérieur est amené à jouer dans le développement socioculturel et économique des pays et dans la définition de nouvelles lignes directrices, qui demandera aux nouvelles générations de nouvelles compétences, de nouvelles connaissances, de nouveaux idéaux⁴.

1.1 Émergence des STIC et leur intégration dans les universités

Dans cette section nous resituons l'émergence progressive des STIC, leur intégration dans l'enseignement supérieur et l'émergence du e-learning.

⁴ Déclaration mondiale sur l'enseignement supérieur pour le XXI^e siècle : Vision et Actions, le 9 octobre 1998, (http://www.unesco.org/education/educprog/wche/declaration_fre.htm).

Ces vingt dernières années, les Sciences des Technologies de l'Information et de la Communication (STIC) ont transformé directement (par la maîtrise de nouveaux dispositifs) et indirectement (par la construction d'une pédagogie spécifique) la façon d'enseigner au sein des universités de plusieurs pays (les pays du Conseil de Coopération du Golfe (CCG) [Wiseman-2012], en Corée du Sud et au Chili [Sánchez et al.-2011], Malaisie [Jamian et al.-2012], et bien sûr dans les pays européens [Hextlearn-2010], etc...) (voir Figure 1.1). Cette transformation concerne tous les domaines d'enseignement (les sciences de l'ingénieur [Jordan-2008, Moum et al.-2009], l'économie [Weber-2011, Schumann et al.-2012], les arts [Ott-2011], les sciences humaines [Louw et al.-2009], les mathématiques [Wang-2011], les langues [Ghasemi-2011, Hubackova-2011], etc.), en provoquant successivement :

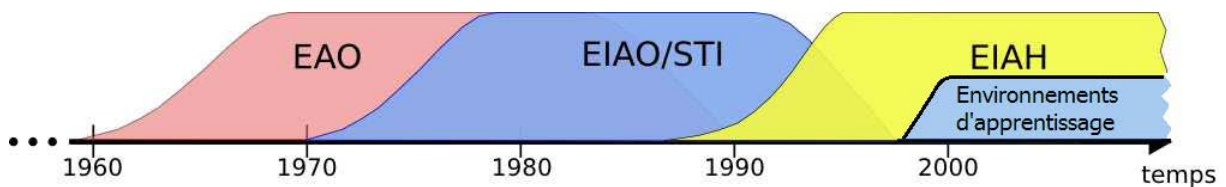


Figure 1.1 : Évolution des termes qualifiant l'utilisation des TIC dans l'éducation⁵.

1. l'émergence dans les années 80 de l'Enseignement Assisté par Ordinateur (EAO) qui reposait sur une délégation à l'ordinateur de contenus pédagogiques « en l'état » (i.e. le plus souvent sous forme textuelle ou de « transparents » d'origine). Cette décennie a assisté à l'émergence des Systèmes Tutoriels Intelligents (STI) liés au développement des systèmes de base de connaissances en Intelligence Artificielle (IA) [Benadi-2003]. Ce fut une période pendant laquelle l'IA a été adoptée pour simuler le comportement humain par la reproduction de ses faits et gestes. En enseignement, l'idée était d'employer l'IA pour construire des systèmes où l'ordinateur devient la copie du tuteur enseignant en remplissant son rôle et le déchargeant d'une partie (voire de la totalité) de sa tâche, ce qui a abouti à l'EIAO (Enseignement Intelligemment Assisté par Ordinateur) ;
2. la Formation Ouverte et À Distance (FOAD), dans les années 90, qui introduit la notion de ressources partagées disponibles via un réseau privé ou public (ouverture à Internet) et accessibles sous la férule d'un formateur-chef d'orchestre. C'est un processus pour créer et fournir un accès à la formation dans le cas où les ressources et les apprenants sont séparés par le temps, la distance ou les deux [Honeyman-1993]. La FOAD désigne les processus de **formation initiale** ou **continue**, individuels ou

⁵ Extrait de [Gravier-2007] p.32 et modifiée

collectifs se faisant à distance, y compris les cas où la présence physique est requise (présence sur place, d'un ou plusieurs intervenants, dans des cours distants comme des cours hybrides [Tabor-2007] ou mélangés « blended » [Hubackova-2011]) ;

3. le e-learning, dès le milieu des années 90, qui intègre une pédagogie propre et un usage maîtrisé des nouvelles technologies de communication (en particulier les IHM⁶). Cette émergence du e-learning a ouvert la voie à un plus grand public : aux apprenants traditionnels des universités et des grandes écoles sont venus s'ajouter d'autres demandeurs de formation tels que les adultes cherchant :

- une deuxième compétence en plus de leur formation initiale, augmentant ainsi leur « valeur » sur le marché du travail ;
- à valoriser leurs acquis professionnels pour progresser dans leur carrière (souvent dans leur entité d'origine, mais pas forcément) ;
- répondre à un désir personnel d'accéder à des connaissances et compétences (e.g. cas de retraités qui du fait de l'allongement de la vie, disposent de beaucoup de temps libre).

Ainsi aux États-Unis, entre 1997 et 1998, le nombre total d'inscriptions délivrées dans les différentes formes de formation distante a été estimé à près de 1.6 million d'apprenants [Harper et al.-2004], ce qui indique un chiffre d'affaire élevé et reflète le grand intérêt pour ce nouvel environnement par les universités, les entreprises et les gouvernements [Oblinger-2000]. Enfin, plus récemment, l'émergence de nouveaux environnements qui répondent aux besoins de tous les acteurs : concepteurs, formateurs, apprenants, gestionnaires, établissements... préfigure une industrialisation du système éducatif dans une « démarche qualité ». L'apparition d'environnements Auteurs/Formateurs répond essentiellement au souci de minimiser le coût de production des systèmes de formation ainsi que la nécessité de capitaliser savoirs et expériences (développement de communautés de pratique). Ils doivent aussi assurer la qualité du contenu et de la pédagogie et permettre la participation des acteurs de profils différents (enseignant, apprenant, etc.) et, à ne pas omettre, fournir une plate-

⁶ IHM : L'Interaction Homme-Machine étudie la façon dont les humains interagissent avec les ordinateurs ou entre eux à l'aide d'ordinateurs, ainsi que la façon de concevoir des systèmes informatiques qui soient ergonomiques, c'est-à-dire efficaces, faciles à utiliser ou plus généralement adaptés à leur contexte d'utilisation

forme d'évaluation des alternatives de stratégies et de modèles du contenu (par exemple : RIDES [Munro et al.-1997], XAIDA [Hsieh-2001], Eon [Kluwer et al.-2003], etc.). Wagner, dans [Wagner et al.-2008] (voir Tableau 1.1) a signalé l'émergence des systèmes de gestion de configuration pour traiter la préparation des environnements de e-learning et les maintenir fonctionnels pendant leur exploitation. [Chen et al.-2012] décrit une structure de portail configurable et extensible qui peut être adaptée à des environnements spécifiques de e-learning. Le but du portail est de construire des configurations e-learning adaptées à des contextes et objectifs pédagogiques spécifiques dans un environnement e-learning ;

Dimension	Attribut	Sens
Synchronisme	Asynchrone	La fourniture du contenu se produit à un moment différent de la réception par l'apprenant
	Synchrone	La fourniture du contenu se produit en même temps que la réception par l'apprenant
Localisation	Co-localisé	Les apprenants utilisent une application au même emplacement physique que les autres apprenants et / ou de l'instructeur
	Distribué	Les apprenants utilisent une application à différents endroits physiques, séparée des autres apprenants et l'instructeur
Indépendance	Individuel	Les apprenants travaillent de façon indépendante les uns des autres pour accomplir des tâches d'apprentissage
	Collaboratif	Les apprenants travaillent en collaboration pour effectuer des tâches d'apprentissage
Mode	Électronique seulement	Tout le contenu est livré via la technologie, il n'y a pas de composante en face-à-face
	Mélangé	Le e-learning est utilisé pour effectuer de l'apprentissage traditionnel en classe

Tableau 1.1 : Les dimensions du e-learning.

4. l'émergence des EIAH (Environnement Informatisé pour l'Apprentissage Humain). Ce terme remplace le précédent sous l'influence d'un nouveau courant en sciences cognitives appliquées à l'éducation. Dans les EIAH, l'ordinateur n'est plus seulement détenteur d'un savoir qu'il prodigue au coup par coup, comme c'était le cas pour les EIAO/STI, mais il a pour but de favoriser les situations où l'apprenant va lui-même construire son savoir, à partir d'observations, de questions ouvertes et de mises en situation ; l'apparition des MOOC⁷, des cours en ligne ouverts et massifs [Gilliot-2013], qui constituent un nouveau mode de formation ouverte et à distance en télé-enseignement. Les participants aux cours, enseignants et apprenants, sont dispersés géographiquement et communiquent uniquement par Internet. Selon Matthieu Cisel

⁷ MOOC : Massive Open Online Course.

[Cisel-2012], le terme est souvent employé pour désigner les plates-formes qui accueillent les cours, mais il est surtout employé pour désigner les cours eux-mêmes. Les MOOC sont donc des cours en général limités dans le temps portant sur un thème spécifique. Ils incluent un ensemble cohérent de ressources pédagogiques, de modalités d'interactions, d'exercices et d'examens conduisant éventuellement à une certification. Ils impliquent une équipe pédagogique, chargée de l'encadrement des apprenants et du bon déroulement des cours.

L'apparition des *Learning Games*, de la réalité augmentée (cf. §1.6.2.6), du *Mobile Learning*⁸, ... concrétisent enfin la véritable naissance d'une Industrie de la formation.

1.2 La formation ouverte et à distance

Nous présentons ici l'apparition de la formation ouverte et à distance, et l'effet TIC sur les différents types d'enseignement.

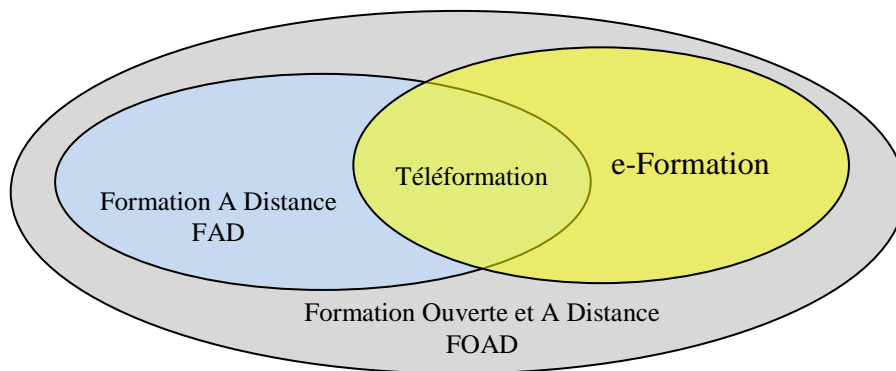


Figure 1.2 : Modes de formation assistée par TIC.

La FOAD⁹ peut être partiellement ou intégralement à distance (voir Figure 1.2). Elle englobe plusieurs sous-ensembles de systèmes de formation. Chacun remplit des objectifs décrits selon les situations qu'il représente et les moyens dont il dispose. Premièrement, la e-Formation¹⁰ (e-Learning) est la formation par voie électronique, c'est-à-dire exploitant l'informatique en tant que moyen pour fournir des ressources électroniques aux apprenants et

⁸ Mobile Learning (ou M-learning) est un concept en plein essor. Actuellement, il vient en complément du e-learning. Il s'agit d'apprendre n'importe où grâce à des modules de formation accessibles sur téléphone portable.

⁹ L'UNESCO caractérise les formations ouvertes par une « *liberté d'accès aux ressources pédagogiques mises à disposition de l'apprenant, sans aucune restriction, à savoir : absence de condition d'admission, itinéraire et rythme de formation choisis par l'apprenant selon sa disponibilité et conclusion d'un contrat entre l'apprenant et l'institution* ».

¹⁰ D'après la commission européenne : la e-formation consiste à utiliser les nouvelles technologies multimédia et de l'Internet, pour améliorer la qualité de l'apprentissage en facilitant l'accès à des ressources et des services, ainsi que les échanges et la collaboration à distance

aux enseignants. Contrairement à ce qui est couramment admis, elle ne se limite pas à la fourniture distante des ressources et outils ([Savaş-2010] par exemple). Elle comprend l'apprentissage/enseignement local à travers des produits pédagogiques (logiciels) exploitables localement sur un PC ou à travers un réseau local ou bien un autre support externe ([Abdulwahed et al.-2011] par exemple). L'utilisation des technologies multimédia (par exemple : formation en linguistique [Hubackova-2011]) pour produire des contenus interactifs, la distribution d'outils logiciels et le support distant à travers Internet (ou un intranet) élargit l'accès aux ressources, la collaboration et également l'interactivité entre les différents utilisateurs de la e-formation. Deuxièmement, dans une situation caractérisée par un aspect de formation distante, la Formation À Distance (FAD¹¹) couvre l'ensemble des dispositifs techniques et des modèles d'organisation qui ont pour but de fournir un enseignement ou un apprentissage à des individus qui sont distants du l'organisme de formation prestataire du service.

Bien évidemment, la FAD et la e-Formation se croisent en ce qui concerne l'accès distant aux ressources et services, mais chacun répond à des besoins résultants des situations de présence de ressources (humaine, logiciel, outil, etc.). Un sous-ensemble de la FAD s'est appelé « téléformation » quand elle s'appuyait essentiellement sur l'utilisation de TIC (plus précisément TICE¹²) via un réseau grand public comme Internet.

Les STIC ont permis la création d'un environnement d'éducation virtuel accessible par un grand public (apprenants, enseignants, tuteurs, chercheurs, etc. soit en présentiel soit à distance) et l'échange facilité de connaissances constamment actualisées. Cet environnement a « bonifié » la qualité des enseignements en donnant la possibilité d'enrichir le contenu pédagogique par des ressources et services électroniques (e-book, e-forums, multimédia, messageries, etc.).

1.3 La pédagogie du « e-Learning »

Le « e-learning » dans la pédagogie est l'utilisation des nouvelles technologies multimédias et de l'Internet pour améliorer la qualité de l'apprentissage en facilitant l'accès à des ressources et des services, ainsi que les échanges et la collaboration à distance.

¹¹ La FAD d'après l'AFNOR: "C'est un système de formation conçu pour permettre à des individus de se former sans se déplacer sur le lieu de la formation et sans la présence physique d'un formateur. La formation à distance recouvre plusieurs modalités (cours par correspondance, e-learning.) et est incluse dans le concept plus général de formation ouverte et à distance".

¹² TICE : Technologie de l'Information et de la Communication pour l'Education.

L'utilisation des TIC pour des fins pédagogiques devra prendre en compte les approches pédagogiques adoptées dans les différents domaines d'éducation, comptons ici les approches :

1. **socioconstructiviste** : une pédagogie collaborative est offerte aux apprenants via les forums, blogs et les activités collaboratives en ligne ;
2. **cognitiviste** : la démarche se concentre sur les processus cognitifs et sur les efforts mentaux impliqués dans le e-learning. Actuellement, cette approche adopte trois concepts pour construire la connaissance chez l'apprenant, ce sont :
 - la progressivité : aller dans le transfert de connaissance du simple au compliqué ;
 - la répétition : répéter à l'apprenant les connaissances déjà transférées en le faisant répéter les activités réalisées par l'enseignant, afin de consolider les notions vues progressivement ;
 - la justification : faire justifier par l'apprenant justifier les activités qu'il a réalisées ;
3. **émotionnelle** : elle se concentre sur l'aspect émotionnel de l'apprentissage comme la motivation, l'engagement, etc. Il s'agit ici de provoquer l'émotion chez l'apprenant en l'impliquant dans des actions (rôle d'acteur), en le transportant dans des micro-mondes, en le valorisant (récompenses, trophées, pouvoirs...), en le mettant dans des conditions où il trouvera **par lui-même** les solutions aux problèmes posés, en lui faisant vivre des situations réalistes (effet catharsis) ;
4. **comportementale** : elle se concentre sur les compétences et les résultats comportementaux du processus d'apprentissage. Il s'agit ici de s'attacher au développement de savoir-être le plus souvent professionnels. Ce sont ces comportements développés qui piloteront les connaissances en leur donnant du sens ;
5. **contextuelle** : elle se concentre sur l'environnement et les aspects sociaux qui peuvent stimuler l'apprentissage comme l'interaction entre les intervenants, la découverte collaborative, la pression et la tension, etc. Il s'agit ici de pratiquer une pédagogie « Bottom-Up » (opposée au « Top-Down » synonyme d'académisme formalisé) en mettant en scène les savoirs dans des situations concrètes inspirées de cas

(professionnels) réels. En contextualisant, la connaissance prend du sens, ce qui facilite sa compréhension et sa rétention.

1.4 Variétés pédagogiques et situationnelles du e-learning

Le E-learning est éligible dans tous les domaines et toutes les formes d'apprentissage ou d'enseignement.

Les EIAH servent de médiateur pour implémenter le processus d'apprentissage [Djamshid et al.-2004]. Les services offerts par les EIAH ont évolué en fonction des exigences pédagogiques exprimées par les différents intervenants du système éducatif (apprenants, tuteurs, instructeurs, responsables des cours, etc.), dans un premier temps, et les variétés situationnelles émergentes de mobilité de ces derniers, dans un deuxième temps.

Il existe aujourd'hui de nombreux types de formation faisant appel aux technologies du e-learning (voir Figure 1.3) :

1. l'apprentissage/formation assisté par ordinateur [Spalter-2000, Balter et al.-2005] (Computer-based learning « **CBL** ») où et quand l'ordinateur est un élément clé de l'environnement éducatif. Ce type de formation ne se limite pas à l'utilisation de l'outil informatique dans la classe ou la salle de cours, mais correspond plus généralement à un environnement construit pédagogiquement. Il existe plusieurs sous-catégories de CBL :

- la (Computer-based training « **CBT** ») correspond à des activités **d'autoformation** accessibles via l'ordinateur. La CBT est souvent adoptée pour un processus d'apprentissage peu dynamique (le CBT utilise des moyens massiques de diffusion comme le CD-ROM), et est utilisée d'une manière interchangeable avec :
- la formation assistée par Internet [Tao-2001] (Web-Based Training « **WBT** ») où la différence clé avec la CBT est le moyen de délivrer le contenu : Internet via un navigateur web.

L'objectif de ces deux modes de formation est de fournir de l'apprentissage plus librement à l'aide des ressources plus riches (manuels, bouquins, etc.), plus conviviales (simulateurs, support multimédia, etc.) et plus échangeable ;

2. **l'apprentissage collaboratif assisté par ordinateur** [Wessner-2001] : (Computer-supported collaborative learning « **CSCL** ») correspond une approche pédagogique dans laquelle l'apprentissage se déroule via les interactions sociales en utilisant l'ordinateur ou par Internet. Il s'agit d'une approche dont le but est d'améliorer l'apprentissage et la formation à l'aide des TIC modernes. En effet, l'apprentissage collaboratif dans le e-learning se distingue du modèle de « transfert direct » traditionnel où l'instructeur est supposé être le distributeur de connaissances et compétences alors que dans e-learning cette distribution est réalisée par l'auto-apprentissage dans un premier temps puis par la collaboration entre les apprenants dans un deuxième temps ;
3. **l'apprentissage assisté par la technologie** [Schmees-2006, Boytchev et al.-2012] : (Technology-enhanced learning « **TEL** ») : quand l'activité d'apprentissage ou l'approche pédagogique est enrichie d'outils technologiques. Les activités d'apprentissage varient selon les ressources (création, distribution, accès, outils, services, etc.), les actions (communication, collaboration, interaction avec des outils logiciels), le contexte (la durée, la localisation et le nombre d'acteurs humains), les rôles (apprenant, tuteur, instructeur, ressources humaines, etc.) et les objectifs pour aider chaque intervenant dans l'achèvement de sa propre mission, en considérant les préférences d'apprentissage d'un individu ou d'un groupe. L'objectif principal du TEL est de faire coopérer toutes ces activités à l'aide des technologies idoines.

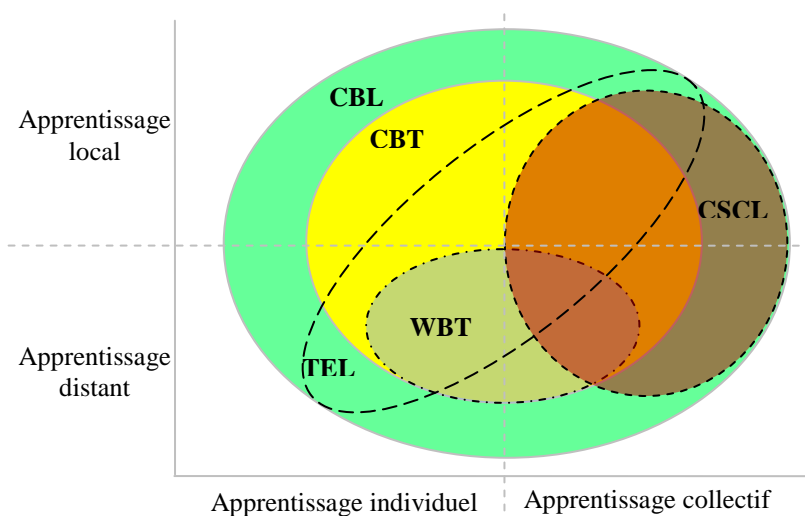


Figure 1.3 : La typologie du « e-Learning ».

Au-delà de ces variétés pédagogiques, le e-learning présente également deux variétés situationnelles, proposé par Romiszowski dans [Romiszowski-2003] et illustrées dans le tableau (Tableau 1.2) : d'une part l'apprentissage en/hors ligne et d'autre part l'apprentissage autonome/collaboratif.

	Apprentissage autonome	Apprentissage collaboratif / Apprentissage coopératif
Apprentissage en ligne Communication synchrone en temps réel	Surf sur Internet, accès à des sites contenant des informations utiles pour une formation (connaissances ou expériences) en ligne (exerciceur web)	<i>Chat rooms</i> avec ou sans vidéo (IRC; tableau blanc, WEB TV), Audio/vidéoconférence (<i>streaming</i> audio et vidéo)
Apprentissage hors ligne Communication asynchrone	Téléchargement d'objet pour une utilisation locale <i>LOD (Learning Object Download)</i>	Communication asynchrone par courriel, liste de discussion ou forum via des plates-formes

Tableau 1.2 : Les différentes formes du e-learning.

1.5 Les outils du e-learning

Les STIC ont engendré le développement de modèles économiques (*business model*) centrés sur un développement de contenus personnalisés (*custom-content development*), ainsi que l'apparition de multiples services et produits (voir Figure 1.4) [Barron-2002].

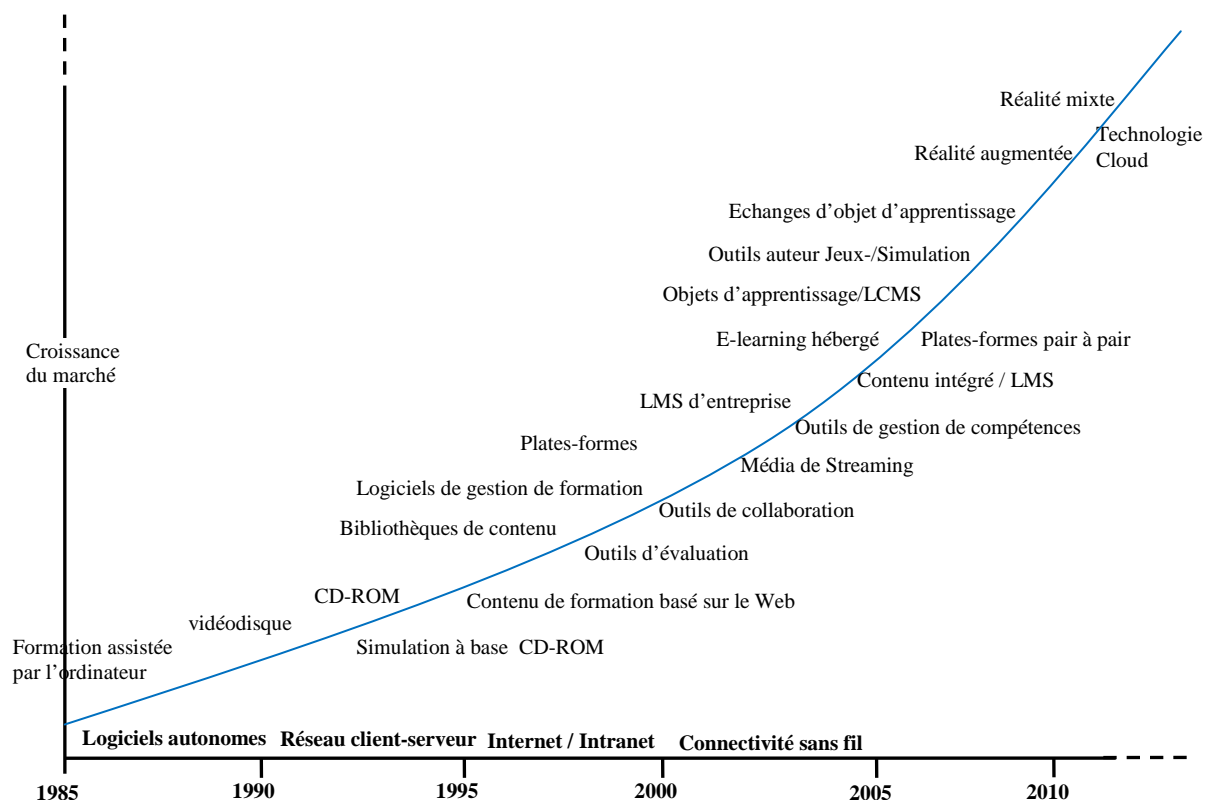


Figure 1.4 : Evolution des technologies dans le e-learning¹³.

Un environnement de type « e-learning » met ainsi à disposition des utilisateurs une variété d'**outils** et de **ressources**, tels que des bases de contenus, des bases de connaissances, des dispositifs multimédia, des bases de traces, des environnements génériques, une gestion de session d'apprentissage, un ajustement de niveau, une gestion d'intervention, ...

1.5.1 Les outils

En fonction des objectifs pédagogiques, on distingue :

1. **les outils d'information** : aides en ligne (Foire Aux Questions, forum, guide d'utilisation, etc.), calendriers, glossaires, listes de diffusion, *newsletters* et autres journaux web ou outils d'édition, etc. Des outils synchrones peuvent aussi être utilisés (*chat*, vidéoconférence, etc.) mais leur vocation communicationnelle les destine à un autre usage que la simple diffusion d'information ;
2. **les outils de communication** : facilitent le travail des tuteurs et permettent la construction d'une véritable communauté d'apprentissage. Ils se subdivisent en trois catégories:

¹³ Consulting Business Intelligence SRIC-BI. Voir : <http://www.sric-bi.com/cfd/?mod=infogr>

- les outils synchrones assurant une communication en temps réel entre des personnes géographiquement distantes : messagerie instantanée (*chat*, messenger, etc.), téléphonie IP, audioconférence et vidéoconférence, etc.;
 - les outils asynchrones qui autorisent des échanges décalés dans le temps et l'espace comme le courrier électronique (*e-mail*), les forums (de discussions), les portfolios, wikis, blogs, etc. ;
 - les outils de collaboration à distance : ils sont essentiellement asynchrones (espace de travail partagé et outils d'édition, tels que blog et autre wiki, portfolio, tableau blanc interactif, etc.). Les outils de communication précédents peuvent également faciliter le travail collaboratif. Un panorama des aspects les plus importants des outils collaboratifs spécifiques au e-learning est présenté dans [Dobre-2006] ;
3. **les outils de dépôt ou de partage de documents** : ils sont essentiellement asynchrones (espace d'échanges et portfolio). Cependant, la plupart des outils de communication (forum, chat) et de collaboration (wiki, blog) assurent ces fonctions ;
 4. **les outils auteur** : ils sont essentiellement destinés aux tuteurs pour créer des contenus pédagogiques. Ces outils servent à créer des modules intégrant textes, images, son, vidéos et animation tout en se conformant à un standard du e-learning (SCORM par exemple). [Kuo-2009] présente un outil auteur pour l'apprentissage mobile (*m-learning*), et compare cet outil à huit autres outils auteurs.

Une étude, détaillée dans [Ardito et al.-2004], identifie quatre dimensions pour analyser l'utilisabilité des outils du e-learning :

1. **la présentation** : regroupe les aspects liés à la visualisation des outils et éléments de la plate-forme e-learning. Cette dimension souligne le niveau de la convivialité de l'interface graphique du système, la possibilité d'opérer graphiquement de manière personnalisée, la traçabilité visuelle des événements, la structuration visuelle des cours ... ;
2. **l'hypermédialité** : les outils hypermédia permettent de communiquer à travers différents canaux et d'organiser les cours de manière non séquentielle. Il est ainsi possible de proposer aux apprenants de choisir un chemin logique différent de celui suggéré. Cette dimension souligne le niveau de dynamisme et d'animation du

système, la variété de contenu, la mobilité entre sujets, la disponibilité de communication personnalisée ... ;

3. **la proactivité de l'application** : rassemble les outils logiciels qui ne sont pas strictement liés à la lecture du contenu (outils de communication par exemple) et qui favorisent l'apprentissage et améliorent la prévention des erreurs lors de l'utilisation des autres outils de la plate-forme. Cette dimension souligne l'agilité et la sûreté du système, la simplicité d'utiliser ses outils, son adaptabilité au contexte d'utilisation ... ;
4. **l'activité de l'utilisateur** : représente la flexibilité du système en réponse aux besoins de l'utilisateur : disponibilité des outils, simplicité d'utilisation, prise en compte des préférences... ... ;

Pour chaque dimension, cette étude considère les principes d'efficacité et d'efficience selon les critères illustrés dans le Tableau 1.3, avec des lignes directrices dérivées de chaque critère.

Dimension	Critère	Ligne directrice
Présentation	Efficacité d'apprentissage/création de contenus	Cohérence de la mise à jour du contenu
		Les matières de haute priorité sont surlignées
		Le schéma (layout) graphique ne conduit pas à distraire l'apprenant
		La structure hiérarchique des sujets du cours est claire et mise en évidence
	Efficience des modalités d'enseignement et support	Les techniques d'enseignement sont affectées d'un espace non-invasive pour ne pas distraire l'apprenant
Hypermediality	Efficacité d'apprentissage/création de contenus	Les outils utilisés sont capables à plonger l'apprenant dans le contexte de domaine enseigné
		Des médias de communication spécifiques sont utilisés pour tout objectif d'apprentissage
	Efficience des modalités d'enseignement et support	Les canaux de communication sont utilisés de façon optimale
		Les liens hypertextes et hypermédias sont soigneusement utilisés
		Matériel d'apprentissage peut être réutilisé
Application proactivité	Efficacité d'apprentissage/création de contenus	Des outils spécifiques de domaine d'apprentissage sont disponibles
		L'assistance et les techniques d'enseignement sont soigneusement choisies
		Les outils de teste sont fiables
	Efficience des modalités d'enseignement et support	Affinement correct des techniques d'enseignement
Les formats de document utilisés n'exigent pas des plugins spécifiques		
Activité de l'utilisateur	Efficacité d'apprentissage/création de contenus	Il est possible de limiter ou de choisir les canaux de médias
		Simulations pour l'apprentissage mixte sont disponibles
	Efficience des modalités d'enseignement et support	La recherche des documents est simplifiée par une spécification claire et correcte de mots clés

Tableau 1.3 : Critères pour l'évaluation de l'utilisabilité des outils e-learning et les lignes directrices¹⁴.

1.5.2 Les ressources et composantes pédagogiques

Dans un environnement e-learning, les ressources de base pour construire des contenus pédagogiques sont : des textes, des images, des fichiers audio, des vidéos et des animations. À cette « **Base de contenus** » s'ajoute la « **Base de connaissances** » qui rassemble les expertises du domaine d'enseignement afin de collectionner, organiser, partager et chercher des informations liées à ce domaine. Elle sert à la fois deux objectifs :

- fournir des connaissances lisibles par une machine, dans le but de réaliser des raisonnements automatisés déductifs logiques (DBpedia¹⁵, par exemple).
- fournir des connaissances lisibles par l'homme capitalisant une variété d'expertises sur différents sujets catégorisés dans plusieurs domaines d'enseignements (par exemple : les manuels, les documentations, les points critiques, les FAQ, etc.). Cette composante est typiquement accompagnée d'un moteur facilitant la recherche et la localisation des informations selon des : mots clés, catégories, classes, etc. (par exemple, les logiciels Wiki peuvent fournir une base de connaissance fondée sur les mécanismes hypertextes¹⁶).

Pour la gestion de sessions d'apprentissage, une autre composante est utilisée : c'est le **LMS** (abréviation de l'anglais : *Learning Management System*). Il s'agit d'un logiciel web développé pour accompagner et gérer le parcours pédagogique des apprenants. Cette composante essentielle donne généralement la possibilité de contrôler les accès, de gérer la communication (synchrone et/ou asynchrone) et d'administrer des groupes d'utilisateurs (par exemple : Blackboard¹⁷, Moodle¹⁸). Par contre il ne permet pas de créer, modifier ou réutiliser directement du contenu. Cette tâche est adressée au système de gestion de contenu d'apprentissage LCMS (abréviation de l'anglais *Learning Content Management System*) qui est un LMS augmenté avec des capacités d'aide à la génération de nouveaux contenus, par greffe de nouvelles entités pédagogiques. Le LCMS fournit ainsi aux auteurs une vraie chaîne éditoriale afin de développer des cours avec économie d'échelle. Moodle, qui intègre

¹⁴ Extrait de [Ardito et al.-2004].

¹⁵ DBpedia est un effort collectif pour extraire des informations structurées à partir de Wikipedia, et de les rendre disponibles sur l'internet. Voir : <http://dbpedia.org/About>

¹⁶ Système hypertexte est un système de base de texte qui peut comporter des groupes de documents avec des hyperliens entre eux.

¹⁷ <http://www.blackboard.com/>

¹⁸ <http://moodle.org/>

SCORM¹⁹ dans ses versions actuelles, est une plate-forme d'apprentissage en ligne sous licence libre servant à créer des communautés s'instruisant autour de contenus et d'activités pédagogiques. Une autre composante s'est ajoutée au tableau, plus récemment : c'est le **Système à Base de Traces**. Il s'agit d'une classe de systèmes à base de connaissances facilitant l'exploitation et le raisonnement sur des traces d'utilisation d'un logiciel d'apprentissage [Settoui-2006]. Selon [Settoui et al.-2007] le principe consiste à enregistrer les « traces de l'activité d'un utilisateur qui utilise un outil informatique pour mener à bien cette activité, s'inscrivant sur un support numérique ». Actuellement, les participants à l'activité d'apprentissage interagissent avec un environnement d'apprentissage individuellement ou en groupes. Suivant leur rôle respectif, les participants ne tireront pas partie des traces de la même manière. Le système à base des traces permet de personnaliser l'environnement d'apprentissage pour différents acteurs à l'aide de leurs traces.

Par exemple, un instructeur peut guider l'activité individuelle ou collaborative en essayant de comprendre les dysfonctionnements éventuels par rapport au scénario qu'il avait préconisé. Il peut alors adapter la session, introduire des aides personnalisées, fournir des supports pédagogiques adaptés aux différents publics. Un auteur peut exploiter les traces pour personnaliser un scénario pédagogique, permettant ainsi de réguler le déroulement d'une session d'apprentissage en tenant compte de certains aspects qui ne peuvent être mesurés qu'en cours de session, tel le temps de réponse à un exercice. L'apprenant peut visualiser sa trace et se faire une image de son évolution dans l'activité, ce qui lui permettra de comprendre son cheminement dans la construction de sa connaissance.

Le Tableau 1.4 présente une description de quelques composantes fréquemment utilisées dans un environnement de e-learning.

Composante pédagogique	Description
Base de contenus	Contient des contenus pédagogiques construits depuis des ressources basiques à l'aide des outils e-learning
Base de connaissances	Une base de données spécifique rassemblant les expertises dans un domaine d'enseignement donné. Elle dispose des moyens pour collectionner, organiser, partager et chercher des informations liées à un domaine.
Dispositifs multimédia	Tout dispositif permettant de créer un objet d'animation sous forme d'une série d'images diverses, présentées de manière dynamique en fonction de l'action de l'utilisateur de manière à l'aider à percevoir un changement animé dans le temps et à développer un modèle mental cohérent de la tâche [Clavien-2003]

¹⁹ **SCORM** : « *Sharable Content Object Reference Model* » une collection de standards et spécifications pour la e-formation assisté par l'internet

Patterns	Un pattern est une description structurelle (textuelle ou graphique) d'une solution qui résout un problème récurrent. En analysant des comportement-types d'apprenants et en définissant les stéréotypes d'usage correspondants, on obtiendra des problèmes communs et récurrents. Ces problèmes sont liés aux solutions adoptées, et la communauté de concepteurs devrait pouvoir partager et réutiliser ces résultats dans le cycle de vie du scénario d'apprentissage. Le pattern est considéré comme une méthode très utile pour capitaliser des résultats et partager des solutions [Randriamalaka-2006]
Méthodes	Une méthode pédagogique décrit le moyen adopté par l'enseignant pour favoriser l'apprentissage et atteindre son objectif. Le choix d'une méthode pédagogique se fait en fonction du profil de l'apprenant, de la complexité des savoirs à transmettre, des contraintes et des moyens (techniques, financiers, organisationnels, pédagogiques) [Sahfogang-2009]. On distingue cinq méthodes selon ce mémoire <ol style="list-style-type: none"> 1. la méthode magistrale (transmissive, passive) où l'enseignant transmet un contenu structuré et ses connaissances sous forme d'exposé 2. la méthode démonstrative où l'enseignant détermine le chemin pédagogique aux apprenants qui acquièrent un savoir-faire par simple imitation 3. la méthode interrogative (maïeutique) où l'apprenant possède des éléments de connaissance, et est incité à formuler ce qu'il sait, ce qu'il pense ou bien ce qu'il représente 4. la méthode de découverte (active) où l'enseignant crée un scénario pédagogique avec du matériel qui permet d'utiliser les essais, les erreurs, le tâtonnement pour apprendre, en suivant l'enchaînement : fait faire à l'apprenant, fait dire à l'apprenant puis l'enseignant reformule 5. la méthode expérimentale où l'enseignant incite à la formalisation du savoir-faire par l'apprenant qui est le vrai producteur du savoir qu'il partage et réélabore avec d'autres. Le savoir est acquis par l'apprenant dans et par l'action en règle générale dans un projet réel
Méthodologies	Tout problème relié à une situation est résolu par l'utilisation et l'exploitation planifiée de ressources éducatives disponibles. [Marton-1994]
Bases de traces	Un système à base des traces se charge d'analyser les traces d'utilisation d'un environnement d'apprentissage en visant de : <ul style="list-style-type: none"> - Amélioration du dispositif (Tableaux de bord lors de la phase de conception, exploitation statistiques en retour d'usage). Destinataire = AUTEUR - Scan des activités apprenant pour suivi, pilotage et évaluation de l'apprenant. Destinataire = INSTRUCTEUR - Auto-localisation de l'apprenant dans sa découverte (du cas, de son parcours) et sa construction cognitive (connaissances validées à quel niveau). Destinataire = APPRENANT - Assurer la sauvegarde et reprise de sessions en toute sécurité. Destinataire = INFORMATICIEN
Environnements génériques	Un environnement pédagogique générique permet de la création de contenus pédagogiques (<i>authoring</i>) flexible et la réutilisation des scénarios d'apprentissage pour plusieurs disciplines. Par exemple [Benmohamed-2007]
Gestion de session d'apprentissage	Un système de gestion d'apprentissage est un système logiciel web développé pour accompagner et gérer le parcours pédagogique des personnes impliquées dans le processus d'apprentissage, comme le LMS.

Tableau 1.4 : Composantes pédagogiques utilisées dans un environnement du e-learning.

1.6 Effets des nouvelles technologies sur les modes d'enseignement

Dans cette section nous introduisons les modes d'enseignement adoptés dans l'enseignement supérieur, puis nous présentons les effets des TIC sur ces modes.

Jean Houssaye a proposé « le triangle pédagogique » comme modèle de compréhension des principes de la pédagogie (Figure 1.5) [Houssaye-1993]. Celui-ci aide à comparer, rapprocher diverses situations pédagogiques qui toutes s'articulent autour de trois éléments

(l'apprenant, la connaissance et l'enseignant), dont deux sont prédominants sur le troisième. En effet, l'intégration des TIC dans les trois actions décrites dans ce triangle (enseigner, former, apprendre) a largement affecté les moyens et les méthodes pour les réaliser. Ce modèle présente cependant deux défauts majeurs :

- il n'intègre pas le **concepteur** qui, nous le verrons, est amené à jouer un rôle décisif dans le processus-même d'apprentissage (c'est en particulier sur lui que se refermeront les boucles d'usage) ;
- il ne prend pas en compte le « **collectif** » : la notion de groupe d'apprenants (entité à distinguer de chacun) est ignorée.

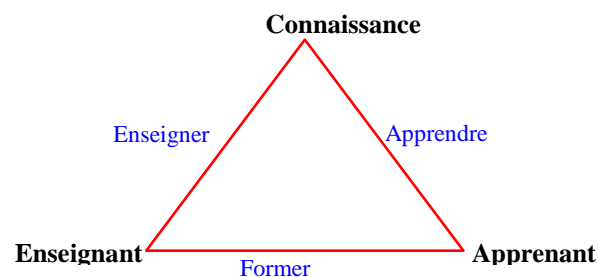


Figure 1.5 : Le triangle pédagogique.

En conséquence, nous proposons donc de mettre à jour ce triangle (voir Figure 1.6) en considérant que la cible du processus pédagogique peut également être un groupe d'apprenants et que le concepteur de contenus pédagogiques joue un rôle pivot pour :

- élaborer le contenu pédagogique et donc la connaissance à délivrer aux apprenants ;
- superviser l'élaboration du contenu didactique propagé par l'enseignant ;
- fournir les moyens et méthodes pour évaluer la performance d'apprentissage de groupe d'apprenants.

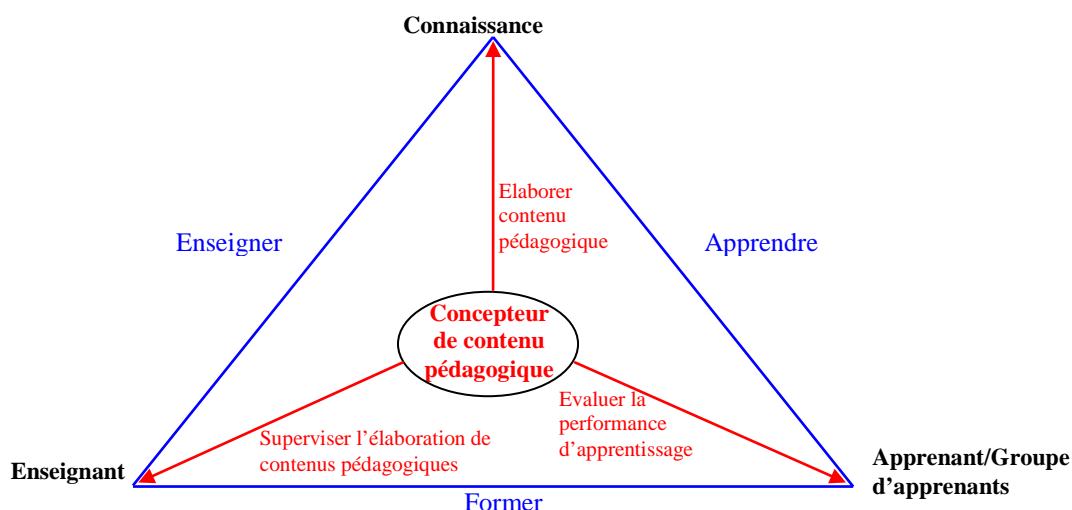


Figure 1.6 : Le triangle pédagogique avec le rôle central du concepteur.

1.6.1 Les modes d'enseignement

Cette section introduit les différents modes pédagogiques rencontrés traditionnellement dans l'enseignement supérieur.

Les apprenants vont successivement aborder des domaines ou disciplines dans lesquels ils sont amenés à construire des connaissances et des compétences que l'on va contrôler, valider et par moment certifier. Les modes pédagogiques, se résumant pendant des dizaines d'années à la trilogie « Cours-TP-TD », se sont enrichis de mises en situation individuelles et collectives plus ou moins simulées (**mode projet**), puis de mise en situation professionnelle réelle via l'émergence du **mode « formation par alternance »** donnant enfin du sens à cette juxtaposition peu communicante entre Stage et Formation dite académique. Rappelons le sens et les variétés de chacun de ces modes ([Benmohamed-2007]):

1.6.1.1 Le cours

Dispensé souvent par unité de 2 heures, il est dédié plutôt à la découverte conceptuelle, à la présentation de principes, lois, modèles, outils, méthodologies.... Par nature non interactif, l'apprenant y est réduit le plus souvent à un rôle de spectateur : sa seule intervention possible est dans l'expression captée de son incompréhension de tel ou tel point de discours. Pour enrichir un cours, le tuteur peut proposer aux apprenants de consulter quelques références, dans une bibliothèque dédiée, liées au domaine. Cela peut aider à : introduire le sujet du cours, si proposé avant le cours, ou bien clarifier les thèmes abordés pendant le cours, si proposé après le cours. Ce choix reste optionnel et dépend de la motivation de chaque apprenant. En résumé ni l'action, ni l'émotion (dépend du talent théâtral de l'enseignant) ne

sont présents. On ne peut donc pas parler de « Pédagogie ». Le cours ne doit donc sa subsistance qu'à son faible coût.

1.6.1.2 Les Travaux Dirigés (TD)

Souvent dispensés par unités de 2 heures, ils sont à mi-chemin entre la théorie et la pratique. Le but est de proposer des exemples et exercices de base où les apprenants peuvent illustrer, comprendre et exercer des concepts, méthodes, modèles ou outils. L'apprenant est beaucoup plus acteur. Par contre le travail reste de nature théorique, on ne passe pas encore au stade de la réalisation. Dans ce mode d'enseignement, le tuteur a un temps limité pour interagir avec tous les apprenants et leurs demandes, achèvement et performance. Les apprenants ont également un temps limité pour poser toute les questions potentielles, interagir avec les directifs du tuteur et profiter de tout conseil proposé.

1.6.1.3 Les Travaux Pratiques (TP)

Unités en général de 4 heures, ils offrent aux apprenants des dispositifs concrets où ils peuvent mettre en application leurs connaissances. La mission des apprenants consiste à réaliser une tâche vérifiable par des résultats. Pour y parvenir, des méthodes et outils sont souvent donnés. Les apprenants doivent instancier les modèles ou paramétrer des dispositifs pour atteindre les objectifs proposés. Ils sont donc pour partie dirigés, tout en bénéficiant de degrés de liberté. L'enseignant n'intervient alors que sur sollicitation directe ou lorsqu'il constate une incompréhension ou mauvaise voie suivie.

1.6.1.4 Le projet

D'une durée de plusieurs semaines, il concerne une activité collective proposée à un groupe d'apprenants (de 4 à 8). Il donne lieu à 2 types de séances : celles programmées en présence du tuteur (1 à 2 séance de 4h par semaine) et celles spontanément organisées par le groupe d'apprenants pour coordonner, distribuer les tâches, échanger, décider, préparer le rendu de compte (recette)...par séances Il s'agit soit de répondre à une vraie étude (inédite), soit de redécouvrir un cas inspiré du réel et contextualisé par celui qui l'a vécu (le tuteur, qui joue donc aussi ici le rôle de témoin). Le projet représente une pratique de pédagogie active qui produit des apprentissages à travers une réalisation concrète. Il est semblable à une entreprise qui permet à un collectif d'apprenants de réaliser une production concrète socialisable, en intégrant des savoirs nouveaux [Hubert-1999].

Selon [Bouquet et al.-1997], la pédagogie par projet est un processus d'apprentissage qui met un groupe de personnes (apprenants ou stagiaires, enfants ou adultes, en formation ou en loisir) en situation :

1. d'exprimer des envies, des questions, des besoins, des manques, des ambitions ;
2. de rechercher les moyens d'y répondre ;
3. de planifier collectivement la mise en œuvre du projet et de le vivre.

Le projet reste global. Il a une dimension collective fondamentale et ne donne pas obligatoirement accès à des dispositifs techniques.

1.6.1.5 Le stage

Il s'agit le plus souvent d'une immersion dans le milieu qui sera celui dans lequel l'apprenant exercera son métier. La dimension culturelle (de l'entreprise d'accueil, de l'entreprise en général et du secteur concerné) est ici fondamentale. Elle éclaire et donne du sens aux connaissances construites par l'apprenant en les transformant en compétences.

Types de stages :

- Stage d'observation : ouvert aux plus jeunes et souvent très court (entre quelques jours et un mois), ce stage laisse peu de temps pour exécuter une véritable mission. C'est un premier stage et l'occasion de comprendre les rouages du fonctionnement professionnel (déniaiser), et donc de sortir de l'état scolaire ;
- Stage d'application : d'une durée de plusieurs mois, ce type de stages donne du sens et donc développe les acquis d'un apprenant déjà avancé dans un cursus ;
- Stage de fin d'études : ce stage a lieu en fin de cursus, il détermine en partie le type d'emploi auquel le stagiaire peut prétendre et débouche parfois sur un premier emploi. De plus en plus d'entreprises se servent de ce type de stages comme d'une période d'essai ;
- Stage en alternance : C'est une pédagogie mille-feuille, alternant périodes académiques et mises en application en milieu professionnel. Le contenu de la formation doit mieux « coller » aux besoins professionnels (qualité) et l'apprenant voit

sa motivation d'autant plus s'accroître que ses connaissances sont crédibilisées par une mise en application efficace et reconnue.

- Année de césure : les apprenants en école d'ingénieurs ont la possibilité de faire un *break* d'une année durant leurs études. Une année est souvent utilisée pour effectuer un stage long en entreprise. C'est l'occasion d'assortir son diplôme d'un an d'expérience professionnelle.

1.6.2 Impact des TIC sur l'évolution des modes d'enseignement

L'apparition des TIC dans l'enseignement a modifié puis transformé les modes pédagogiques en une vingtaine d'années. Une étude menée dans [Ben Youssef-2010], sur l'impact des TIC sur le processus d'apprentissage dans l'enseignement supérieur, rassemble et analyse les résultats de plusieurs études menées avec le même objectif. Les points de vue paradoxaux sur l'impact positif ou négatif des TIC dans le processus d'apprentissage, (voir Tableau 1.5), confirment que cette technologie a des effets sur la performance des apprenants, et des effets donc sur les modes pédagogiques des Enseignants.

Sujet de débat	Avis positif	Avis négatif
Nouvelle dimension à ajouter au processus d'apprentissage	Les TIC constituent une entrée dans le processus d'apprentissage qui doit aider à produire une meilleure sortie d'apprentissage pour les apprenants.	L'introduction de l'apprentissage basé sur les TIC peut se traduire par une réaffectation des capitaux au profit de TIC, et éventuellement remplacer des matériels pédagogiques efficaces.
Possibilité d'indépendance des apprenants et d'auto-apprentissage	L'utilisation de cette technologie peut améliorer l'apprentissage en réduisant la dépendance sur des tuteurs de différentes qualités.	L'apprentissage basé sur les TIC réduit la créativité des apprenants. TIC tend à permettre d'agir seulement d'une manière prédéfinie avec des possibilités interactives limitées. Cela peut réduire la capacité des apprenants en termes de résolution des problèmes et de réflexion créative dans un cadre de schème prédéterminé, mais pas leur capacité de trouver leurs propres solutions créatives.
Liberté de l'accès au processus d'apprentissage	L'utilisation de cette technologie peut améliorer l'apprentissage en fournissant l'apprentissage chez soi tout au long du jour.	L'accès internet chez soi peut être une source de distraction à cause de <i>chat rooms</i> et jeux en ligne, ce qui conduit à réduire le temps d'apprentissage.
Disponibilité des outils et des ressources d'apprentissage	L'exploitation des possibilités énormes d'acquisition des informations par les apprenants pour des fins scolaires.	L'impact des TIC sur l'apprentissage de l'apprenant dépend fortement sur ses usages spécifiques.

Tableau 1.5 : Avis paradoxaux sur l'impact des TIC sur la performance des apprenants dans l'enseignement supérieur²⁰.

²⁰ Résultats Synthétisés de [Ben Youssef-2010].

1.6.2.1 Évolution des cours

Le fait que l'apprenant est réduit à un rôle de spectateur, pendant le cours et avec des interventions très limitées, a abouti à des tentatives d'habillages technologiques pour adoucir ce défaut sans le faire disparaître :

- les **tableaux interactifs** : les commentaires inscrits en superposition des *transparents* par le formateur sont automatiquement numérisés et communiqués aux apprenants ;
- les **boîtiers de vote** : donnent la possibilité d'interpeller l'auditoire, de le sonder pour vérifier la bonne acquisition des concepts par chaque apprenant, et de mesurer instantanément l'efficacité du cours afin de l'adapter en temps réel. Ces systèmes de vote interactif sont donc des outils adaptés aussi bien en formation initiale que professionnelle. Ils sont supportés par plusieurs fournisseurs (BAF²¹, POWERVOTE²², EASYVOTE²³).

Un rapport sur l'impact des TIC sur l'enseignement supérieur (Enseignement, recherche et services) a résumé la valeur ajoutée que l'on pouvait en attendre [Balasubramanian et al.-2009] :

- diversité des medias ;
- accès (en mode guidé ou libre) via des bibliothèques virtuelles à des documentations et ressources électroniques ;
- accès immédiat à de l'information grâce à des services de recherche, de filtrage, de dictionnaires électroniques.

Le Tableau 1.6 la valeur ajoutée des TIC pour l'enseignement « magistral ».

Problématiques liées à un cours traditionnel	Conséquences pédagogiques	Solutions apportées par les TIC
Incertitude du formateur sur l'efficacité de son cours	- La connaissance délivrée est inadéquate, insuffisante ou ambiguë. - Risques d'échec des modes d'enseignement situés en aval (TD et TP, projet...).	Possibilité d'interpellation de l'auditoire via des boîtiers de vote, ou de sondage via un site web, afin de vérifier la bonne acquisition des concepts pour chaque apprenant
Difficultés à trouver les bonnes illustrations de certains concepts ou idées	Incertitude sur l'a qualité et la pérennité des connaissances construites par l'apprenant par	Utilisation de vidéoprojecteur, d'enregistrements audio/vidéo, d'animation multimédia...

²¹ Voir : <http://www.cbaf.fr/>

²² Voir : <http://www.powervote.com/fr/>

²³ Voir <http://www.easyvote.fr/>

	rapport à ses besoins	
Absence épisodique d'un apprenant	Connaissances incomplètes (lacunes).	Utilisation de tableaux interactifs afin d'enregistrer des commentaires contextuels
Construction cognitive déficiente chez l'apprenant	Connaissance incomplète ou possiblement erronée	Utilisation de cas illustrés par du multimédia afin de mieux acrer les savoirs par une démarche « Bottom-up »
Manque d'approfondissement de certains concepts	Connaissance incomplète entraînant des difficultés potentielles dans les TD ou TP	Accès à des sites web dédiés disposant de ressources et de fonctionnalités de recherche

Tableau 1.6 : Apports des TIC pour améliorer les cours.

Rappelons que ce discours se situe dans un contexte d'enseignement en présentiel. Dans un contexte d'enseignement à distance l'impact des TIC est encore plus visible :

- le tuteur donne un cours via vidéoconférence ou des enregistrements vidéo consultables à tout moment et de n'importe ;
- les apprenants ont la possibilité d'interagir via audio/vidéo et chat en temps réel ;
- les matériels textuels/graphiques, générés par le(s) tuteur(s) sont mis à la disposition des apprenants via un système de gestion d'apprentissage (LMS) qui, à son tour, gère les différentes activités d'apprentissage ;
- les bibliothèques numériques complètent le panorama en fournissant des ressources numériques de tout type (textuel/graphique, audio, vidéo), et des fonctionnalités simplifiant leur utilisation (recherche, dictionnaires, etc.).

1.6.2.2 Les TD

Les Travaux Dirigés (TD) ont pour objectif de donner du sens et d'approfondir les connaissances théoriques construites dans les cours, en introduisant des notions nouvelles et complémentaires (on fait grandir l'arbre du savoir). Selon [Ney et al.-2004], les problématiques rencontrées dans un TD traditionnel sont :

- le TD traditionnel se réduit trop souvent à un "cours de rattrapage" soit parce que le cours magistral a été incompris (fréquent compte tenu de la pauvreté pédagogique de ce mode), soit parce qu'il faut « préparer les examens (on assiste là à un détournement d'objectif) ;
- les problèmes posés dans un TD traditionnel restent trop souvent académiques ne justifiant pas l'utilité des connaissances construites. Ils sont donc peu propices à éveiller le questionnement chez les apprenants.

Selon [Ney et al.-2004] l'utilisation des TICE a obligé à repenser la pédagogie en TD vers un apprentissage actif soutenu par des TICE. Ce qui nous intéresse dans cette approche est l'introduction des TIC comme un outil essentiel pour passer d'un TD traditionnel à un TD actuel en apprentissage actif, en faisant des activités de recherche d'informations sur un site web (par exemple, MathSV²⁴) et de résolution de problème passant par l'analyse de données acquises par expérience simulée.

Le temps limité d'une session de TD (et le nombre important d'apprenants) est une contrainte : le tuteur fait souvent du « point à point » au détriment d'activités collectives organisées.. D'où l'intérêt de créer des forums électroniques. Il s'agit de bulletins en ligne où les apprenants inscrits peuvent discuter d'un sujet lié à une matière, entre eux et avec le tuteur. La discussion est une communication asynchrone entre les différents intervenants au sein de laquelle ils peuvent poser des questions, recevoir des réponses ou simplement lire ce qui est écrit par les autres.

Dans un contexte **d'enseignement à distance**, la Communication Médiatisée par Ordinateur « CMC » (traduction de l'anglais : *Computer Mediated Communication*) joue un rôle important particulièrement lors des TD. Dans cet environnement où les contacts directs entre les différents intervenants sont rares, la discussion en ligne y remédie. Néanmoins, le tuteur doit faire plus d'effort pour suivre les activités de ses apprenants dans un TD en ligne afin d'intervenir/interagir à bon escient et évaluer avec pertinence leur apprentissage. Pour résoudre cette problématique, l'outil (TrAVis²⁵) proposé dans [May et al.-2011] a pour objectif la traçabilité, l'analyse et la visualisation des données récupérées des activités supportées par CMC. La nouveauté de cet outil est son utilisation par le tuteur et l'apprenant, chacun selon ses propres objectifs :

- pour le tuteur, TrAVis fournit trois outils de surveillance en temps réel d'un apprenant ou d'un groupe d'apprenants (voir Figure 1.7) pendant le cours et les activités de communication. TrAVis assiste le tuteur pour évaluer ses apprenants selon plusieurs aspects (sociaux, cognitifs, comportementaux...);
- pour l'apprenant, TrAVis donne une idée sur ses activités en suivant ses données, et lui permettant de s'auto-évaluer.

²⁴ Voir le site : <http://spiral.univ-lyon1.fr/mathsv/>

²⁵ TrAVis : Tracking Data Analysis and Visualization.

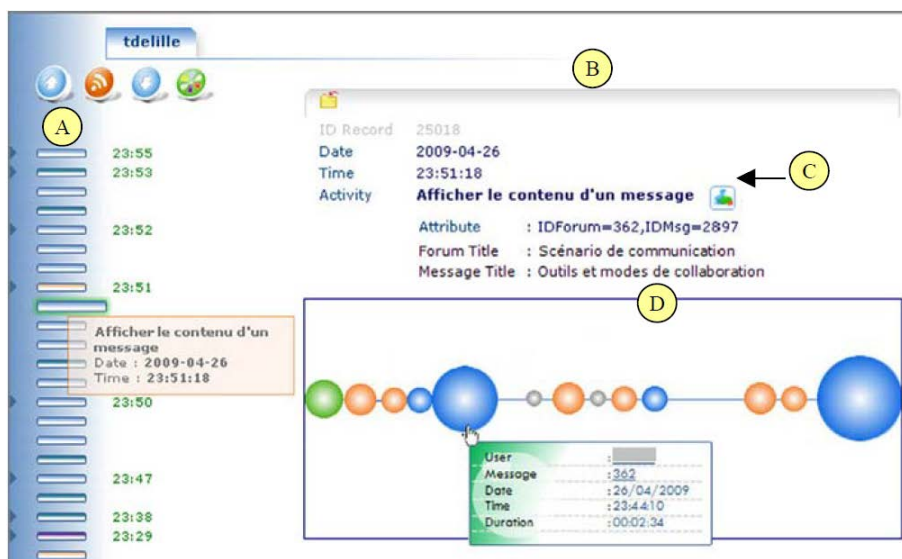


Figure 1.7 : Liste d'activités d'utilisateur affichée dans la vue de Time Machine de TrAVis « Tdelille »²⁶.

- A- Liste d'activités.
- B- Informations détaillées sur l'activité sélectionnée.
- C- Menu supplémentaire permettant de visualiser d'autres activités liées à l'activité en cours.
- D- Qui d'autre lit le même message lu par Tdelille. Chaque sphère représente une activité d'affichage de message, son diamètre est proportionnel au temps passé par chaque utilisateur en lisant le message, la distance entre les sphères représente le temps entre deux lectures.

1.6.2.3 Les TP

Afin de prendre conscience et mesurer l'impact opérationnel de leurs connaissances, les apprenants, à travers des Travaux Pratiques (TP), réalisent des expériences permettant de vérifier et compléter les connaissances dispensées et développées dans les cours théoriques²⁷ et les TD. La réalisation de TP passe par les étapes suivantes :

- introduction du sujet de TP, les objectifs de TP et les moyens à utiliser ;
- adoption d'une méthodologie de réalisation des tâches (à proposer par l'instructeur ou à laisser à l'initiative des apprenants) ;
- réalisation des tâches et vérification expérimentale.

La littérature scientifique classe les motivations qui ont provoqué l'utilisation des TIC dans les TP, surtout dans les disciplines techniques et scientifiques, en deux catégories :

1- problématiques **organisationnelles** : l'acquisition de « savoir-faire » à travers les TP traditionnels est limitée par les facteurs suivants :

- le nombre croissant d'apprenants, qui limite la capacité de l'instructeur à suivre et répondre à toutes les requêtes ;

²⁶ Extrait de [May et al.-2011].

²⁷ Définition extrait du site web de « La faculté des sciences et technologie » : <http://sciences-tech.u-pec.fr/foire-aux-questions-262117.kjsp?RH=SCTECHNO>

- les équipements matériels, à jour, ne sont pas toujours abordables par l'établissement enseignant (coût d'investissement, coût élevé de maintenance et des techniciens chargés de faire fonctionner et maintenir la plate-forme [Ishutkina et al.-2004, Kikuchi et al.-2004]. Il est commun que plusieurs établissements partagent des ressources matérielles coûteuses, ce qui rend difficile et handicapant l'obligation de présence en un lieu donné à un moment donné de tout le monde [Guimaraes et al.-2000, Kikuchi et al.-2004]. A cela s'ajoute le ;

2- problématiques **d'achèvement** : cet aspect est lié directement à la réalisation d'une tâche par l'apprenant, surtout dans la phase de récupération et d'analyse des données. Par exemple, les apprenants dans une expérimentation en chimie doivent noter régulièrement la température d'un mélange chimique afin d'ajuster les autres matériaux à ajouter. Dans ce cas surviennent des erreurs inhérentes au relevé de la température, ou des quantités de matériaux à ajouter. À la fin des années soixante-dix, le terme « Laboratoire Assisté par Ordinateur » [Kaiser-1978] a émergé et ouvert un nouvel horizon dans le domaine des techniques de formation pratique au sein de plate-forme de TP. Pour la première fois, dans un contexte éducatif, les manipulations techniques sont devenues automatisables grâce à la médiation de l'ordinateur. Les utilisateurs ont pu consigner, enregistrer, extraire, visualiser et exploiter automatiquement l'ensemble des données [Consonni-2001, Williams et al.-2004]. On a assisté ainsi à l'émergence de l'Expérimentation Assistée par Ordinateur (ExAO), en tant que sous-discipline de l'EAO [Gravier-2007].

Plus finement, les TIC ont apporté des solutions, présentées dans le Tableau 1.7 (mais également en §2.2, §2.3, §2.4, §2.5), pour améliorer les TP traditionnels et les faire évoluer vers des TP « électroniques » (e-TP) :

Problématique de TP traditionnel	Conséquences	Solution apportée par les TIC
Nombre élevé d'apprenants	<ul style="list-style-type: none"> - Impossibilité pour l'instructeur de suivre en permanence les activités de tous les apprenants ; - Chaque apprenant passe moins de temps avec le dispositif de TP partagé ce qui limite la consolidation des connaissances. 	<ul style="list-style-type: none"> - Les systèmes de gestion d'apprentissage (LMS) fournissent des outils de suivi des activités des apprenants (ex : voir Figure 1.7) permettant aux instructeurs de mieux les gérer et d'interagir au bon moment ; - la télémanipulation d'un dispositif au sein d'une plate-forme de TP distant « RLab » (voir Figure 2.5) donne à l'apprenant plus de liberté de réaliser des tâches, sans les contraintes présentes et en optimisant l'utilisation des ressources.

Difficultés à fournir des équipements et du matériel modernes	Manque de crédibilité de la formation effectuée sur du matériel « dépassé » développant des connaissances lacunaires et de moindre qualité (n'est plus en adéquation avec les besoins du marché)	- Partage de ressources entre plusieurs établissements via un réseau permettant la télémanipulation des dispositifs technologiques et réduisant l'impact de l'achat de nouveaux équipements (ex : AIP-Primeca-RAO) ; - Proposition de simulateurs des systèmes réels soit sous forme de logiciels indépendants, soit au sein d'une plate-forme de TP virtuel (voire Figure 2.4). Un simulateur est facilement diffusable, actualisable, personnalisable et sans risques techniques. On peut s'affranchir du temps réel en comparant un simulateur avec un dispositif qui prend un temps considérable pour arriver à un régime établi
Difficultés de contrôler le dispositif	- L'attention de l'apprenant est mobilisée (stress) par des considérations de risques techniques souvent loin des objectifs pédagogiques ; - développement d'un « savoir piloter » procédural au détriment d'une maîtrise globale du processus ; - échec dans l'élaboration d'une tâche privant du retour d'expérience ; - temps important consacré pour préparer les actions, en attendre les résultats afin d'extraire les informations utiles, avec risque de commettre des erreurs de relevé ou d'interprétation (charge mentale).	- La connexion du dispositif à l'ordinateur, via un logiciel dédié, fournit des instructions, des précautions à prendre en particulier par des tests préalables sur un simulateur (avant de l'appliquer sur le dispositif) ; - la collecte, la représentation et le traitement des données peut être automatisée et aidée par des outils d'analyse et de visualisation
Temps trop important consacré à la configuration et à la maintenance des équipements	- Réduction du temps « pédagogique » ; - asservissement des séances à la présence d'un technicien (en plus du formateur)	- Maintenance du dispositif effectuée via une configuration spécifique virtualisant ou assurant à distance le rôle du technicien (sauf en cas de risque ou d'incident grave).

Tableau 1.7 : Apports des TIC pour améliorer les TP.

1.6.2.4 Les projets

Leur émergence est plus récente en tant que type d'enseignement (à part entière bon nombre de formations en ont encore une vision déformée). Le projet s'identifie par un travail collectif et actif avec définition d'objectifs, prise de risque et de décisions pour répondre à un cahier des charges « client ». À la différence des autres modes d'enseignement :

- ils ne répondent plus à une organisation méthodologique s'appuyant sur un enchaînement de questions qui visent un but précis afin d'amener l'apprenant à un constat ou à une conclusion « préprogrammée ». Au contraire, le projet favorise l'**initiative** et le développement de méthodes et d'organisation de l'apprenant ;
- ils favorisent une **activité** permanente collective répartie en « missions » ou tâches synchronisées (l'action est un pilier principal de la pédagogie). L'apprenant n'est

plus passif devant l'organisation de son travail, il est obligé de s'y investir s'il veut mener à bien son projet, alors que cette organisation lui est imposée lors des TP [Amat-2003] ;

- ils valorisent les résultats obtenus par leur **défense** devant un jury d'experts (recette) ;
- ils se rapprochent du rôle que les apprenants auront à tenir dans leur futur métier

Le Tableau 1.8 résume les apports des TIC à ce mode pédagogique.

Situation	Exigences	Solution apportée par les TIC
Un projet se déroule sur un laps de temps de plusieurs semaines, les idées des apprenants se développent et nécessitent des échanges d'information	Négociation et communication synchrones/asynchrones entre les membres du groupe	- Communication via chat, audio et vidéo, courriel ; - Négociation via des forums électroniques. - Conduite de Brainstormings « informatisés »
Les apprenants s'accordent autour d'une même méthodologie de conception/réalisation	Concrétiser des schémas décrivant le système à modéliser et détaillant les étapes de la méthodologie adoptée.	- Outillage (informatique) projet (Gantt, AMDEC, Aide à la décision, PDCA...) - Outils de modélisation comme UML et SysML permettant de réaliser des modèles
Nécessité de prendre connaissance de projets similaires, de disposer d'informations techniques (faisabilité)	Il faut accéder à ces informations	- Bibliothèques numériques disposant de ressources techniques + Recherche sur Internet permettant tout au long du projet d'accéder aux informations souhaitées - Discussions dans les forums électroniques ou les communautés pour profiter de l'expérience des « autres ».
Les apprenants ont terminé leur travail et veulent le valider	Les apprenants doivent tester la solution adoptée	Dans les disciplines techniques, les TIC fournissent des simulateurs permettant de tester une solution avant de la valider. Exemple : le logiciel « Unity ²⁸ » disponible dans le domaine de l'automatisation.

Tableau 1.8 : Apports des TIC pour améliorer la pédagogie du projet.

Exemple de l'apport des TIC en projet : Pour illustrer les apports des TIC dans la pédagogie de projet, voici un l'exemple d'un projet d'Automatisation vécu par les apprenants de quatrième année à l'INSA de Lyon, au département Génie Industriel. Ce projet se situe dans un cadre industriel classique et aborde une application typique des automatismes dans des contextes définis par l'instructeur (industrie agro-alimentaire, industriel manufacturier, chaîne de fabrication, etc.). Ce projet s'appuie sur l'environnement logiciel de développement de programmes automatiques « Unity » commercialisé par la société Schneider Electric. Ce logiciel,

²⁸ Unity Application Generator: <http://www2.schneider-electric.com/sites/corporate/en/customers/alliance/system-and-architectures/system-platform-and-engineering-tools/uag-unity-application-generator.page>

outre les fonctions d'écriture de programmes automate, offre la possibilité de simuler le système automatisé cible et donc de tester et déboguer ses programmes avant d'être exécutés sur le vrai système automatisé. Les sessions du projet étant interrompues par les vacances d'hiver, les membres du groupe ont continué de communiquer via courriel. Pour l'écriture des rapports de conception et de réalisation, avec la contrainte d'utiliser le standard de représentation SysML, un logiciel dédié (Artisan Studio²⁹) a été utilisé. A la dernière séance du projet, la démonstration du programme réalisé s'est déroulée à travers l'outil de simulation précédent afin de tester toutes les situations envisagées dans le cahier des charges du projet ainsi que des situations anormales qui pourraient être dangereuses dans la réalité avec le vrai système automatisé. Cet outil logiciel a donc permis, d'une part de gagner du temps entre les tests car il n'était pas nécessaire de réinitialiser physiquement la machine et de vérifier de manière exhaustive que les contraintes de sécurité ont bien été respectées et que toutes les fonctions élaborées étaient conformes au cahier des charges.

1.6.2.5 Les stages

Quelque soit le type de stage, le stagiaire se trouve dans une situation où il est mené à concrétiser ses compétences scientifiques et techniques dans un contexte professionnel. La particularité de cette situation est l'effort individuel du stagiaire, même dans un environnement de travail collectif, pour achever les tâches lui adressées sous la direction d'un encadrant.

Dans les différents types de stages précédemment présentés (cf. §1.6.1.5), des points communs se retrouvent en termes d'activités à faire par le stagiaire ainsi que les exigences y liées pour simplifier leur réalisation, ce sont :

- la communication que le stagiaire doit faire avec son encadrant ainsi qu'avec le nouveau milieu professionnel pour s'informer, discuter, échanger des informations, etc. ;
- la recherche des ressources liées au sujet de stage (document, notices, applications, etc.) ;
- la manipulation des données/informations dans le but de les analyser et extraire des résultats à exploiter dans un cadre académique et /ou professionnel ;

²⁹ Voir sur le site : <http://www.atego.com/products/artisan-studio/>

- l'évaluation le travail du stagiaire par son encadrant afin de mesurer la qualité de son travail par rapport aux objectifs du stage.

Si la diversité des domaines techniques et scientifiques, ainsi que la pluridisciplinarité dans chacun, imposent des exigences particulières sur le déroulement des différents stages, les activités générales communes entre eux nous aident à recenser les apports des TIC pour améliorer la pédagogie de stages (voir Tableau 1.9).

Activité	Exigences	Solution apportée par les TIC
Recherche	Trouver des ressources liées à un sujet, projets, application, etc.	- Recherche sur l'internet (particulièrement des ressources spécialisées de type Google Scholar, Wiley, ACM, ...) ; - bibliothèques numériques (par exemple : Service Commun de la Documentation Doc'INSA ³⁰).
Communication	Échanger des informations avec l'encadrant et les collègues de l'équipe.	- Communications synchrones (<i>chat</i> ou visioconférence) ; communications asynchrones (courriel ou forums électroniques).
Manipulation des informations	Faire des statistiques, analyser des données, extraire des résultats.	- Logiciel permettant de faire de statistiques et des courbes illustratifs (ex : « Epi Info » ³¹ dans le domaine d'épidémiologie, « gretl » ³² dans le domaine d'économétrie) ; logiciels permettant d'analyser les données enregistrées ou récupérer d'une expérimentation (ex : Matlab)
Évaluation du travail	Évaluer le travail du stagiaire	- Les tableaux interactifs permettent au stagiaire de présenter régulièrement son travail via vidéoprojecteur à encadrant ; - les simulateurs de systèmes réels permettent au stagiaire puis à l'encadrant de tester les solutions élaborées.

Tableau 1.9 : Apports des TIC pour améliorer la pédagogie du stage.

1.6.2.6 Emergence d'un nouveau mode : les Learning Games

Les TIC ont fait évoluer plus ou moins fortement les modes pédagogiques précédents. Mais ils ont surtout provoqué l'émergence de ce qu'on peut appeler un nouveau mode : les *Learning Games (LG)* (aussi connu comme *Serious Games* « SG »). Un **LG** est un environnement d'apprentissage dans lequel le formateur et les apprenants (en tant qu'individus et groupe) prennent les rôles d'acteurs. Cet environnement est dédié (construit autour d'un cas) et informatisé. Il combine une intention sérieuse, de type pédagogique, informative, communicationnelle, marketing, idéologique ou d'entraînement avec des ressorts ludiques. La vocation d'un **LG** est d'inviter l'utilisateur à interagir avec une application informatique [Alvarez-2008] dans le but de construire, en le vivant, une ou plusieurs compétences. Formateur et apprenants ont une vision permanente de leur processus d'apprentissage grâce à l'exploitation de traces qui seront également utilisées pour un retour

³⁰ Voir : <http://scd.docinsa.insa-lyon.fr/>

³¹ **Epi Info** est un logiciel statistique du secteur publique. Voir : <http://wwwn.cdc.gov/epiinfo/>

³² **gretl** est un logiciel de statistiques Open Source dans le domaine d'économétrie. Voir : <http://gretl.sourceforge.net/>

qualité vers le concepteur du jeu. Les ressorts ludiques jouent le rôle de « déclencheur de motivation ». Les **LG** offrent un potentiel immense. Il reste encore à les apprivoiser pour en exploiter toute la richesse.

Plusieurs recherches sont menées dans l'objectif de proposer un modèle de conception et de production de **SG**. Dans [Marfisi et al.-2009], les auteurs ont abordé la problématique liée à la conception de **SG** dans le but éducatif qui ne prend pas en compte l'aspect ludique, ils ont donc proposé un modèle de chaîne de production pour concevoir et produire des **SG** certifiés pour leurs gains éducatifs et leur qualité ludique.

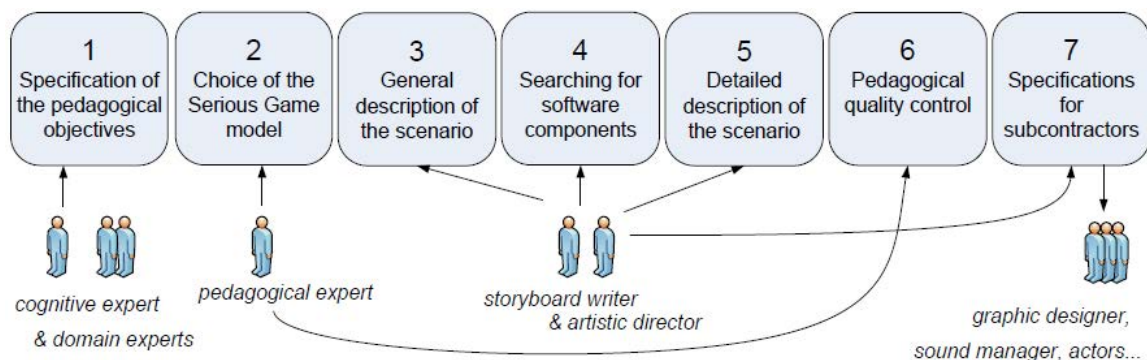


Figure 1.8 : Les 7 étapes de conception de Serious Games³³.

Dans la continuité de ces travaux, les auteurs présentent dans [Marfisi et al.-2010] le processus de conception d'un **SG** (voir Figure 1.8) et énumèrent les différents acteurs qui participent à la réalisation de **SG** (chef de projet, spécialiste cognitif, experts du domaine, écrivain, directeur artistique, expert pédagogique, développeurs, etc.), en proposant une méthode les aidant à collaborer d'une manière efficace. Finalement, les auteurs de [Tran et al.-2010] proposent, pour la création et développement de **SG**, un environnement qui prend en compte tous les aspects des travaux précédents.

Les **LGs** peuvent avoir différentes facettes comme affiché dans (Figure 1.9) [Marne et al.-2011]. Cependant, la création des environnements virtuels d'apprentissage à travers les jeux vidéo, inspirés des événements actuels de l'environnement réel, a permis de combiner la réalité avec la virtualité. Il est ainsi de plus en plus fréquent de trouver des environnements de contenu réels supportés par des visualisations 2D ou 3D permettant de mieux interagir avec des cas réels dans cet environnement. La « Réalité Augmenté » (RA) est un procédé qui consiste à ajouter un objet virtuel à une image filmée, en temps réel ou sur une vidéo. L'objectif étant de simuler la présence d'éléments divers dans notre environnement réel

³³ Extrait de [Marfisi et al.-2009].

[Thillou-2002]. Elle désigne les différentes méthodes qui permettent d'incruster de façon réaliste des objets virtuels dans une séquence d'images. Elle s'applique aussi bien à la perception visuelle (superposition d'images virtuelles aux images réelles) qu'aux perceptions proprioceptives comme les perceptions tactiles ou auditives. Ces applications sont multiples et touchent de plus en plus de domaines, tels que les jeux vidéo, l'éducation par le jeu (par exemple : Genomics Digital Lab³⁴), les industries (conception, *design*, maintenance, assemblage, pilotage, robotique et télérobotique, implantation, étude d'impact, etc.), business (Novicraft HRD³⁵), etc.

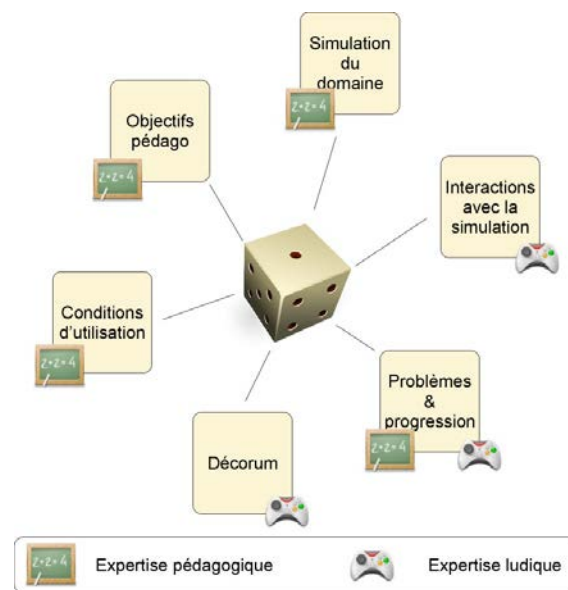


Figure 1.9 : Les 6 facettes du jeu sérieux³⁶.

Si la RA concerne l'ajout des éléments complètement réels (personnes, textures, etc.), l'ajout des éléments complètement virtuels (son, images, etc.) est concerné par la « Virtualité Augmentée » (VA). Les systèmes de VA sont couramment utilisés dans les simulateurs pour positionner l'utilisateur en situation d'interaction avec le monde virtuel. Un dialogue doit donc s'établir entre l'homme et la machine, c'est la boucle « perception, cognition, action » [Muratet-2010] (voir Figure 1.10).

³⁴ **Genomics Digital Lab** : une série basée sur un navigateur de jeux éducatifs, simulations et animation pour enseigner les étudiants des écoles secondaires sur la biologie, y compris la photosynthèse, la respiration, la transcription et la traduction. Voir : http://www.spongegdl.com/game_pages/gdl.cfm

³⁵ **Novicraft HRD** est un jeu sérieux pour soutenir les clients dans d'affaires dans l'excellence sociale, apprendre à construire une compréhension partagée avec des personnes différentes dans des contextes changeants. Voir : <http://www.teamingstream.com/>

³⁶ Extrait de [Marne et al.-2011].

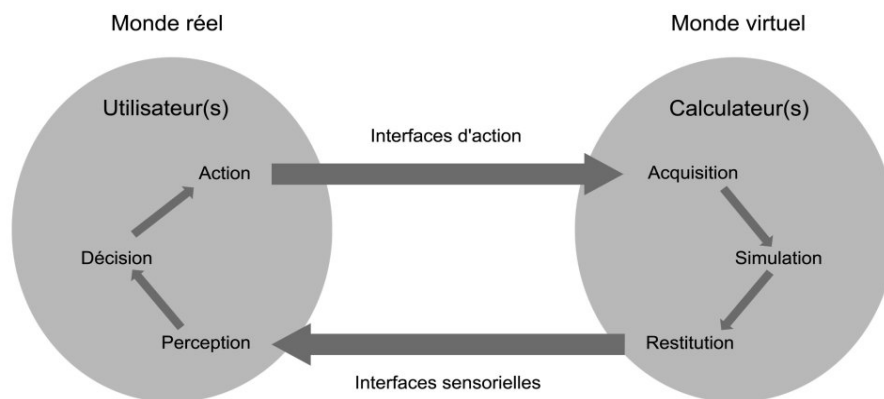


Figure 1.10 : Boucle « perception, cognition, action » en communication avec le monde virtuel³⁷.

RA et VA sont regroupés sous le terme de « Réalité Mixte » (RM) dont l'objectif est de fournir un environnement réel supporté par un moyen logiciel permettant aux intervenants d'y interagir (voir Figure 1.11). La RM représente donc un moyen intéressant pour délivrer de la connaissance autant pratique que théorique aux apprenants pour améliorer leur niveau scientifique. Actuellement, des efforts sont faits pour proposer une spécification pédagogique pour la réalité mixte dans les jeux d'apprentissage, comme dans [Orliac et al.-2011], ce qui ouvre une perspective plus favorable d'intégration de tels SG dans les processus de formation.

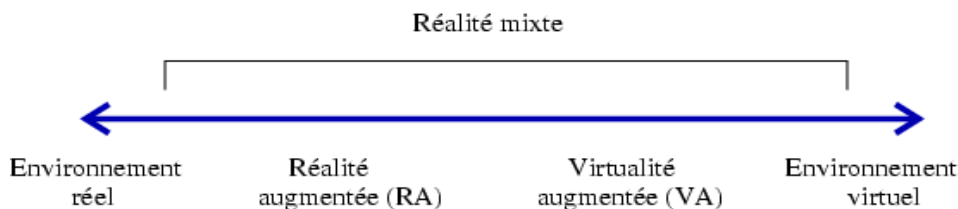


Figure 1.11 : Le continuum réel / virtuel³⁸.

1.7 Conclusion

De manière générale, l'intégration des TIC dans les différents modes d'enseignement présentés dans les sections précédentes (cours, TD, TP, projets, stages, etc.), a ouvert de nouveaux horizons de transfert et d'acquisition de connaissance. Des nouveaux outils ont été mis à la disposition de différents intervenants du processus éducatif (apprenants, tuteurs, instructeurs, chercheurs, etc.). Concernant les formateurs, il s'est cependant posé le problème de l'accompagnement du changement de la pratique de leur profession. Il en est malheureusement résulté une non utilisation de produits pédagogiques disponibles. Le travail

³⁷ Extrait de [Muratet-2010] p.26.

³⁸ Extrait de [Thillou-2002] p.16.

pour développer de nouvelles compétences pour utiliser ces nouveaux outils avec pertinence reste en grande partie encore à faire.

L'observation locale dans notre voisinage scientifique nous a révélé que qu'il y a une exploitation actuelle des TIC dans les différents modes d'enseignement. Nous avons également noté cette exploitation dans les e-TP nécessite encore des efforts à mener en ce qui concerne les outils à fournir et les compétences à construire. Cette observation a été soutenue par une enquête lancée auprès des enseignants chercheurs dans les disciplines techniques et scientifiques (détaillée dans §2.7.5), ce qui nous a aider à mieux identifier les problèmes liés aux e-TP et de préciser le domaine de notre participation.

2. Contexte de la recherche : les Travaux Pratiques

Introduction

Ce chapitre aborde plus précisément le contexte de notre recherche. Nous y présentons, à travers une étude bibliographique, les travaux pratiques dans l'enseignement supérieur et leur environnement type. Les STIC font évoluer lentement l'environnement des TP par rapport aux autres formes pédagogiques impactées depuis plusieurs années par les outils du e-learning.. La raison vient sans doute, dans le cadre des TP :

- de la nécessité de **reconcevoir les dispositifs** technologiques d'origine pour y intégrer de nouveaux modes d'exploration, de pilotage, de simulation et d'exploitation (données, mesures, contextes...). Ces adaptations nécessitent une forte ingénierie et sont donc longues et coûteuses, car chaque fois spécifiques de l'installation locale ;
- de l'intégration de dispositifs « de pointe » telle que la **réalité augmentée**, à plus grande valeur ajoutée que pour les autres modes pédagogiques. Il y a là une possibilité de pénétrer au cœur du système, d'en mieux comprendre les mécanismes, de mieux percevoir et diagnostiquer, d'intervenir naturellement et sans danger ;
- de la difficulté de greffer une pédagogie sur mesure, apportée par les STIC, à un dispositif le plus souvent **unique et partagé**. Il faut profiter de l'avènement des STIC pour mettre en place des TP « caméléon » susceptible de s'adapter à tous les types de formation (initiale, professionnelle), à toutes les origines (DUT, Ingénieurs, Maîtrises, MS, ...), à tous les niveaux (simple découverte ou sensibilisation, maîtrise, expertise...).

Nous préciserons dans ce qui suit ces problématiques et les pistes de résolution apportées par la communauté scientifique ainsi que celles proposées dans ce projet.

2.1 Objectifs généraux des TP

Dans cette section nous présentons les objectifs généraux de TP : consolidation ou construction de compétences (sens, savoir, savoir-faire...) La première caractéristique

essentielle des travaux pratiques (TP) est de favoriser un apprentissage **actif** autour des concepts abordés en « cours ».

La deuxième caractéristique est de marier l'approche **déductive** (héritage des cours et TD) et l'approche **inductive** (consolidation et construction à partir de l'observation et la découverte).

La caractéristique pédagogique absente (et qui sera apportée par les STIC) est la **composante émotionnelle**. Ayant découvert trop récemment le dispositif à « maîtriser », l'apprenant ne le domine pas. Il ne fait qu'exécuter un plan d'expérimentation proposé par un tuteur afin de « tenir dans les temps » et ne pas « prendre de risques matériels ou humains ».

Selon [Millar-2009], les objectifs de TP se résument en :

- aider les apprenants à développer la connaissance du monde réel et leur compréhension de quelques idées principales, des théories et des modèles que la science utilise pour l'expliquer (consolidation) ;
- aider les apprenants à apprendre comment utiliser des appareils technologiques et/ou suivre certaines procédures scientifiques (savoir-faire) ;
- développer, chez l'apprenant, la compréhension de l'approche scientifique d'enquêter ou d'investiguer (sens).

Lors d'une séance de TP, les apprenants sont en général en « mode guidé ». Ils ont à répondre à une série de questions grâce à des manipulations ou des expérimentations, et préparer un compte rendu témoignant des activités effectuées et commentant les résultats obtenus. Cette activité est sensée les aider à développer des compétences « opératoires » dont ils auront (en principe) besoin dans leur vie professionnelle.

Une compétence est « *la faculté de mobiliser un ensemble de ressources cognitives (savoirs, capacités, informations, etc.) pour faire face avec pertinence et efficacité à une famille de situations* »³⁹, et le TP représente la situation convenable pour la développer.

³⁹ Entretien de Philippe Perrenoud, Université de Genève. Paru en portugais dans Nova Escola (Brasil), Septembre 2000, pp.19-31. Voir : http://www.unige.ch/fapse/SSE/teachers/perrenoud/php_main/php_2000/2000_30.html

La construction de compétences via les TP implique de différents types de savoirs (voir Figure 2.1) :

1. le **savoir conceptuel** : est une représentation personnelle des concepts majeurs dans un système, ce qui implique de réaliser un réseau interconnecté de connaissances riches en relations et en compréhension. Le savoir conceptuel ne peut pas être appris par cœur mais à travers un apprentissage réfléchi, où l'apprenant doit reprendre l'initiative et raisonner d'une manière personnelle. Les connaissances apportées en amont par l'enseignant, la préparation de l'environnement de TP et l'organisation de la séance conduisent l'apprenant à en faire ;
2. le **savoir procédural** : correspond à la mise en pratique de connaissances dans la réalisation d'une tâche, ce qui implique de posséder un langage de représentation symbolique répondant à une situation donnée, connaître des règles, algorithmes et procédures. Le savoir conceptuel conduit à augmenter la compréhension, la génération et le transfert d'une procédure correcte, tandis que le savoir procédural conduit à augmenter l'adoption (et également la compréhension) et le transfert d'une procédure instructive ;
3. le **savoir-faire opératoire** : correspond à une façon de faire basée sur les savoirs conceptuel et procédural précédents et préalables. Ce genre de savoir est souvent tacite et ne peut pas être transféré d'une personne à l'autre verbalement ou par écrit. Il faut le vivre pour l'acquérir personnellement.

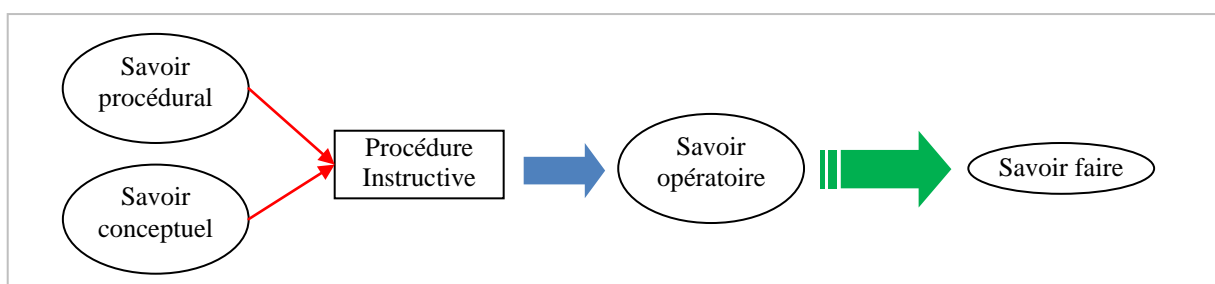


Figure 2.1 : Construction du Savoir-faire.

Les TP doivent en principe conduire à la construction de compétences chez l'apprenant, afin qu'il puisse décider en autonomie (le projet y contribuera aussi largement) et maîtriser des situations ou dispositifs avec pertinence et **performance**. Par exemple : dans un TP d'électronique où les apprenants ont à concevoir puis réaliser un circuit logique selon un cahier des charges proposé par l'enseignant, la pratique traditionnelle consiste à écrire l'expression booléenne décrivant les fonctionnalités requises, choisir les éléments logiques

correspondants (AND, OR, NOT, etc.), construire le circuit correspondant (avec Electronics Workbench⁴⁰ par exemple⁴¹), puis l'expérimenter. La bonne pratique conduit l'apprenant à minimiser l'expression booléenne (en utilisant l'algorithme de Quine–McCluskey, ou le tableau de Karnaugh) afin de réduire le nombre d'éléments logiques à utiliser (notion de performance ou de qualité). Autre exemple : face à une anomalie de fonctionnement d'une machine dans un système automatisé de production (bras robot d'assemblage de colis sur une palette par exemple), quelles sont les étapes d'arrêt d'urgence à enchaîner pour éviter l'accumulation des palettes sur le convoyeur et donc le blocage du système ? Il s'agit là d'entraîner les apprenants à réagir avec pertinence face à une situation de panne ou d'urgence, dans les environnements industriels et de fabrication. Seul le TP fournit un environnement adéquat.

Ces exemples simples montrent des cas d'apprentissage où la connaissance théorique doit être complétée par une pratique (cognitive ou manuelle) pour construire des bonnes compétences chez l'apprenant lui permettant d'affronter une situation professionnelle avec efficacité.

2.2 Les TP dans l'enseignement supérieur

Dans cette section nous présentons les rôles des TP dans l'université, l'environnement pour les accueillir (les plates-formes) et l'émergence des STIC dans ces plates-formes. Cette section introduit les TP et leurs environnements pour motiver, dans la section qui suit, la discussion sur l'**intégrabilité** de TP dans la démarche du e-learning.

Dans l'enseignement supérieur, la formation pratique est un facteur traditionnel et incontournable pour préparer les apprenants à leur vie professionnelle [Blanch-2010]. L'intégrabilité des apprenants dans le marché du travail, après l'obtention de leurs diplômes, dépend de leur niveau scientifique, des connaissances pratiques et savoir-faire qu'ils ont acquis et de l'adéquation de ces derniers aux besoins du marché les plus fréquents.

Réciproquement des professionnels, faiblement diplômés, peuvent accumuler un savoir-faire opérationnel performant grâce à de nombreuses années de pratique de terrain. En venant chercher une « **unification** » de ces savoirs expérimentaux auprès des instituts de formation

⁴⁰ Voir le site officiel du logiciel Eelectronics Workbench <http://www.electronicworkbench.com/>

⁴¹ Exemple de TP sur les circuits logiques : <http://electrostists.free.fr/DIVERS/fonctions%20logiques.pdf>, et un autre exemple de TP plus complexe : <http://www.isi.rnu.tn/isi/images/tp/tpcircuitlogique.pdf>

(VAE, VAP, Badge, MS...) ils rejoignent par cette démarche « Bottom-Up » ceux qui ont suivi un cursus traditionnel relevant plutôt d'une démarche « Top-down ».

Le savoir-faire est la connaissance des moyens qui permettent l'accomplissement d'une tâche, ou simplement c'est la connaissance de pratiques requises pour transformer les entrées dans des sorties d'une manière la plus efficace [Majchrzak et al.-2005].

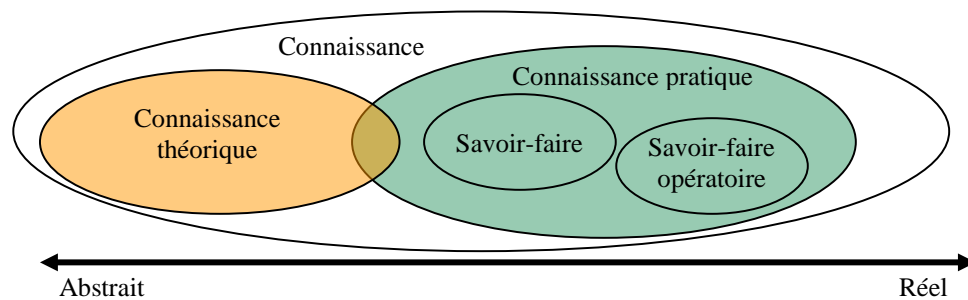


Figure 2.2 : Catégorisation des connaissances dans l'éducation.

Tous les établissements éducatifs mettent à disposition des **plates-formes** expérimentales (TP) complémentaires des autres modes pédagogiques. Ce qui les différencie est leur **complexité**, leur **volume** et la **fréquence** de leur emploi tout au long des cursus. Ces plates-formes mettent en scène des appareillages technologiques, des simulateurs via des environnements informatisés, des maquettes et modèles réduits, des environnements reconstruisant les contextes d'usage, des appareils d'analyse et de traitement, des outillages pédagogiques (tableaux de bord, outils statistiques, cartes de contrôle, ...). Ces plates-formes n'ont pas cessé d'évoluer en intégrant de plus en plus de technologies avancées et en proposant des pédagogies mieux adaptées. [Feisel-2005].

Ainsi l'instrumentation « standard » des équipements (oscilloscopes, spectromètres, enregistreurs de trajectoire, ...), l'interfaçage électronique de ces équipements et le contrôle fourni en utilisant les ordinateurs a fait apparaître des **laboratoires assistés par ordinateurs**. L'archivage et l'historisation des expérimentations sont devenus possibles, et les équipements sont devenus accessibles à distance. A cela s'ajoutent des ressources virtuelles sous formes de simulateurs qui fournissent la capacité de dérouler des expérimentations sur des systèmes virtuels similaires aux réels quand ces derniers ne sont pas disponibles, ou pour des raisons de sécurité.

2.3 Les acteurs principaux dans un environnement de TP

Différents intervenants sont impliqués dans un TP, et chacun a son rôle selon le travail adressé à lui/elle (voir Figure 2.3). En général, le TP est réalisé en deux phases : la préparation et le déroulement. Selon ces phases, nous pouvons identifier les acteurs principaux comme :

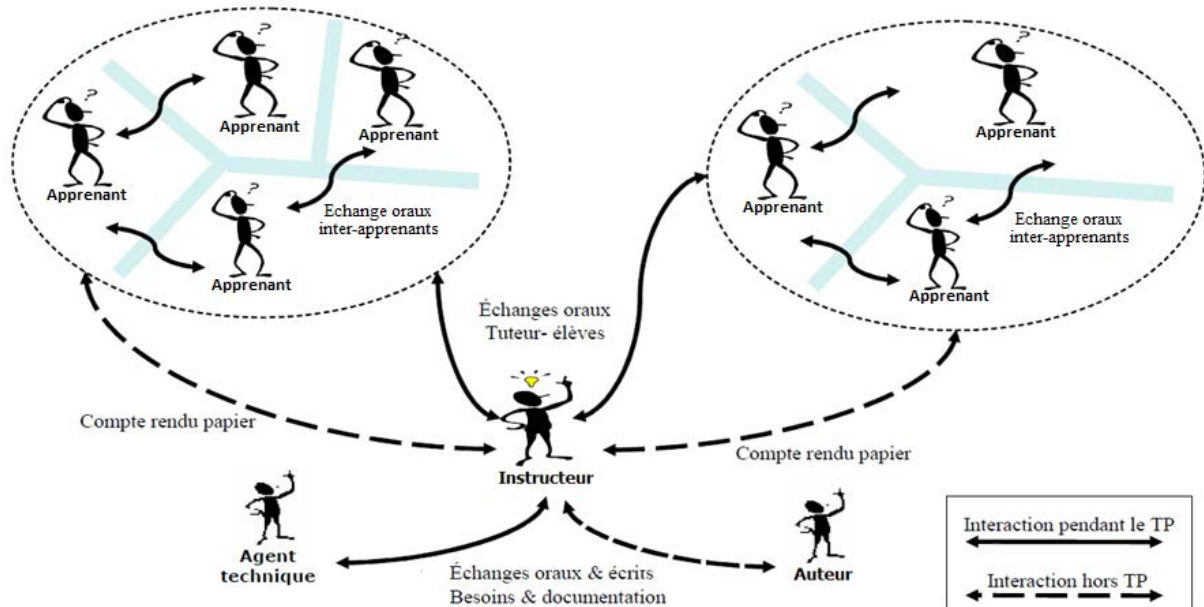


Figure 2.3 : Les différents acteurs et leurs interactions dans un TP en présentiel⁴².

- l'auteur: c'est celui qui identifie les besoins cognitifs, définit les objectifs pédagogiques, recense les ressources techniques à mobiliser lors du TP, propose des scénarios pédagogiques et prépare les documents associés. C'est l'**architecte** du TP ;
- le technicien : c'est la personne qui prépare (en amont) l'environnement (en fonction de la spécificité de la séance), le maintient (entretien et prévention des incidents), et intervient en temps réel (lors des séances) en cas de panne, d'incident ou de problème de sécurité. La bonne tenue de son rôle est une **assurance** pour le bon déroulement du TP (en général non reprogrammable en cas d'incident majeur) ;
- l'instructeur : c'est celui qui conduit la séance de TP et garantit son bon déroulement pédagogique. Il est **garant** de l'atteinte des objectifs pédagogiques fixés ;
- l'apprenant : il s'agit d'une ou de plusieurs (cas d'activités coopératives autour du même dispositif) personnes qui suivent les différentes activités proposées de façon plus ou moins **préprogrammée**.

⁴² Extrait de [Benmohamed-2007] p.104 et modifié.

Tous ces intervenants sont bien entendu en interaction. Pour assurer une efficacité globale.

2.4 Comment intégrer les TP dans une démarche e-learning globale

Les Travaux Pratiques (TP) jouent un rôle clé dans le processus éducatif et la construction cognitive des apprenants en complémentarité des autres modes (Cours, TD, Projet...).

Les propositions d'intégration des TP dans un environnement e-learning varient selon la discipline, les exigences et les objectifs pédagogiques. Une étude bibliographique sur les conditions de mutation vers un environnement e-learning, révèle que cette intégration est conditionnée par les types de ressources mobilisées autour d'un TP (en dehors des ressources humaines⁴³), comptons ici :

- ressources **logicielles** liées au domaine (simulateurs, logiciels de pilotage et de traitement, ...): certains environnements disposent de logiciels de simulation utilisables en présentiel ou à travers un réseau (par exemple [Candelas et al.-2006]) permettant aux apprenants d'expérimenter sans risques et avec une maîtrise temporelle (temps compressé quand le système réel aurait des temps de réaction incompatibles avec la durée d'une séance). D'autres utilisent des logiciels de pilotage des dispositifs technologiques, traitement et analyse de données, etc. (par exemple [Logar et al.-2011]). Les types de logiciels varient selon le domaine ;
- ressources **technologiques** (matériel, capteurs, organes de commande, ...) : c'est le cas de dispositifs technologiques représentatifs de ceux utilisés en situation professionnelle. Ils sont souvent onéreux (achat et entretien), encombrants (même s'il peut s'agir de « modèles réduits ») et peu avarés de consommables (matières premières, produits finis pas toujours recyclables...). Ils sont accessibles au sien d'un TP en présentiel (par exemple [Corter et al.-2007]) ou à distance (par exemple [Domínguez et al.-2011]) ;
- ressources **pédagogiques** : il s'agit de scénarios pédagogiques à exécuter pendant un TP, documentations liées au sujet de TP (notices, tutoriels, instructions), animation illustratives à présenter pendant le TP, exercices à faire par les apprenants, résultats à analyser, ... La disponibilité de ces ressources en format électronique simplifie leur

⁴³ Ressources humaines de TP : auteur, instructeur, technicien, apprenants.

accès par les différents acteurs de TP (auteur, instructeur, apprenant, technicien) via LMS, ainsi que leur manipulation (création/édition) via LCMS ;

- ressources **financières** (achat, renouvellement, réparation, maintenance...) : il s'agit de consacrer une partie du budget d'un établissement éducatif pour la possession des différents types de ressources à utiliser dans un TP. Les ressources financières sont liées directement aux autres types de ressources fournies par l'établissement, et varient selon leur quantité et qualité, la nécessité de leur renouvellement, la fréquence de leur réparation et maintenance. Dans un cadre e-learning, certaines dépenses **baissent** en ce qui concerne les ressources :
 - pédagogiques : l'accès/exploitation de la version électronique de ces ressources est plus facile qu'en version papier ;
 - technologiques : le **partage** d'un dispositif technologique, via un accès mutualisé à base de réseaux, par plusieurs partenaires diminue la nécessité d'en posséder une instance chacun, ce qui conduit par conséquence à réduire les dépenses de l'achat, renouvellement, réparation et maintenance des dispositifs partagés.
- ressources **consommables** (matières premières, produits fabriqués, adjuvants, ...) qui restent généralement inchangeables dans le TP assisté par les STIC. Néanmoins, la simulation peut jouer un rôle alternatif pour émuler l'utilisation de certains éléments dans une expérimentation (par exemple : simulation d'un composant manquant dans un système manipulé à distance [Emami et al.-2008], remplacer une caméra fournissant un retour visuel sur le mouvement d'un bras robot par l'utilisation d'un modèle 3D pour obtenir le retour d'une expérimentation [Candelas et al.-2006]).

Dans le cadre du e-learning, le mode d'accès aux différentes ressources de TP peut être **en présentiel** pour manipuler des ressources locales via un ordinateur (ExAO cf. §1.6.2.3). C'est le cas lorsqu'un logiciel installé sur un ordinateur local est utilisé (MATLAB ou un simulateur par exemple), ou lorsque la manipulation d'un dispositif passe nécessairement par un ordinateur. Cet accès peut aussi bien être **distant**, dans le cadre d'un téléTP pour manipuler à distance des ressources virtuelles (par exemple [Candelas et al.-2006]) ou réelles (par exemple [Domínguez et al.-2011]).

L'objectif pédagogique d'un TP est un facteur complémentaire pour **déterminer** le type de ressources et le mode d'accès les plus adaptés à son atteinte. Par exemple, un environnement virtuel de TP est proposé dans [Norton et al.-2008] pour motiver l'**auto-apprentissage** à travers un système de réalité virtuelle. D'autres proposent un environnement combinant la réalité avec la virtualité pour améliorer la **cognition** de cet environnement (réalité mixte [Knecht et al.-2012]). D'autres encore, introduisent le concept de Serious Games pour **solliciter l'apprentissage** comme dans [Hakulinen-2011].

Le Tableau 2.1 résume les principales mutations s'effectuant dans l'utilisation de chacune des ressources lors de passage d'un TP traditionnel à un TP dans un cadre e-learning.

Type de ressources	Mutation dans l'utilisation	Eléments inchangés
Logicielles	logiciels supplémentaires à utiliser	Les mêmes logiciels utilisés dans un TP traditionnel sont utilisés dans un cadre e-learning
Technologiques	- accès aux composants technologiques médiatisé par l'ordinateur - partage des ressources technologiques - de nouveaux composants s'ajoutent pour l'interfaçage entre l'ordinateur et les ressources exploitées	Les mêmes types de ressources technologiques sont exploités
Pédagogiques	- disponibilité au format électronique en remplaçant la version papier, en simplifiant leur accès n'importe quand et de puis n'importe où - possibilité de partager/réutiliser les ressources	
Financières	- coûts réduits de possession des différents types de ressources - coûts supplémentaires pour l'interfaçage entre l'ordinateur et les ressources exploitées	Coûts de maintenance, réparation et assistance technique
Consommables	peu changés comme dans le cas de simulation de certains consommables dans certains TP	Généralement inchangés

Tableau 2.1 : Principales mutations dans l'utilisation des ressources dans un cadre de TP en e-learning.

2.5 Evolution de l'environnement TP vers e-TP

Une plate-forme de TP représente l'environnement type où sont installés les équipements, les dispositifs technologiques, les maquettes pédagogiques et se déroulent les différentes activités des TP.

Aujourd'hui, les établissements d'éducation supérieure tentent de tirer profit des STIC (surtout dans les disciplines techniques et scientifiques) pour fournir aux apprenants une formation de **meilleure qualité** à travers des environnements e-learning (e-TP). L'e-TP prend plusieurs formes selon :

- le type des ressources dont il dispose ;

- le mode d'accès à ces ressources ;
- l'objectif pédagogique à atteindre (cf. §2.4).

Nous détaillons ces formes dans les sections suivantes en profitant de l'analyse présentée dans [Gravier-2007]. Nous détaillons également les différentes architectures qui leur sont associées.

2.5.1 Plate-forme de TP virtuels (VLab)

Le TP virtuel (*VLab* étant l'abréviation de terme anglais « *Virtual laboratory* ») s'appuie essentiellement sur la **simulation** (voir Figure 2.4). Il représente une solution largement adoptée [Schaf-2009], lorsque le système réel :

- n'est pas disponible sur site ou à distance (e.g. pour raisons financières) ;
- n'est pas pilotable sans dommages (e.g. écosystème) ;
- n'est pas homothétiquement réductible (modèle réduit) sans modifier de façon majeure les règles de fonctionnement qui le gouvernent (e.g. haut fourneau).

Cette solution réduit l'écart entre **concepts et mises en application**, mais, en même temps, dénuée de **risques et décontextualisée**, elle ne contribue pas à développer un réel savoir-faire opérationnel.

La simulation est parfois utilisée par exemple en robotique. [Candelas et al.-2006] présente un TP virtuel (Robolab2) permettant aux apprenants de travailler en simulation avec un robot industriel. L'environnement logiciel proposé autorise l'intégration de nouveaux modèles de robot à travers une interface flexible. En informatique, nous retrouvons dans [Christou et al.-2007] un TP virtuel permettant aux apprenants d'élaborer et tester des réseaux de capteurs, dans le cadre d'un cours de master informatique et réseaux. Nous pouvons trouver d'autres exemples de TPs virtuels en électronique [LI-2005], en génie civil [Kuester-2007], en automation, en informatique [Buhler et al.-2000], ...

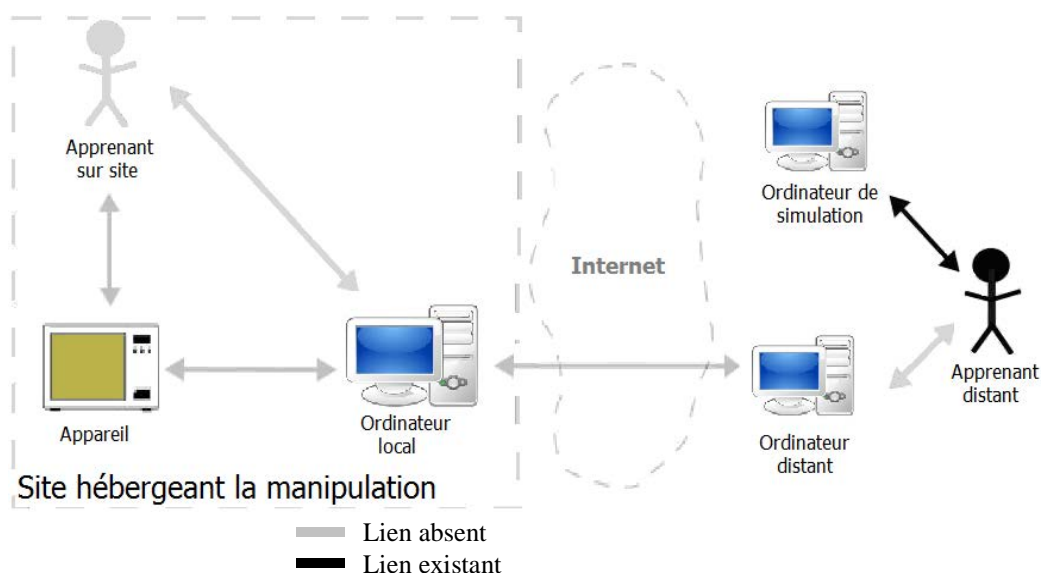


Figure 2.4 : Les plate-forme de TP virtuels.

Nous avons élaboré dans le Tableau 2.2 une analyse SWOT⁴⁴ concernant les TP virtuels.

Forces	Faiblesses	Opportunités	Menaces
<p>Technique : nous pouvons facilement adapter un simulateur à une nouvelle version du système réel qu'il émule, et la mise à jour du simulateur devient beaucoup moins chère que l'achat de nouveaux équipements.</p>	<p>Réalité d'expérimentation : En général, les résultats de la simulation sont basés sur des règles et des lois calculatoires qui ne permettent pas de montrer un comportement naturel, mais plutôt idéal (non prise en compte des frottements secs, par exemple, en mécanique).</p>	<p>La prise en compte de l'écart entre la théorie et la réalité, pour améliorer les modèles de simulation, permettra d'élargir le domaine d'utilisation du simulateur vers d'autres disciplines que la formation (recherche scientifique par exemple).</p>	<p>Une perte du sens de la réalité de la part des apprenants qui risquent de ne plus savoir manipuler le vrai système</p>
<p>Sécurité : l'utilisation erronée de dispositif réel par les non spécialistes peut dégrader le dispositif et être potentiellement dangereux pour l'utilisateur, selon la nature du dispositif et du TP (par exemple : TP en chimie ou électricité). En simulation, le système ne tombe pas en panne (sauf programmée). De plus, un simulateur peut fournir une traçabilité des erreurs et même proposer un voie à suivre afin de résoudre un problème.</p>	<p>L'adoption de simulateur de système pour un TP ne permet parfois pas de familiariser les apprenants avec l'aspect de sécurité lié.</p>	<p>Développer des cas simulant les aspects de sécurité liés à un système.</p>	<p>Il est difficile de prédire tous les aléas qui peuvent être reproduits liés à la sécurité, surtout s'il s'agit de l'utilisation erronée du système simulé.</p>
<p>Economique : les simulateurs sont nettement moins chers comparés aux équipements réels qu'ils émulent (un seul développement logiciel pour émuler un</p>	<p>Faisabilité : Développer et faire évoluer un simulateur nécessite une</p>	<p>Les méthodes, modèles et algorithmes utilisés pour développer un</p>	

⁴⁴ SWOT est une abréviation de l'anglais « Strengths, Weaknesses/Limitations, Opportunities, and Threats » signifiant respectivement : Force, Faiblesses / Limitations, Opportunités et Menaces.

nombre illimité d'équipements).	compréhension théorique du système simulé qui peut être difficile à transférer aux développeurs non spécialistes dans le domaine.	simulateur dans un domaine donnés peuvent être exploités dans d'autres domaines de simulation, ce qui peut réduire le coût / temps de développement / amélioration de simulateurs	
Disponibilité : l'utilisation des simulateurs n'est pas limitée aux campus, les apprenants n'ont qu'à se munir d'un ordinateur et du logiciel du simulateur ou bien, une connectivité vers le serveur fournissant le logiciel consacré à la simulation (par exemple : Matlab-Simulink model [Logar et al.-2011]). Cet aspect satisfait bien le besoin personnel et la volonté de s'instruire n'importe où et n'importe quand.	L'aspect collectif et l'échange entre les apprenants sont absents dans un VLab	Développer des moyens permettant la communication entre les apprenants pour réduire l'impact de l'absence physique	
Compréhension : pour un nombre élevé d'apprenants dans un TP sur un dispositif technologique, la compréhension de ses fonctionnalités et la disponibilité du temps pour appliquer des manipulations deviennent difficiles à fournir, alors qu'un simulateur de ce dispositif peut fournir aux apprenants la base de connaissance nécessaire et assurer l'élaboration des activités pratiques demandées.	Cognition : la simulation ne remplace en aucun cas la présence physique de l'intervenant dans une expérimentation présente et ce que cela peut entraîner pour la cognition visuelle ou manuelle.		

Tableau 2.2 : Analyse SWOT pour la plate-forme de TP Virtuel (VLab).

2.5.2 Plate-forme de téléTP (RLab)

Le téléTP, ou le TP distant, (voir Figure 2.5) est introduit dans la littérature sous plusieurs titres, comme “*Web-Based Experimentation*” [Gillet et al.-2005] ou “Laboratoire collaboratif distant [Fayolle et al.-2010]”, et même parfois sous la bannière des TP virtuels (comme [Kikuchi et al.-2004] ou [Said et al.-2012]) mais cela ne correspond pas au terme VLab présenté dans la section précédente. Ce type de TP, comme son nom l'indique, est réalisé à distance [Coquard et al.-2008]. Une plate-forme de téléTP « RLab » (abréviation de terme anglais « Remote laboratory ») contient a priori les mêmes équipements qu'une plate-forme de TP traditionnelle mais avec une conception différente et des outils additionnels pour répondre aux besoins liés à l'éloignement des intervenants (**vison et intervention** à distance). Le téléTP fournit une communication entre les composants de TP et les apprenants via un réseau local ou étendu [Kikuchi et al.-2004], ou l'internet [Rohrig-2003]. Des capteurs spécifiques (camera web, microphones, capteurs de force, capteurs de mouvement, ...) assurent la capture et la numérisation des informations pour leur transmission à distance [Candelas et al.-2006], [Lallican et al.-2007]. Réciproquement, l'apprenant intervient grâce à

un panneau de commande et de contrôle déporté. En général, le RLab est donc équipé d'un serveur de contrôle régissant la commande et l'observation à distance du système étudié.

De nombreuses références (plus de 80 auteurs différents) font état de recherches menées sur l'utilisation d'Internet pour conduire des expérimentations sur des dispositifs technologiques, comme dans [Ashish-2006]. Ce type de plate-forme est largement abordé dans des domaines scientifiques comme le Génie Civil [Kuester-2007], l'Électronique [Davoli et al.-2006], la Robotique [Ramaswamy et al.-2006], l'Automation [Bellmunt et al.-2006], etc.

Le RLab représente une opportunité de réaliser des TP à distance tout en étant confronté à la réalité. Mais, comme dans le cas de VLab, le RLab ne fournit pas l'interaction directe avec les composants du système.

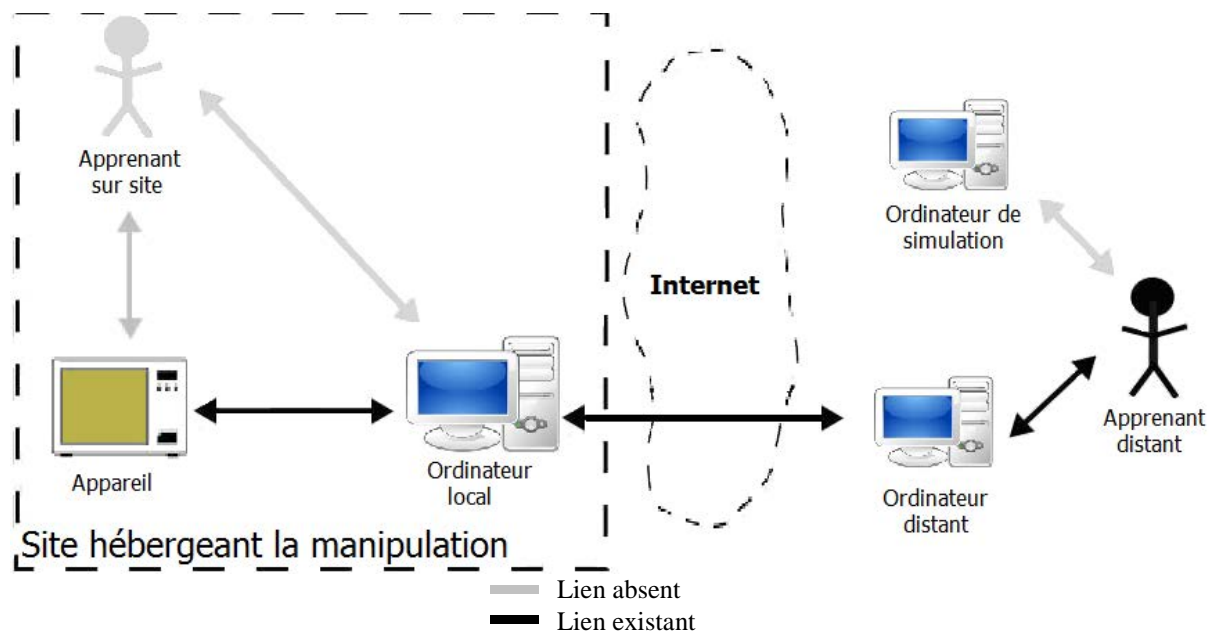


Figure 2.5 : La plate-forme de téléTP ([Gravier-2007] p.41).

Le livre « *Advances on remote laboratories and e-learning experiences* » [Jesus-2007], donne un aperçu des différents contextes de développement et d'utilisation des R Labs, et leur impact sur les pratiques pédagogiques. Nous avons élaboré dans le Tableau 2.3 une analyse SWOT sur les R Labs.

Forces	Faiblesses	Opportunités	Menaces
Réalité de l'expérimentation : les apprenants peuvent suivre à distance des TP sur des matériels réels, ce qui leur permet de se construire des	- Dépendance à une connexion réseau ; - dépendance à un système audiovisuel temps réel pour	Choisir son TP à la carte depuis des plateformes partout dans le monde	Le TP distant omet l'aspect social entre les apprenants et ne permet pas la communication directe avec l'instructeur

compétences pratiques	observer une expérimentation et ne pas rater des événements importants		
Disponibilité : les utilisateurs de RLabs sont libérés des contraintes temporelles et présentes, et peuvent dérouler des expérimentations depuis n'importe où et n'importe quand	- Une tentative multiple d'accéder à une même ressource peut la rendre indisponible pour certains utilisateurs ; - la gestion d'accès et du temps d'utilisation des ressources peut être compliquée	- La disponibilité des ressources alternatives avec une gestion performante de leur temps d'utilisation peut assurer la disponibilité de ces ressources pour les apprenants intéressés ; - le partage entre plusieurs RLabs permettra d'élargir l'échelle d'échange des expériences et des technologies exploitées	Pour les établissements qui dépendent entièrement sur les RLabs (université virtuelle) pour aboutir des TP, l'échec du système de communication, d'accès ou de partage des ressources peut empêcher ces TP
Mutualisation et partage des ressources : RLabs permettent de mutualiser des ressources matérielles et les partager entre plusieurs partenaires, tout en profitant de l'accès distant à ces ressources			
Economie : le partage des ressources au sein du RLab permet aux établissements partenaires de réduire le budget dédié à l'acquisition et à la maintenance de ces ressources	Coût additionnel des équipements à installer pour permettre l'interaction distante	Le coût réduit par le partage permet d'enrichir le RLab par des nouvelles ressources qui n'étaient pas envisageables	
Sécurité : la manipulation distante des matériels dangereux (en chimie par exemple) assure la sécurité de l'apprenant ou de l'intervenant non professionnel	Un système distant non protégé contre les actes erronés peut être endommagé lors de son utilisation par les utilisateurs non expérimentés	Le développement du système de prévention des actes erronés permettra de rendre le RLab plus fiable	La sécurité liée à l'utilisation d'un dispositif est un aspect essentiel à enseigner aux apprenants, l'existence d'un système interdisant l'utilisation erronée peut empêcher l'acquisition de compétences liées à cet aspect

Tableau 2.3 : L'analyse SWOT pour la plate-forme de téléTP (RLab).

2.5.3 Plate-forme de TP hybrides (HLab)

Ce type de TP est un **mixte** des deux types précédents (VLab et RLab) (cf. Figure 2.6). Il fusionne les caractéristiques virtuelles et distantes afin de présenter un environnement d'expérimentation plus flexible et plus puissant. Une plate-forme de TP hybride (*HLab* étant l'abréviation du terme anglais « *Hybrid Laboratory* ») est une plate-forme de téléTP muni d'un système de simulation associé, où la **simulation** peut jouer plusieurs **rôles**. Comme en VLab, la simulation peut être appliquée pour préparer l'apprenant et le familiariser avec le système ou le sujet de TP. Mais, au contraire du VLab, pendant une session de téléTP sur un système réel la simulation émule le comportement d'une partie de ce système [Guimaraes et al.-2000, Ashish-2006, Emami et al.-2008]. Ce cas se produit quand la partie simulée n'existe pas (par exemple pour des raisons de coût), ou bien quand il n'est pas possible de connecter cette partie au système du HLab pour récupérer des informations sur son évolution (par exemple : mesurer la température du noyau magnétique d'un moteur électrique). De même, la simulation

est **efficacement** exploitée pour clarifier le comportement du système, comme dans [Candelas et al.-2006] où la supervision via une caméra n'est pas suffisante pour observer le mouvement d'un bras de robot : une simulation fournit une vision 3D de ce mouvement.

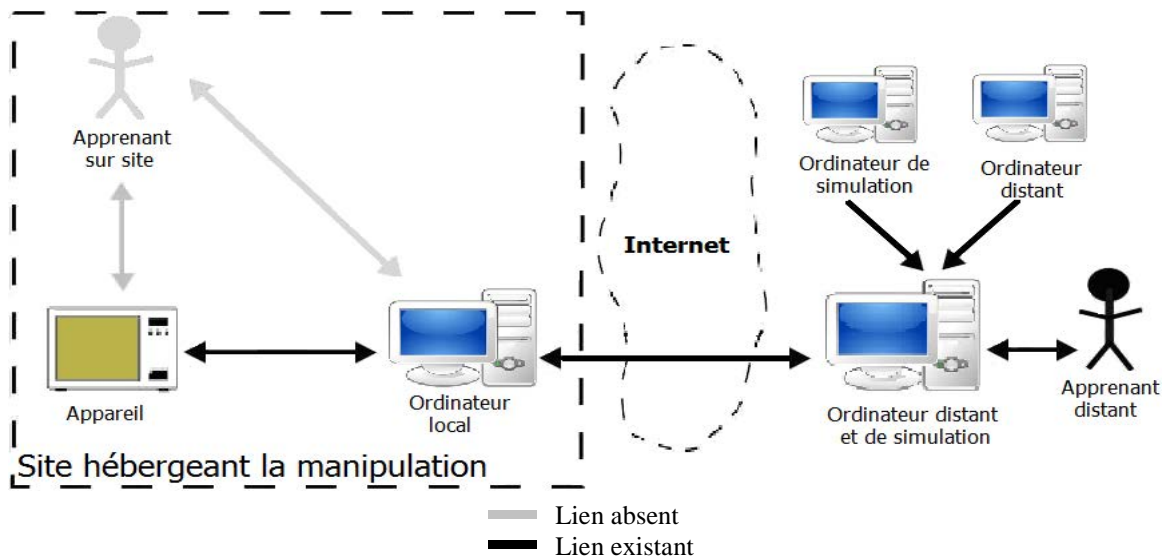


Figure 2.6 : La plate-forme de TP hybride.

Nous avons élaboré dans le Tableau 2.4 une analyse SWOT de plate-forme de TP hybride (HLabs), en n'y faisant figurer que les aspects que nous voyons spécifiques aux HLabs, comme ils partagent des caractéristiques de VLabs et RLabs présentés précédemment.

Forces	Faiblesses	Opportunités	Menaces
Economie : l'impossibilité de fournir des équipements lourds est compensée par leur simulation.	L'adoption de simulations pose parfois la question sur la qualité des compétences pratiques acquises	Simuler des systèmes et les tester afin de décider s'ils méritent d'être achetés permet d'élargir l'efficacité du HLab	
Sécurité : la simulation d'une opération dont les résultats sont imprédictibles permet d'éviter un risque potentiel lors de l'élaboration de cette opération sur un système réel	La simulation d'une opération liée à un dispositif réel peut être facultative, et la sûreté de l'utilisation non erronée peut se limiter à l'initiative de l'apprenant	Elargir le niveau de sécurité en simulant tous les systèmes réels disponibles dans un HLab, et obliger l'utilisateur à simuler son expérimentation avant de l'élaborer sur un dispositif réel afin de réduire les risques potentiels de son utilisation erronée	<ul style="list-style-type: none"> • L'obligation à simuler toute expérimentation peut surcharger le TP. • L'indisponibilité de simulateur d'un système réel dans un HLab expose ce système à une utilisation erronée par les non professionnels
Disponibilité et partage : la simulation permet de compenser l'indisponibilité d'une ressource matérielle en cours d'utilisation et permet donc d'améliorer le partage de ses ressources à une échelle	L'accès à un simulateur au lieu d'un dispositif réel, dans le cas d'accès multiples au même système, ne permet pas à certains apprenants d'élaborer une expérimentation réelle, ce qui peut	Le partage et la mutualisation des ressources entre différents HLabs permettront de réduire l'indisponibilité de certaines ressources matérielles, et d'élargir l'échelle d'échange des expériences et des technologies exploitées	

plus grande	limiter l'acquisition de compétences attendues		
-------------	--	--	--

Tableau 2.4 : L'analyse SWOT pour la plate-forme de TP Hybride (HLab).

2.5.4 Synthèse sur l'e-TP

La littérature fournit des comparaisons, entre les TP traditionnels et les e-TP [Corter et al.-2004] et entre les différents types de ELabs [Lindsay-2005, Baccigalupi et al.-2006, Ashish-2006].

Dans la majorité des cas, les modèles d'e-TP proposés sont propriétaires (scénario pédagogique, documentations et autres ressources), ce qui les rend spécifiques et inapplicables sur un système similaire. Peu de recherches s'attaquent aux problématiques de **généricité** des modèles d'e-TP et de la réutilisabilité des ressources [Benmohamed-2007].

Le Tableau 2.5 reprend les aspects manquants.

Aspect manquant	Description
Evolutivité des composants de ELab	Rares sont les recherches qui se penchent sur l'évolutivité des contenus d'un ELab (comme le confirme [Sivakumar et al.-2005]), pour faire d'évoluer le contenu matériel et logiciel d'un ELab (comme dans [Fayolle et al.-2010]).
Echange de contenu pédagogique entre différentes plates-formes	Rares sont les recherches qui s'intéressent à proposer des contenus pédagogiques échangeables entre e-TP de différentes plates-formes (telles que [Bagnasco et al.-2008, Benmohamed et al.-2009]).
Réutilisabilité des activités et ressources pédagogiques	Les activités de TP au sein d'un ELab (présentiel ou distant) sont, dans la plupart des cas étudiés lors de notre étude bibliographique, spécifiques aux systèmes sujets de formation, y compris les ressources de ces activités. En général, les ressources ne sont pas conçues pour être utilisées ailleurs (avec des systèmes similaires par exemple), et peu de recherche s'intéressent à leur généricité (comme dans [Benmohamed et al.-2009]).

Tableau 2.5 : Aspects manquants dans les ELabs existants.

2.5.5 Les TP dans la plate-forme de TP Electronique (ELab)

Le TP électronique (e-TP) est une catégorie de TP regroupant tous les types de TP nécessitant l'utilisation de l'ordinateur pour contrôler le déroulement d'une expérimentation, soit en présentiel soit à distance. Autrement dit, une plate-forme d'e-TP est un environnement assisté par ordinateur permettant de dérouler des manipulations sur des dispositifs réels ou des simulateurs aussi bien en **présentiel** qu'à **distance** (voir Figure 2.7).

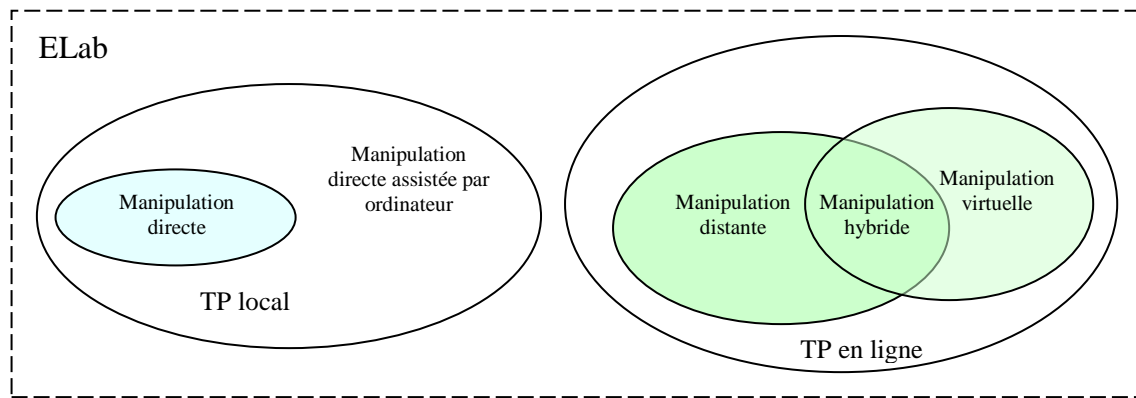


Figure 2.7 : ELabs : TP assistés par des TIC.

La **structure** et les services fournis par un ELab diffèrent selon le domaine scientifique et les objectifs des expérimentations. Dans certains domaines comme l'instrumentation et mesures, la simulation peut être suffisante et efficace [Grimaldi-2009]. Les auteurs présentent un TP virtuel complètement basé sur la simulation : à travers une architecture répartie, des outils de mesure sont simulés afin de réaliser des activités pédagogiques autour de leur réglage et leur utilisation. Les calculs sont effectués entre plusieurs PC d'un réseau simulé. Dans [Norton et al.-2008], en génie chimique, l'objectif des concepteurs est de permettre l'auto-apprentissage (traduction du terme anglais *Self-Learning*) au sein d'un environnement de réalité virtuelle. À travers des jeux d'apprentissage et d'un environnement favorisant la résolution de problèmes, une banque de connaissances est construite à partir des expériences acquises par les utilisateurs lors de tâches d'apprentissage. Dans [Buchanan-2011], l'objectif est d'entraîner les apprenants sur la cyber-sécurité à l'aide d'un jeu d'apprentissage.

Dans d'autres situations, les RLabs sont **enrichis** par un environnement virtuel, combinant les aspects réel et distant. Par exemple, en électronique, [Bagnasco et al.-2008] considère chaque expérimentation (virtuelle ou distante) en tant qu'objet d'apprentissage enregistré dans des répertoires dédiés, ce qui permet d'exploiter les données de ces objets à travers un paquetage pédagogique (*Learning Package*) exécutable plus tard via un LMS.

Le Tableau 2.6 compare les différents types de plate-forme de TP [Gravier-2007].

	Avantages	Inconvénients
Plate-forme de TP Virtuel (VLab)	<ul style="list-style-type: none"> - Changer d'échelle (depuis l'infiniment petit à l'infiniment grand) - Palier la dangerosité - Maîtriser le temps d'expérimentation (en accélérant l'expérimentation en cas de forte inertie du dispositif par exemple) - Minimiser le risque technologique 	<ul style="list-style-type: none"> - Manipulation aseptisée où tous les aléas ne peuvent être reproduits - Complexité des modèles croissante avec le réalisme désiré - Se substitue parfois à l'apprenant quand le simulateur réalise une opération (d'une manière automatique)

	(exposition au danger, utilisation erronée de dispositif, etc.) - Limiter les coûts d'achat et d'entretien d'une manipulation trop onéreuse	que l'apprenant doit lui-même réaliser - Pose le problème de la validation des acquis et de la qualité des modèles - Isolement des apprenants
Plate-forme de TP Distant (RLab)	- Garder la conscience de la réalité de la manipulation - Acquérir des compétences pratiques plus qualifiées que celles dans le cas de VLab - Plus facile à mémoriser la manipulation élaborée avec des systèmes réels que celle avec des systèmes simulés - Favorise les situations de communication - Favoriser la mutualisation des équipements et leur usage intensif, parfois même en dehors des heures d'ouverture	- Coût additionnel des équipements à installer pour permettre l'interaction distante, sans oublier le coût des systèmes réels à manipuler - Gestion du temps d'utilisation peut être plus compliquée que dans le cas d'une plate-forme de TP traditionnel disposant les mêmes systèmes à manipuler - Utilisation rigoureuse en fonction de l'absence physique et les contraintes de connexion distante
Plate-forme de TP Hybride (HLab)	- Illustrer les limites de la théorie - Fournir parfois plus de sûreté aux manipulations pouvant porter un risque en permettant de les simuler avant de les valider sur des systèmes réels - Réduire le coût d'une manipulation en permettant de simuler des parties onéreuses d'un système manipulé - Favoriser la mutualisation des équipements et leur usage intensif	- Gestion du temps d'utilisation peut être aussi compliquée que dans le cas du RLab - Utilisation peut être aussi rigoureuse que dans le cas du RLab - Complexité des modèles croissante avec la diversité des systèmes simulés - Pose parfois le même problème de la validation des acquis que dans le cas du VLab
Plate-forme de TP Electronique présentiel (ELab)	- Assurer la conscience de la réalité de la manipulation - Plus de maîtrise de système manipulé que dans les cas précédents - Favoriser la mutualisation des équipements et leur usage intensif - Favoriser l'interaction entre les différents intervenants	- Gestion du temps d'utilisation peut être compliquée en fonction de l'usage multiple et intensif - Coût élevé des matériels à manipuler, sans oublier le coût additionnel des équipements permettant l'interaction via l'ordinateur

Tableau 2.6 : Comparaison entre les différents types de plate-formes de TP.

2.6 Notre terrain d'expérimentation

Dans cette section nous présentons notre terrain d'expérimentation, sa structure, ses différents acteurs, ses rôles et les problématiques découvertes lors de notre observation.

2.6.1 L'AIP-Primeca-RAO

L'AIP-Primeca-RAO⁴⁵ est un centre de ressources et de compétences doté de moyens industriels mis à la disposition commune de plusieurs établissements d'enseignement supérieur de la région Rhône-Alpes⁴⁶, comme support de formations approfondies dans le domaine de la productique. Les études et travaux qui y sont effectués portent sur la conception, aussi bien que sur les produits (PLM) et les moyens de production, leur mise en

⁴⁵ Voir : http://aiprao.insa-lyon.fr/fr/presentation_aip/index.htm

⁴⁶ Etablissements partenaires : ECAM, Ecole Catholique des Arts et Métiers-ECL, Ecole Centrale de Lyon-ENISE, Ecole Nationale des Ingénieurs de Saint Etienne-ENSMSE, Ecole Nationale Supérieure des Mines de Saint Etienne-INSA, Institut National des Sciences Appliquées de Lyon,-IUT Lumière de l'Université Lyon II-IUT de Roanne de l'Université de Saint Etienne-UCB, Université Claude Bernard Lyon I.

œuvre, leur gestion et leur exploitation. Dans le domaine des automatismes industriels, l'AIP met à disposition des enseignants des **équipements** (automates interfaces homme-machine, réseaux locaux industriels, ...) **technologiquement à jour**. Il est également à noter que l'AIP propose d'utiliser ce matériel à travers une plate-forme de réservation et de mis à distance, mais sans intégrer des outils pédagogiques.

Nous avons observé cet environnement, sous différents points de vue : acteur (apprenants, instructeurs, techniciens, etc.), technique (installation de ressources, mises à jour, développement, etc.), pédagogique (exploitation de ressources, déroulement de TP, etc.).

2.6.2 Le transfert libre

Le système que nous avons choisi pour dérouler nos expérimentations est le transfert libre situé dans l'atelier de l'AIP Primeca RAO. Il s'agit d'un système automatisé à caractère pédagogique, représentant les **technologies et méthodes** liées à l'utilisation de **robots d'assemblage**, d'un système de **contrôle visuel** et d'un système de **transfert**. La production utilisée comme support est le positionnement de pièces de différentes couleurs sur des porte-pièces véhiculés par des palettes (Figure 2.9). Ce système est composé de six postes semblables dont deux équipés d'un robot (la structure de chaque poste est montrée dans (Figure 2.10)), et un poste de supervision (Figure 2.8):

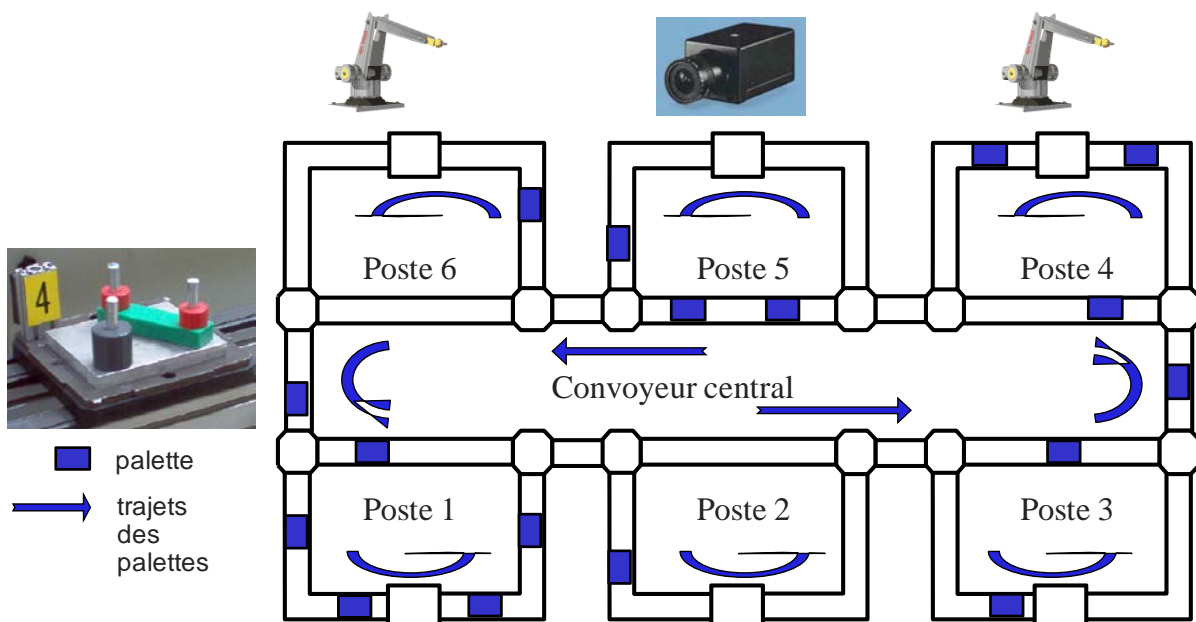


Figure 2.8 : Structure du système de transfert libre.

- Poste 1 et Poste 2 : postes d'écriture des gammes d'assemblage et de désassemblage sur les étiquettes RFID de chaque palette ;

- Poste 3 : poste de secours en cas d'indisponibilité d'un autre poste ;
- Poste 4 : robotisé (SCARA ADEPT ONE) pour poser ou retirer les pièces noires ;
- Poste 5 : poste de validation Qualité, équipé d'un système de vision IN-SIGHT 1000 pour détecter les pièces mal positionnées sur les porte-pièces ;
- Poste 6 : robotisé (Staubli RX90) pour poser ou retirer des pièces rouges et vertes ;
- Poste 7 : poste de supervision gérant :
 - la gestion des énergies de puissance (moteurs des convoyeurs, alimentation pneumatique de la ligne) ;
 - la gestion de l'arrêt d'urgence de la ligne ;
 - la gestion des modes de fonctionnement de l'ensemble du SAP.

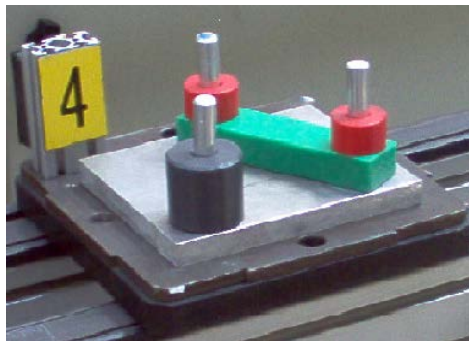


Figure 2.9 : Pièces de différentes couleurs à poser sur une palette.

Ce SAP est **partagé** entre plusieurs instituts pour des niveaux d'étude différents et des disciplines différentes, ce qui implique impérativement des **reconfigurations successives** à appliquer sur le SAP afin de remplir les différents objectifs d'utilisation dans les différents TP. En effet, la reconfiguration pour passer d'un TP à un autre est une **opération longue et fastidieuse** qui implique une reconfiguration matérielle et logicielle de plusieurs parties :

- reprogrammation des API (API n°1 à 7) en fonction du contexte de fonctionnement (ordonnancement, MES, etc.), et le paramétrage de chaque API ;
- remplissage des stocks de pièces ;
- reprogrammation des étiquettes RFID des palettes ;
- reprogrammation des robots.

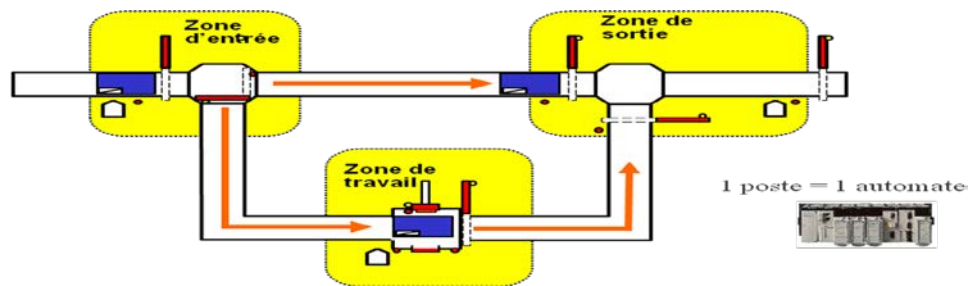


Figure 2.10 : Structure de poste.

2.6.2.1 Exemple d'utilisation : le projet MES

Le projet MES (Manufacturing Execution System), destiné aux étudiants des départements Génie Industriel et Génie Electrique, utilise abondamment le transfert libre.

Un MES est un logiciel intégrant toutes les fonctions de pilotage d'atelier telles que l'ordonnancement, le suivi de production, la gestion de la documentation et des données techniques. Il se situe entre **le niveau atelier et le niveau gestion d'entreprise** et permet d'optimiser la productivité des hommes et des machines au niveau de l'outil de production, de donner des informations en temps réel sur les ordres de fabrication et de contrôler les moyens de fabrication.

Dans le cadre de ce projet le transfert libre est utilisé comme **support de conception et d'intégration** d'un MES dans un atelier. Les apprenants ont notamment pour objectif de l'adapter pour être supervisé par un MES, ce qui les occupe (au minimum) 32 heures. À la première séance, une démonstration du fonctionnement initial du transfert libre est présentée aux étudiants. Pour préparer cette démonstration, l'instructeur doit d'abord préparer les programmes à installer sur chaque poste. Les programmes transmis aux apprenants sont les mêmes. Une documentation leur est associée et est distribuée à la première séance. Pour illustrer la complexité de ce travail, nous présentons la liste de documentations à fournir lors du projet MES :

- structure générale de l'environnement d'un MES ;
- structure de l'application MES ;
- architecture globale du transfert libre (6 postes + 1) ;
- architecture générale de chaque poste (boucle principale et dérivation) ;
- architecture détaillée de chaque poste (capteurs et actionneurs) ;

- liste et emplacement des cartes E/S spécifique à chaque API ;
- données d'entrées/sorties spécifiques à chaque poste ;
- paramètres dialogue avec boîtiers lecture/écriture pour chaque poste entrée/poste/sortie ;
- complément pour la fonction *Contrôle de la production* ;
- guide d'étude des modes de marches et d'arrêts ;
- temps d'états selon la norme NFE 60-182 ;
- vue SQL à copier sur le serveur mySQL;
- code d'accès au logiciel ;
- fichier de projet de démarrage ;
- programmes Unity (un programme par poste).

La reconfiguration du système s'effectue en fonction de l'activité pédagogique à réaliser (configuration d'automates, programmation basique API, paramétrage réseau atelier, ...).

L'exemple du MES illustre la **taille** de travail à effectuer pour reconfigurer le système de transfert libre afin d'y dérouler différentes activités pédagogiques. La **complexité** ne vient pas seulement du grand nombre de ressources à préparer, mais aussi de la sélection entre ces ressources pour atteindre les objectifs pédagogiques désignés pour chaque session de TP.

2.7 Problématiques liées à la plate-forme « transfert libre »

Nous recensons ici les problématiques découvertes liées au manque d'efficacité dans l'exploitation de la plate-forme « transfert libre » par les différents intervenants. Pour les **catégoriser**, nous avons réalisé une **observation** locale de la plate-forme, ainsi qu'une **interrogation** des intervenants et également une **enquête** générale distribuée aux instructeurs de TP dans différentes disciplines scientifiques et technologiques.

2.7.1 Problématique 1 : la variété des modes de fonctionnement

Quel que soit le système automatisé, il doit pouvoir changer de mode de fonctionnement : s'initialiser, rester en « arrêt dans l'état initial », démarrer, fonctionner en boucle, s'arrêter

proprement ou dans l'urgence, ... Ces modes, s'ils ne sont pas tous implémentés dans tous les SAP, sont standards et répertoriés dans le GEMMA.

En prêtant attention à cet aspect nous faisons ressortir une classe de scénarios pédagogiques définis une fois et réutilisés, moyennant des adaptations ponctuelles mineures pour « coller » à chaque SAP cible. Un **gain évident de temps** de développement des scénarios liés à la définition de ces modes de marche et à leurs conditions d'évolution est réalisé en réutilisant des ressources retrouvées dans la plupart des cas d'étude d'automatismes industriels.

2.7.2 Problématique 2 : la sécurité (matérielle, logicielle et humaine)

Dans le même esprit, la prise en compte des contraintes de sécurité homme et machine est fondamentale : la législation impose des mesures de **précaution** plus ou moins **drastiques** selon la dangerosité de la manipulation. L'informatisation de la procédure expérimentale peut être perçue comme moyen d'améliorer la sécurité en inhibant par exemple des combinaisons de paramètres dangereuses ou en détectant des gestes inappropriés.

Apprendre à utiliser un système industriel passe aussi par l'apprentissage des gestes préservant l'intégrité de la machine et des personnes qui l'utilisent. Au-delà des scénarios apprenant à **maîtriser** le système en fonctionnement « normal », il faut aussi apprendre à **réagir** (et souvent dans l'urgence) face à des situations inhabituelles ou d'incident. Ainsi, préparer et enrichir progressivement une base de scénarios par type d'incidents repérés ou rencontrés donnera une **valeur ajoutée à notre outil de préparation de TP**.

Toutes les organisations industrielles se doivent de gérer les risques inhérents à leurs activités. La maîtrise des risques, dans les domaines principalement de la sécurité industrielle, de la santé, de la sécurité au travail et de l'environnement, mobilise des acteurs des phases de conception, de construction et tout au long de l'exploitation, en particulier lors du démarrage et de la maintenance des installations et lors de séquences d'incidents, d'accidents ou de crises.

Dans un contexte industriel, nous distinguons trois catégories de sécurité à tenir en compte lors de déroulement d'expérimentations :

- la sécurité **humaine** qui s'intéresse à fournir des mesures de sécurité aux acteurs dans un environnement industriel. Une analyse des phénomènes et événements dangereux

pouvant mettre en péril la sécurité humaine dans un environnement de travail industriel est présentée dans [Doucet et al.-2004], et une étude sur les défaillances des dispositifs de sécurité dans un tel environnement sont discutées dans [Backström-2000] ;

- la sécurité **matérielle** qui s'intéresse à fournir des mesures pour protéger les matériels impliquées dans un processus industriel ;
- la sécurité **informatique** qui représente un aspect indispensable lors de l'utilisation de l'informatique pour piloter les différents processus industriels (fabrication, supervision, testes, etc.). Une étude profonde sur la sécurité des systèmes informatiques dans l'industrie est présentée dans [Stouffer et al.-2011] ;

Face à la problématique de sécurité dans un milieu industriel, il est indispensable de prédire les dangers pouvant menacer l'environnement de travail. La méthode AMDEC⁴⁷, qui appuie sur un raisonnement **inductif** « Causes et Conséquences », pour l'étude organisée des causes, des effets des défaillances et de leur criticité, répond à cette exigence. L'AMDEC établit un Indice de Priorité de Risque (IPR) qui détermine l'opportunité de déclencher des plans d'actions à caractère préventif, et l'ordre de priorité de ces plans.

Trois facteurs interviennent pour évaluer la criticité d'un risque. Chaque facteur est évalué sur une échelle de 1 (le moins pénalisant) à 5 (le plus pénalisant) :

- G = **Gravité** = faibles (1) ou fortes (5) conséquences de l'incident (s'il survient) ;
- D = **Défectabilité** = facilité (1) ou difficulté (5) à déceler l'incident dès qu'il survient ;
- P = **Probabilité** = probabilité faible (1) ou forte (5) pour que cet incident survienne ;
- C = **Criticité** est le produit = $G \times D \times P$.

Un exemple sur l'analyse des risques, leurs causes, leurs criticités, actions préventives proposées et leurs efficacités est présenté dans le Tableau 2.7. La colonne « Code » indique la fonction ou l'opération ayant le potentiel d'un risque d'incident, « Type » indique le type de

⁴⁷ La méthode AMDEC : l'Analyse des Modes de Défaillance Et de Criticité.

risque potentiel, « Descriptif incident » décrit le besoin client non satisfait, « Cause » décrit les causes possibles de l'incident.

Tableau des Risques									
Code	Type	Descriptif incident	Cause	G	D	P	C	Action préventive	Efficacité
Ri1	Client	Besoins client non satisfaits	Besoin client oublié	5	4	3	60	Tableau objectifs qualité §VII RAO	<i>n</i> %
Ri2									

Tableau 2.7 : Structure du tableau des risques selon la méthode AMDEC.

2.7.3 Problématique 3 : la chaîne éditoriale

Nous présentons dans cette section la problématique liée à la chaîne éditoriale, pour un TP sur un Système Automatisé de Production (SAP) complexe, découverte pendant notre observation locale. Actuellement, la préparation de TP sur un SAP complexe est une opération aussi complexe que le système sujet du TP. Il faut prendre en compte plusieurs aspects lors de cette préparation en ce qui concerne les ressources, les intervenants, les cas et les façons d'utilisation de SAP.

2.7.3.1 Gestion des ressources pédagogiques

La préparation d'une session de TP sur au SAP complexe commence par définir le sujet, le(s) système(s) sur le(s)quel(s) se déroule le TP et les objectifs pédagogiques à atteindre à travers cette session. Puis, et en vue de ces définitions, il faut préparer plusieurs types de ressources indispensables pour le déroulement de la session, cela inclut :

- une notice expliquant le système et ses composants ;
- une/plusieurs notices expliquant comment est programmé chaque composant du système (APIs, robots, IHM, etc...), à cela s'ajoute une notice d'usage par des enseignants et des apprenants en automatisme étant donné qu'un système automatisé complexe peut être utilisé par d'autres utilisateurs en autres disciplines que l'automatisme ;
- les programmes API à télécharger sur les automates (composant du système), les programmes robots, programmes d'IHM permettant de contrôler le système. Cette tâche est très importante et prend du temps considérable ;

- une/plusieurs notices expliquant comment faire fonctionner chaque poste (API+robot) et comment les utiliser via leur IHM. Notons que ces notices seront utilisées par des utilisateurs provenant de différentes disciplines ;
- les scénarios d'apprentissage définissant les objectifs pédagogiques à atteindre et aussi les activités pédagogiques à suivre ;
- aux sujets de TP s'ajoutent les corrections à faire et les feuilles de route pour les enseignants.

Cette liste de ressources à préparer montre **l'abondance de documents** divers à rédiger avant le déroulement d'une session de TP. Ces ressources sont de différents types (documentation, scénarios, programmes) et exigent l'intervention de plusieurs intervenants, de différentes spécialités, afin de les réaliser, alors que l'instructeur sera la seule personne présente pendant la session. Il est important de constater à ce stade que l'instructeur n'est pas forcément **familiarisé** avec tous ces types de ressources, surtout la programmation, et nécessiterait donc l'**assistance** soit pour préparer un ou plusieurs types de ces ressources (ou dans la situation la plus courante, au cas où le fonctionnement du système exerce une anomalie lors de sa préparation), ou bien si l'instructeur simplement ne sait pas comment le configurer. Effectivement, les ressources sont développées **à la demande**, pour des sujets et des systèmes spécifiques, d'une façon séparée et sans prendre en compte la possibilité d'une mutualisation entre les acteurs de différentes disciplines. Cette situation révèle la **perte de temps** exercée par les différentes personnes responsables de la préparation des ressources d'une session de TP sur un SAP complexe.

2.7.3.2 Les différents intervenants de TP

Le TP sur un SAP complexe implique l'intervention de plusieurs acteurs afin de garantir la bonne préparation et le bon déroulement (cf. §2.3, voir Figure 2.3), ce sont : l'auteur, le technicien, l'instructeur et l'apprenant. Nous redéfinissons le rôle du technicien (en tant qu'un automaticien) dans le contexte d'automatisation comme le responsable de la maintenance de la plate-forme accueillant les systèmes automatisés sujets de TP. Il joue un rôle très important dans le contexte de TP sur des SAPs complexes. En effet, il s'occupe notamment de la **reconfiguration** des SAP avant le déroulement de TP, ce qui implique notamment de **développer** des programmes API conformes aux objectifs décrits par l'auteur (si l'auteur ou l'instructeur n'est pas spécialiste en automatisme) et de **paramétrer** les machines. Cette tâche

se fera à chaque reconfiguration de SAP soit si le TP implique une seul ou plusieurs configurations pendant la session. La présence de l'automaticien s'avère donc indispensable pour répondre à ce besoin ;

Les différents intervenants doivent coopérer pour assurer la bonne préparation et le bon déroulement du TP en automation.

2.7.3.3 Les façons d'utiliser un SAP dans un TP

Puisque le SAP complexe est généralement exploité pour plusieurs disciplines et donc selon différents objectifs pédagogiques, il est courant d'y trouver différents cas d'utilisation. Nous distinguons notamment deux grandes catégories d'utilisation : à des fins de développement ou d'exploitation (voir Tableau 2.8).

Nous constatons ici que ces deux catégories de cas d'utilisation d'un SAP (développement et exploitation) s'adressent, à la fois, aux différentes disciplines (spécialistes/non spécialistes en automatisme), et aux différents objectifs pédagogiques dans la même discipline. Nous pouvons donc **extrapoler** le nombre de (re)configurations pouvant être appliqués sur un même SAP afin qu'il réponde aux exigences techniques et pédagogiques des TP et, par conséquence, **l'emploi du temps très chargé** pour élaborer les différentes configurations concernées, à ne pas oublier que ces configurations peuvent être modifiées ou mises à jour en correspondance aux besoins potentiellement occurrent.

Cas d'utilisation	Description
Développement	Dans le domaine des automatismes et de la robotique industrielle, les apprenants ont à développer des programmes API à télécharger sur les automates du système, à programmer leur IHM et à programmer le(s) robot(s) faisant partie du système. Ces cas d'utilisation sont adressés aux apprenants spécialistes du domaine et leur rôle est de développer des fonctionnalités pour chaque composant du système.
Exploitation	A l'inverse, les apprenants dans d'autres domaines (en logistique par exemple) que les automatismes (et ceux liés comme la robotique) ont à exploiter le SAP selon d'autres objectifs que le développement. Les cas d'utilisation de ce type peuvent se limiter au paramétrage des composants du système à côté de l'utilisation de ses fonctionnalités, alors que les programmes pilotant ces composants ont déjà été développés et téléchargés sur les APIs.

Tableau 2.8 : Les cas d'utilisation d'un SAP.

Nous pouvons donc répertorier les modes d'utilisation d'un SAP dans deux groupes :

- les tâches à remplir sur le SAP : l'utilisation du SAP pour remplir une tâche donnée varient essentiellement selon les objectifs pédagogiques à atteindre pendant une

session de TP et aussi selon ce qui utilise actuellement le SAP (apprenant ou instructeur). Cela comprend :

- l'utilisation pour **démontrer** les fonctionnalités du système ; la configuration du SAP associé est, dans ce cas, une configuration « démo » qui permet aux apprenants de comprendre (seulement) la logique du système et le but final à atteindre. Cette configuration est adressée à l'instructeur du TP pour lui permettre d'introduire et illustrer le sujet de la session ;
- l'utilisation pour **programmer** des parties du système : dans ce cas, la configuration du SAP associée contient des programmes vides ou incomplets à compléter par les apprenants. Il est courant que la configuration utilisée pour ce cas soit associée à une configuration démo qui joue le rôle de point de départ ;
- l'utilisation pour **paramétrer** des programmes complets : dans ce cas, les apprenants ont à remplir les paramètres des programmes afin d'atteindre certaines fonctionnalités du système selon des objectifs prédéfinis.

2.7.3.4 Interdisciplinarité

Les différences dans les façons d'utiliser un SAP complexe entre spécialistes en automatisme et non spécialistes impliquent de développer différentes configurations qui servent différents buts. Nous constatons ici que ces différentes configurations comportent des parties communes quelle que soit la discipline ou la tâche à atteindre mais, sans l'interopérabilité entre les intervenants, la préparation de ces configurations comporte une perte du temps considérable.

Les différents aspects présentés précédemment démontrent une complexité générale en ce qui concerne la préparation des configurations sur le même SAP complexe. Nous retrouvons des nombreuses ressources et parties communes, toutes spécialisées pour chaque discipline et pour chaque niveau d'utilisation, et **mutualisées d'une manière souvent artisanale** au coup par coup du fait de l'absence d'un outil permettant de les mutualiser d'une manière ordonnée. Cela caractérise donc, dans un premier temps, une **problématique de la réutilisation de ressources** développées au sein des configurations qui, à son tour, empêche de gagner du temps lors de la préparation des configurations pour des cas d'utilisation similaires à ceux déjà existants et, à ne pas oublier, empêche de profiter des expériences menées dans ces cas-là.

D'après notre observation locale, nous avons trouvé que pour préparer une configuration de session de TP, chaque instructeur développe ses propres sujets et documentations en reprenant les ressources apportées par le responsable de la plate-forme gérant un SAP. L'emploi du temps chargé de chaque instructeur, la méconnaissance de ce que réalise son voisin et la forte focalisation de chaque enseignant sur un domaine précis empêche cet échange d'informations, et réduit donc l'opportunité de profiter des ressources partageables entre eux. Nous ne retrouvons donc **pas d'apport collectif** des instructeurs une fois leur sujet ou documentation réalisés ce qui implique sûrement une perte du temps non reconnue par chaque enseignant.

Par ailleurs, la mise à jour d'un SAP est une longue opération nécessitant beaucoup de temps. Par exemple, en ce qui concerne simplement les logiciels de développement de programmes automate, la migration de PL7Pro à UNITY sur le transfert libre, a montré le besoin de sections de programme API de bas niveau communes à tous les acteurs. Ces sections sont verrouillées et modifiées en fonction des pannes ou des mises à jour logicielles et matérielles du SAP **uniquement par le responsable de la plate-forme** afin d'améliorer la cohérence des programmes tournant sur cette plate-forme.

2.7.4 Problématique 4 : de la reconfiguration d'un SAP

Dans, cette section nous présentons la problématique de reconfiguration découverte lors de notre observation locale. L'AIP-Primeca-RAO mutualise des ressources matérielles et logicielles pour les disciplines techniques ou scientifiques (automatismes industriels, logistique, maintenance, ...) impliquées dans l'exploitation de la plate-forme. Actuellement, la plupart des ateliers similaires possède des SAP complexes, cette complexité provient de la variété des thèmes techniques et pédagogiques pouvant être menés lors d'un TP ou un projet par les intervenants de différentes disciplines.

Pour illustrer la complexité de cette structure ainsi que les technologies liées à un SAP pédagogique, revenons sur le système du transfert libre présenté dans la section précédente §2.6.2. Les besoins de reconfiguration s'appliquent sur les différents composants du système en exigeant une reprogrammation et un paramétrage généraux, et nécessitent donc une expertise sur l'ensemble du système, une expertise que tous les instructeurs intervenant sur cette plate-forme ne détiennent pas toujours, notamment ceux qui ne sont pas spécialistes en automatisme (par exemple : instructeurs en logistique). Cette situation implique donc une **dépendance sur une personne spécialiste** qui possède toutes les compétences nécessaires

pour mener une reconfiguration du système afin de préparer les manipulations. L'absence ou l'indisponibilité de cette personne risque de bloquer cette organisation à tout moment, et donc empêcher a fortiori les enseignements pratiques en cas d'imprévu (une panne ou une anomalie de fonctionnement).

À cela s'ajoute la problématique de la coordination entre les différents établissements faisant partie du partenariat de la plate-forme. Actuellement, les emplois du temps de chaque institut imposent des contraintes qui créent un agenda d'utilisation non optimisé pour minimiser les phases de reconfiguration. Les partenaires étant eux-mêmes contraints par ailleurs dans leur organisation, la contrainte consistant à limiter le nombre de reconfigurations successives en groupant les enseignements utilisant une même configuration n'est pas prépondérante. La solution est donc plutôt du côté de **l'optimisation des reconfigurations** afin de limiter les pertes de temps inévitables.

L'exemple du MES (cf. §2.6.2.1), illustre la taille de travail à effectuer pour reconfigurer le système de transfert libre afin d'y dérouler différentes activités pédagogiques. La complexité ne vient pas seulement du grand nombre de ressources à préparer, mais aussi de la sélection entre ces ressources pour atteindre les objectifs pédagogiques désignés pour chaque session de TP.

2.7.5 Besoins identifiés en automatique

Pendant notre observation dans l'atelier AIP RAO Primeca, nous avons noté que le processus de (ré)configuration d'un SAP, dans le but de préparer une session de TP, est une longue opération. Une **procédure rigoureuse** doit être suivie : chargement des APIs et les programmes des robots industriels, réglage des aspects mécaniques, remplissage des stocks, redémarrage des robots et les systèmes automatisés liés, etc. Cette procédure varie selon la session de TP et ses objectifs pédagogiques, le niveau des apprenants, la discipline enseignée. Cette procédure exige un niveau élevé d'expertise pour la totalité du système manipulé, une expertise qui n'est pas nécessairement disponible chez l'instructeur (par exemple : un instructeur en Génie Industriel n'est pas spécialiste en automatisme). Le technicien est généralement la personne ayant l'expertise nécessaire, il fait les opérations requises pour configurer un système afin d'y mener un TP. Cette personne s'occupe de cette procédure dans les cas où l'instructeur n'est pas autonome (non spécialiste en automatisme, instructeur temporaire ou nouveau), et il doit se présenter parmi les utilisateurs de la plate-forme.

L'absence ou l'indisponibilité de cette personne peut bloquer la plate-forme ou limiter son exploitation en termes de temps ou de fonctionnalités.

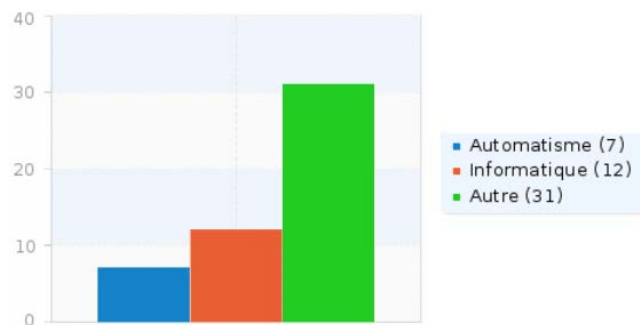


Figure 2.11 : Enquête: domaines d'enseignement des contributeurs.

Pour mettre en lumière les difficultés rencontrées lors de la préparation de session de TP, nous avons lancé une enquête destinée aux instructeurs des universités dans les disciplines techniques et scientifiques. 47 contributeurs ont participé à cette enquête dont 15% d'entre eux sont dans le domaine d'automatisme, et 25% sont dans le domaine d'informatique, et les autres sont de différentes disciplines (Génie industriel, Génie Electrique, Electronique, traitement de signal, Mécanique et Mécanique Productique, Plans d'Expériences, Génie des Procédés, Chimie et Chimie analytique, Physique, Sciences de gestion, Résistance Des Matériaux, Mécaniques des solides, Cinématique, Mathématique numérique, conception, thermique) (Figure 2.11). Cette enquête a révélé que 36% des contributeurs ont besoin entre **15 à 60 minutes** pour préparer une session de TP (voir Figure 2.12), et 72% d'entre eux ont **besoin d'un technicien** (voir Figure 2.13) (même en automatisme, 5 contributeurs à cette enquête ont déclaré d'avoir besoin de l'aide d'un technicien).

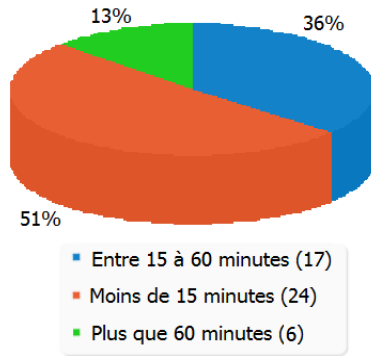


Figure 2.12 : Enquête : temps pour configurer une session de TP.

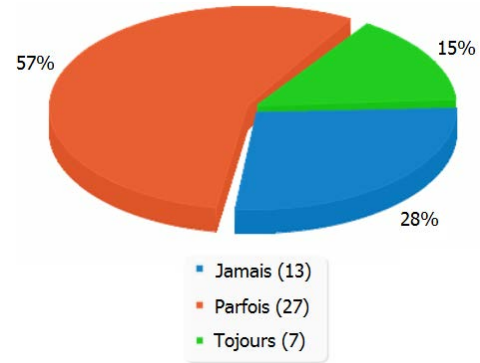


Figure 2.13 : Enquête : instructeurs ayant besoin de l'aide d'un technicien.

Le partenariat de plusieurs établissements et le partage lourd des ressources dans l'atelier « AIP RAO Priméca » imposent un **calendrier mélangé**. Les configurations de chaque matériel doivent être régulièrement modifiées entre deux sessions, et leur usage intense impose aussi de réduire le temps nécessaire pour les configurer. En plus de la taille de travail requis pour configurer les matériels d'un système, nous avons observé un autre **souci concernant la collaboration** entre les instructeurs. Actuellement, la manière courante de préparer les ressources de TP se résume par le fait que chaque instructeur développe ses propres programmes sans avoir une vue globale concernant qui développe quoi, ou qui utilise le système pour quel usage, et l'échange d'expériences s'avère être difficile. L'échange et la gestion des ressources pédagogiques, comme les documentations et les scénarios pédagogiques, est aussi limité à cause de cette façon de travailler. Selon notre enquête, la **moitié** des contributeurs ont déclaré leur **intérêt pour une amélioration de la réutilisabilité des ressources pédagogiques** et pour trouver un moyen de mieux les gérer. 83% des contributeurs réutilisent leurs propres ressources alors que 63% d'entre eux réutilisent celles des autres collègues et la moitié d'entre eux croient que les configurations qu'ils ont développé peuvent être réutilisées (voir Figure 2.14). Finalement, un avis favorable a été exprimé par les contributeurs sur l'utilisation d'une bibliothèque de configurations dont le but est d'archiver ces configurations et de permettre leur réutilisation, soit par les personnes qui les ont générés, soit par d'autres collègues (voir Figure 2.15).

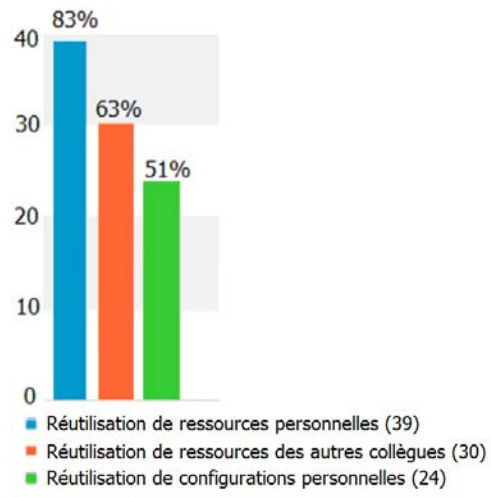


Figure 2.14 : Enquête : réutilisation et gestion de ressources.

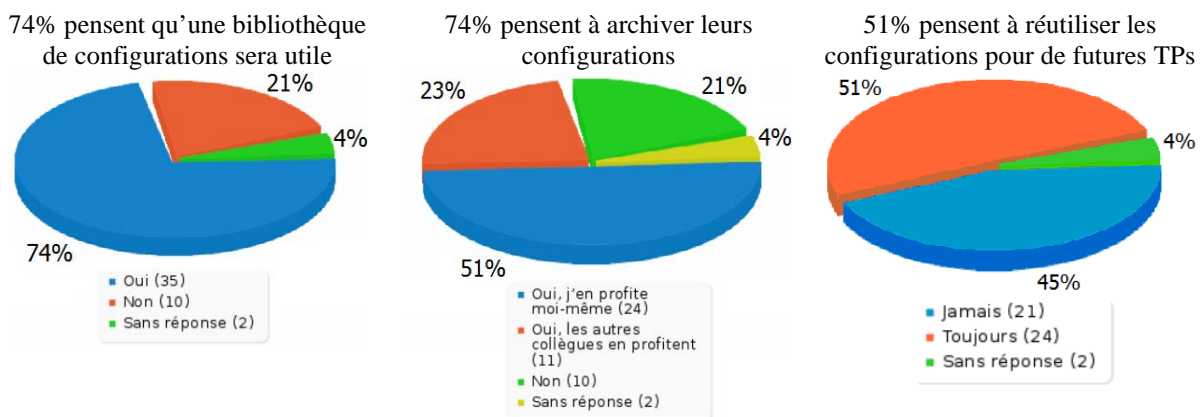


Figure 2.15 : Enquête : archivage et gestion des configurations via une bibliothèque de configuration.

Notre observation a montré qu'il y a un intérêt dans deux points majeurs. D'abord, **l'archivage de ressources** y compris leur indexation, ce qui donne aux différents intervenants (instructeurs/auteurs) de TP une vue plus globale sur les expertises menées dans une discipline donnée et permet de réutiliser ces ressources et donc **gagner plus de temps**, voir les adapter aux contextes d'utilisations similaires à ceux d'origine. Le deuxième point représente l'intérêt à **automatiser le processus de configuration d'un SAP** (ou bien, automatiser le maximum d'opérations du processus de configurer un SAP), ce qui **réduit la dépendance** de l'instructeur à un technicien, d'une part, et permet de réduire le temps nécessaire pour reconfigurer un SAP entre deux sessions et améliorer donc les conditions de partage de ressources matérielles dans l'atelier entre les différents partenaires, d'une autre part.

2.8 Objectifs de ce travail de recherche

Ce chapitre a présenté les apports de l'utilisation des STIC dans les différents types d'enseignement, avec un zoom sur l'objet de notre étude : les travaux pratiques. À travers une étude d'un environnement local propice (le transfert libre de l'AIP Primeca RAO), nous avons

mis en valeur des problématiques rencontrées au jour le jour sur le terrain. Nous avons donc développé une recherche visant à améliorer la qualité pédagogique des travaux pratiques à travers un outil informatique : donc d'un e-TP. Parmi tous les problèmes rencontrés, voici ceux que nous avons décidé de traiter :

- **la gestion des ressources pédagogiques** : comprend la préparation des programmes API pour un SAP, des documentations pour les apprenants et l'instructeur (liées au sujet) et la préparation des scénarios pédagogiques. Nous y répondrons en proposant un outil de gestion assurant :
 - l'archivage et l'indexation des ressources pédagogiques à travers une bibliothèque dédiée (programmes, documentations et scénarios pédagogiques), avec les fonctionnalités de recherche. Cela devra permettre la réutilisation de ressources, et aussi la collaboration entre les instructeurs de différentes disciplines, et réduire le temps requis pour développer des ressources ;
 - la génération de configurations pour un SAP à partir d'une bibliothèque de composants automatés ;
 - la génération automatique de configurations d'une session d'e-TP sur un SAP donné, y compris l'aspect technique lié au SAP et l'aspect pédagogique lié aux documentations et scénarios d'apprentissage ;
- **la configuration d'un SAP** : comprend le développement de plusieurs versions de programmes APIs d'un SAP donné pour différentes disciplines, différents objectifs pédagogiques et différents contextes d'utilisation et de fonctionnement. Il en résulte un questionnement sur l'autonomie/dépendance de l'instructeur pendant la session de TP : peut-il s'affranchir de la présence d'un technicien ?
- **la réutilisabilité des ressources** : comment organiser une collaboration entre les instructeurs de différentes disciplines qui utilisent les mêmes SAP ? Nous répondrons à ces deux dernières questions en proposant une chaîne éditoriale pour faciliter la préparation pédagogique et technique du e-TP.

3. Le cadre de modélisation du système

Introduction

Ce chapitre présente la conception de notre système, les outils utilisés, ainsi que les travaux de recherche sur lesquels nous nous sommes appuyés. **L’approche objet, classique en informatique**, s’immisce progressivement dans le domaine de l’automatique. Elle a été le fil conducteur de notre phase de conception. Une étude bibliographique sur cette approche et son application en automatique est disponible en annexe A.

Nous nous appuyons sur une architecturation guidée par les modèles (MDA = *Model Driven Architecture*), démarche de modélisation et développement de logiciels proposée et soutenue par l'OMG⁴⁸. La MDA sépare la spécification d’un système des détails de son implémentation sur une plate-forme particulière. Le modèle de domaine d’un système particulier est abstrait : il s’agit du « modèle indépendant de plate-forme » (PIM⁴⁹). Le modèle d’un système pour une plate-forme donnée est ensuite défini sous la forme d’un « modèle spécifique de plate-forme » (PSM⁵⁰) [Jin-2006] (voir Annexe B). Dans notre cas, la distinction entre ces deux phases de modélisation évite, lors de la phase de conception, de nous contraindre dans des choix technologiques le plus longtemps possible. Les choix technologiques résulteront de besoins intermédiaires générés en sortie de cette première phase de spécification. Concomitamment, nous avons adopté le langage de modélisation de systèmes SysML comme outil de conception. Il est compatible avec la démarche MDA du fait de la réutilisation dans SysML des normes d’UML [Turki-2008].

Dans ce chapitre nous présentons d’abord les travaux à l’origine de notre recherche : le premier se situe dans le domaine de la téléformation (plus précisément, dans le domaine des téléTP), le second dans le domaine de la conception des systèmes automatisés. Le croisement de ces deux travaux est le point de départ de notre contribution. Après présentation de l’approche méthodologique adoptée, nous spécifions le système, puis en détaillons la structure. Nous clôturons ce chapitre par une analyse critique du modèle élaboré.

3.1 Travaux à l’origine de notre recherche

Notre recherche s’inspire principalement de deux travaux datant de 2007.

⁴⁸ OMG « Object Management Group » : <http://www.omg.org/>

⁴⁹ PIM : Platform Independent Model.

⁵⁰ PSM : Platform Specific Model.

La première recherche, [Benmohamed-2007] concerne l'environnement de génération et d'exécution de scénarios de téléTP. Elle propose un **modèle générique de téléTP** en sciences de l'ingénieur accompagné d'une méthodologie de mise à distance de dispositifs technologiques, dans un souci de généricité et de réutilisabilité des ressources. Ce travail porte principalement sur l'étude d'un système ouvert d'édition (mis à disposition de l'auteur formateur) et d'exécution (par les apprenants et le tuteur) de **scénarios pédagogiques** pouvant être réutilisés sur des dispositifs fonctionnellement similaires. La plate-forme de téléTP ainsi conçue est réutilisable quelle que soit la matière scientifique (physique, chimie, automatique, productique, ...) et adaptable en fonction des objectifs pédagogiques recherchés par le formateur et des contraintes liées au dispositif technologique utilisé. Il a également été prouvé que ce système s'intègre de manière homogène dans un environnement pédagogique général comprenant notamment téléCours, téléTD, téléProjets. Ces travaux de thèse ont conduit à élaborer un ELAMS⁵¹ assurant l'interaction entre un LMS et les maquettes pédagogiques d'e-TP. Ainsi, un auteur peut distribuer ses scénarios à des tuteurs travaillant avec des systèmes similaires, i.e. identiques en terme de fonctionnalités mais implémentés différemment (oscilloscopes, automates, logiciels de modèles et/ou marques différentes, ...). L'adaptation d'un scénario pédagogique s'effectue à l'aide d'ELAMS qui fournit les adresses de fonctionnalités d'une maquette cible et les rend accessibles par le scénario adapté.

La deuxième recherche, [Chiron-2007] améliore la flexibilité et à la rapidité de conception des systèmes automatisés par l'utilisation d'UML et répond à une problématique induite par les tendances actuelles en matière d'organisation du travail, à savoir comment augmenter l'efficacité des phases de conception pour qu'elles conduisent, d'une part, à des mises en œuvre plus rapides au niveau de la production en aval et, d'autre part, à une réutilisation future. Une approche « composants » a été proposée pour la conception des systèmes automatisés, plus précisément pour le pilotage des systèmes de convoyage. Une méthodologie assurant la génération automatique des programmes de contrôle, fournit un moyen d'obtenir un code source compatible avec le standard IEC 61131-3. Cette approche vise à réduire le temps requis pour concevoir la partie de contrôle, dans le contexte des systèmes de convoyage, et à simplifier la création de contrôles dans le contexte de la reconfiguration, étant donné qu'il est nécessaire de fournir plusieurs versions de ces contrôles.

⁵¹ ELAMS : Electronic Laboratory Management System.

L'intérêt de la première recherche [Benmohamed-2007] est la chaîne éditoriale proposée, l'ontologie générique pour la description des maquettes pédagogiques et « ELaMS », le *middleware* développé pour adapter un scénario pédagogique générique dans un ELab. L'intérêt de la deuxième recherche [Chiron-2007] est la méthodologie de génération automatique de code source compatible avec le standard IEC 61131-3 à partir de blocs UML.

3.2 Choix méthodologiques et techniques lors de la conception

La démarche MDA⁵² a l'avantage de reporter au maximum le choix du mécanisme d'implémentation. Elle s'organise autour d'un modèle « PIM » (*Platform Independent Model*), qui représente les spécifications d'un système indépendamment d'une plate-forme. Ces spécifications s'adressent au modèle indépendant CIM de calcul (*Computational Independent Model*) qui se concentre sur les besoins auxquels doit répondre le système. Ce système peut être compilé vers une plate-forme donnée en transformant le PIM en un « PSM » (*Platform Specific Model*). Le PSM décrit comment le système utilise un type particulier de plate-forme [Deelstra et al.-2003]. C'est avec cette démarche que nous avons développé des modèles génériques afin de simplifier l'implémentation sur différentes plates-formes.

Le langage de modélisation SysML est une extension d'UML, prenant en compte simultanément les aspects matériels et logiciels. Cet outil répond bien à nos besoins, le système à développer comportant une partie logicielle (programmes, documentations, etc.) et une partie matérielle (dispositifs et maquettes pédagogiques). SysML, qui supporte la MDA par la réutilisation des normes connexes d'UML [SysML-2007], peut être utilisé pour réaliser les modèles MDA [Turki-2008].

Abordons la phase de modélisation et conception du système, en adoptant l'approche MDA. Nous modélisons d'abord un modèle CIM pour produire un modèle PIM de notre système. Le choix d'une plate-forme sera présenté dans le chapitre suivant.

3.3 Modélisation et conception du système

Nous présentons ici les besoins du système à concevoir, en partant de la situation existante, en délimitant ses contours, en intégrant les besoins décrits par les intervenants principaux et en précisant les moyens utilisés pour le développer, l'approche adoptée et les objectifs de qualité proposés dans le cadre de ce projet.

⁵² Voir l'annexe B-3.2

3.3.1 Modélisation des exigences : modèle CIM⁵³

Il s'agit ici de développer un système informatique de (re)configuration d'un Système Automatisé de Production (SAP) afin de préparer puis dérouler une session d'e-TP. Ce système est mis à la disposition de l'auteur et de l'instructeur d'e-TP ; il en découle des exigences (besoins et contraintes) « brutes » décrites par eux et qui ont été énumérées dans la section §2.7.5. La première étape consiste à « décortiquer » (sourcer, hiérarchiser, prioriser) ces exigences afin de pouvoir les transformer en spécifications.

3.3.1.1 Synthèse des besoins exprimés par les utilisateurs directs :

Il s'agit des besoins recensés lors de l'enquête :

- **Besoins Auteur (BA)** : les auteurs de scénarios pédagogiques (de TP d'automatique industrielle) interrogés veulent avoir à disposition **un seul et même outil** intégrant la création et l'assemblage de leurs ressources pédagogiques, dans le but de créer des sessions d'e-TP sur mesure. Cet outil devra donc :
 - **BA₁** : créer, manipuler et mémoriser de manière pérenne des ressources pédagogiques (programmes API, documentations, scénarios pédagogiques, notices, etc.) ;
 - **BA₂** : retrouver des ressources selon des critères de recherche, depuis n'importe quel ordinateur ;
 - **BA₃** : réutiliser les ressources développées dans un contexte identifié similaire à celui de leur création afin de réduire le temps de création de nouvelles configurations de sessions ;
 - **BA₄** : générer des configurations de session d'e-TP en assemblant des ressources pédagogiques.
- **Besoins Instructeur (BI)** : les instructeurs interrogés attendent un outil disposant d'une **bibliothèque de ressources pédagogiques** et capable de **configurer automatiquement** le matériel industriel utilisé lors d'une session d'e-TP sur un SAP donné. Les fonctionnalités attendues sont :

⁵³ CIM « *Computational Independent Model* » (voir Annexe B) : il se concentre sur les besoins auxquels doit répondre le système plutôt que sur les moyens qui seront mis en œuvre pour y parvenir, il s'agit donc de modéliser les spécifications. Le CIM centralise les informations relatives aux services que devra implémenter le système et implique la participation des futurs utilisateurs du système pour la spécification des besoins et des cas d'utilisation. Ce modèle est également un moyen de communication entre les personnes qui vont concevoir le système, et les personnes exprimant des besoins sans la moindre idée de la façon dont il faut les implémenter.

- **BI₁** : chercher et rendre opérationnelles des configurations (déjà développées) correspondant aux objectifs pédagogiques formulés par l'instructeur ;
- **BI₂** : simplifier et réduire le temps d'installation d'une configuration;
- **BI₃** : améliorer l'autonomie de l'instructeur ;
- **BI₄** : réduire la dépendance de l'instructeur au technicien.

3.3.1.2 Analyse de la situation existante en fonction des besoins

Pour situer les besoins des utilisateurs, une analyse causale (voir [Robertson-2006], ch.16) de la situation existante s'avère nécessaire pour estimer les limitations liées à la façon d'exploiter les ressources de l'atelier « AIP-Primeca-RAO » dans un contexte d'e-TP. La technique utilisée pour structurer l'analyse causale est le « *Fishbone* » (ou Ishikawa, voir Figure 3.1) qui représente un arbre de dépendances « cause-effet ». La racine représente la mesure d'efficacité sur la façon d'exploiter un SAP du point de vue de chaque utilisateur, et les nœuds représentent des propriétés pouvant influencer cette mesure. Nous utilisons la classification des « **5M** »⁵⁴ pour regrouper les causes. Notre analyse a mis en exergue 3 aspects critiques pour (nous utilisons les 5M pour chacune d'eux) :

- **autonomie de l'instructeur** : lorsqu'il n'est pas spécialiste en automation l'instructeur manque d'autonomie lors de la reconfiguration d'un système automatisé ;
- **temps de configuration** : la configuration d'un SAP exige un temps qui varie selon le TP à y dérouler. L'optimisation de ce temps s'avère nécessaire du fait de la non coordination des emplois du temps entre partenaires ;
- **réutilisation des ressources** : la réutilisation des ressources développées par d'autres auteurs n'est pas facilitée (absence d'un système de gestion des différents types de ressources) ou encouragée (les auteurs travaillent plutôt de manière solitaire et ont tendance à ne réutiliser que leurs propres ressources).

⁵⁴ 5M : la méthode 5M est une classification qui sert à répertorier les différentes causes possibles d'un problème. Les 5M sont : Matière, Milieu, Méthodes, Matériel et Main d'œuvre. Voir : <http://www.logistiqueconseil.org/Articles/Methodes-optimisation/5m-ishikawa.htm>

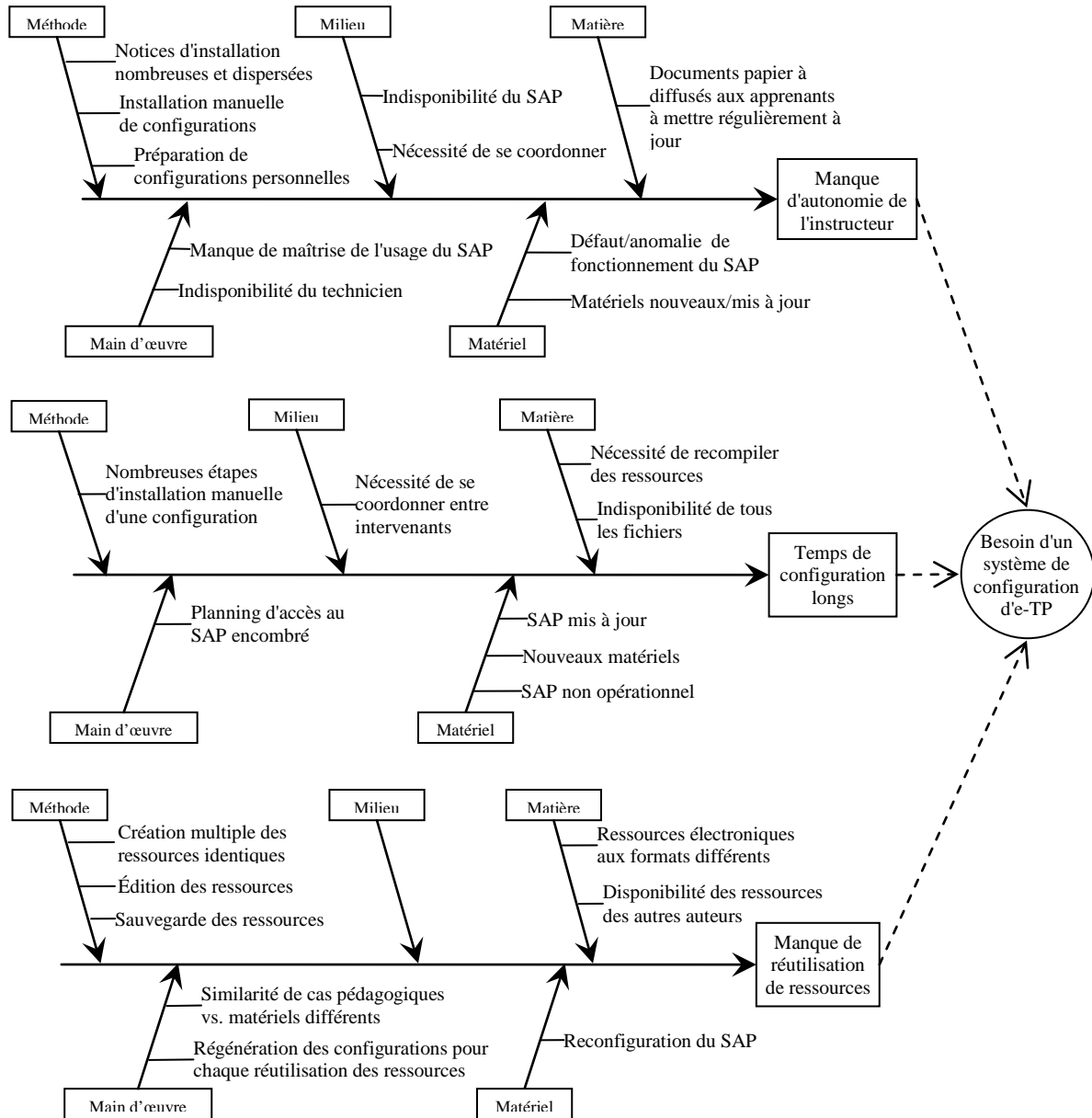


Figure 3.1: Analyse causale de la situation existante illustrant les freins qui motivent ces travaux.

Du point de vue système, une limitation est le résultat d'une exigence non satisfaite traduit par l'absence d'une fonctionnalité. Pour que les spécifications du système soient bien construites, l'étape prochaine consiste à extraire les exigences du système en se basant sur les besoins décrits par les utilisateurs et sur l'analyse de la situation existante. Mais auparavant, développons un diagramme d'activité illustrant le comportement attendu du système en fonction des besoins (voir Figure 3.2).

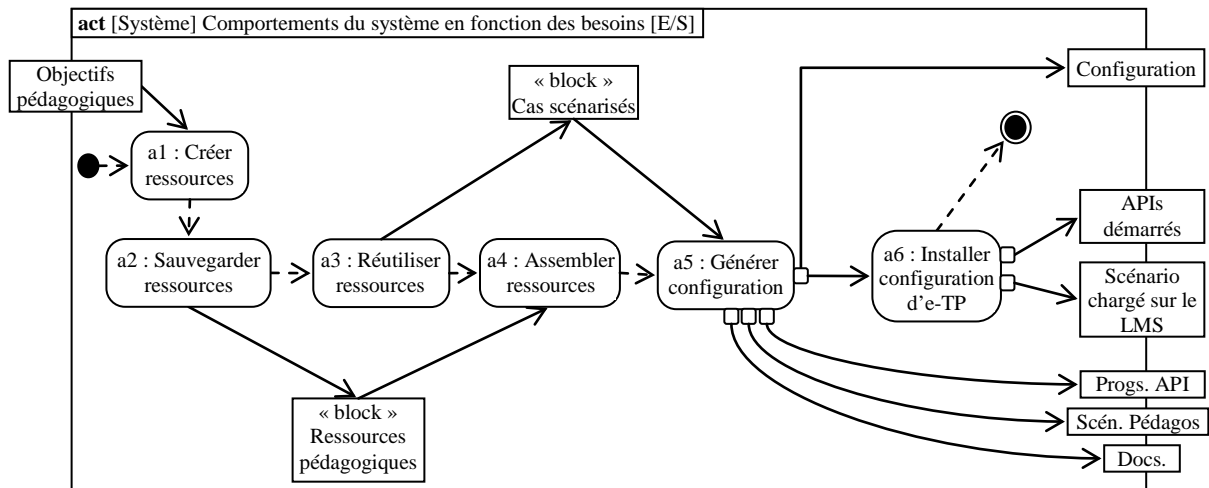


Figure 3.2 : Schéma comportemental du système en fonction des besoins des utilisateurs.

La « sortie » de ce diagramme est un ensemble de programmes API, documentations et scénarios pédagogiques prêts à exploiter dans une session d’e-TP sur un SAP donné, ainsi qu'une configuration (qui regroupe de manière logique l'ensemble des autres sorties), prête à être utilisée pour une installation ultérieure.

L'analyse de ce schéma révèle des exigences non exprimés par les utilisateurs interrogés (voir Figure 3.3) :

- **BE₁** : les utilisateurs devront pouvoir travailler sur des **ressources communes** (documentation, programmes API, configurations, ...) sans qu'il y ait de conflit. Cette exigence concerne l'auteur ;
- **BE₂** : les utilisateurs devront savoir s'ils travaillent avec une **version** aboutie, révisée ou obsolète d'une ressource ou d'une configuration. Cette exigence concerne l'auteur et l'instructeur ;
- **BE₃** : connaître l'**historique** d'un document. Cette exigence concerne l'auteur et l'instructeur ;
- **BE₄** : les auteurs devront pouvoir **sécuriser** l'accès aux documents en fonction des profils d'utilisateurs. Cette exigence concerne l'auteur ;
- **BE₅** : gérer la **maturité des ressources** (brouillon, en cours de modification, publié, obsolète, ...). Cette exigence concerne l'auteur ;

- **BE₆** : gérer les **versions** (Génération, Révision, Version, Itération ...). Cette exigence concerne l'auteur et l'instructeur ;
- **BE₇** : gérer les **dépendances** : signaler les impacts d'un changement d'une ressource sur les autres ressources, scénarios, configurations. Cette exigence concerne l'auteur et l'instructeur ;
- **BE₈** : tracer les **modifications** effectuées sur chaque ressource (qui, quand, pour quoi faire). Cette exigence concerne l'auteur et l'instructeur ;
- **BE₉** : gérer les permissions d'**accès aux données**. Cette exigence concerne l'auteur ;
- **BE₁₀** : **compiler les ressources** d'une configuration afin de les rendre utilisable sur la plate-forme d'e-TP. Cette exigence concerne l'instructeur

Ces exigences sont intégrées dans un diagramme d'exigences global pour la spécification du système (cf. §3.3.1.3).

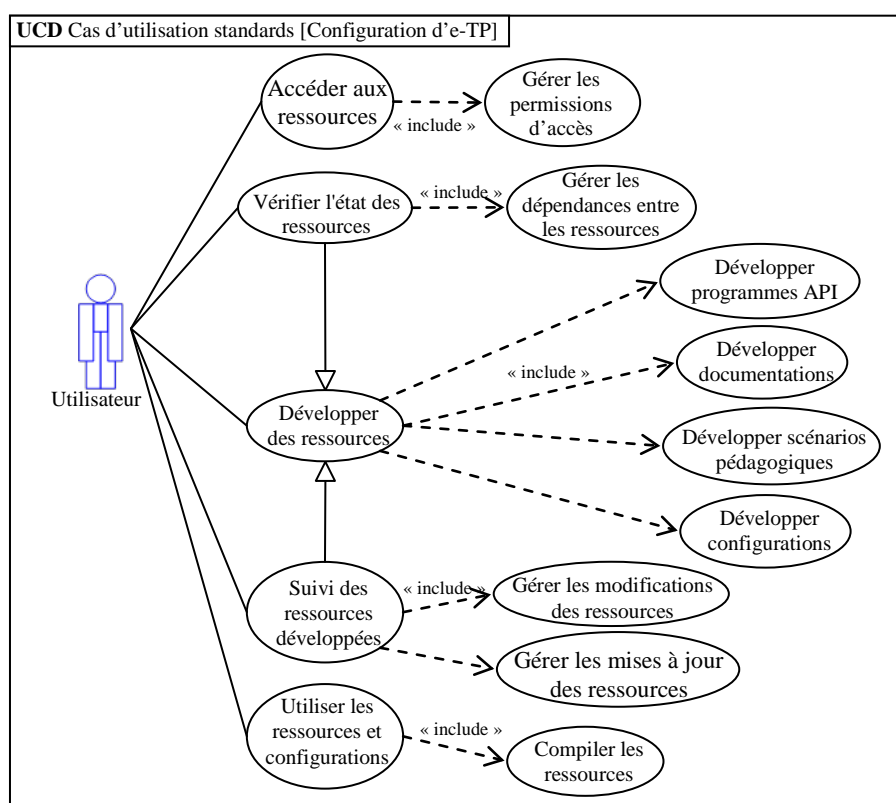


Figure 3.3 : Les cas d'utilisation standards pour configurer un e-TP, extraits de l'analyse de situation existante.

3.3.1.3 Extraction des exigences du système

Compte tenu des besoins décrits par les utilisateurs (cf. §3.3.1.1) et de la situation existante (cf. §3.3.1.2), nous extrayons un ensemble d'exigences en associant une référence à chacune pour assurer la traçabilité des diagrammes ultérieurs. Il s'agit ici de fournir une spécification du configurateur. Cette spécification sera le point de départ pour élaborer la structure du système et son fonctionnement.

Nous évitons ici les trois faiblesses identifiées dans [Simon et al.-2008]: « *penser en solution* », « *repérer les détails* » et « *manquer de raisonnement* ». En conséquence, nous tentons plutôt dans cette étape de fournir une vue globale et hiérarchique des exigences du système. Cette hiérarchie n'inspire pas une solution à implémenter, mais catégorise les fonctionnalités et/ou services à remplir. Nous associons un raisonnement à chaque exigence d'une part pour rendre compréhensible le besoin réel, et d'autre part pour rendre logique la hiérarchie des exigences.

Les exigences sont d'abord listées dans une table illustrant leurs descriptions, raisonnement et besoins d'origine.. Nous transformons ensuite cette table en diagramme d'exigence afin de faire passer les besoins initiaux d'une description subjective à une spécification.

Compte tenu du schéma comportemental du système (voir Figure 3.2) qui décrit de manière générale les fonctionnalités à fournir, nous avons identifié trois catégories d'exigences à satisfaire (voir Tableau 3.1 et Figure 3.4) :

1. création et édition des ressources pédagogiques qui représentent les briques de construction de toute configuration de session d'e-TP sur un SAP. Ces ressources sont : programmes API, scénarios pédagogiques et documentations. Cette exigence est extraite du besoin auteur **BA₁**, et de l'analyse de l'existant **BE₁** ;
2. exploitation des ressources en favorisant leur réutilisation par l'auteur, et en utilisant les configurations disponibles. Cette exigence est extraite des besoins auteur **BA₂** et **BA₃**, du besoin instructeur **BI₁** ;
3. génération de configuration exploitable dans une session d'e-TP à partir des ressources disponibles. Cette exigence est extraite du besoin auteur **BA₄**.

Exigence	Réf	Source	Description	Raisonnement	Origine
----------	-----	--------	-------------	--------------	---------

Création / Edition des ressources	EX1	Auteur	Le configurateur aide l'auteur à créer/modifier les ressources pédagogiques (programmes, documents, scénarios) des configurations destinées aux instructeurs.	Les ressources pédagogiques au sein du configurateur sont indispensables pour générer des configurations destinées aux instructeurs.	BA ₁ BE ₁
Génération de configurations	EX2	Auteur	Le configurateur aide l'auteur à construire des configurations à partir des ressources prêtes à l'emploi.	Un nombre élevé de ressources différentes par nature et en version nécessite un outil dédié à leur compilation en vue de les exploiter dans un e-TP.	BA ₄
Exploitation des ressources	EX3	Auteur / Instructeur	Le configurateur aide l'auteur à (ré)utiliser les ressources existantes pour générer de nouvelles configurations. Il aide aussi l'instructeur à profiter de ces ressources ainsi que des configurations existantes pour configurer une session d'e-TP.	Le fait de permettre aux auteurs de (ré)utiliser des ressources et à l'instructeur, des configurations est crucial pour remplir l'objectif global d'améliorer les conditions de préparation et démarrage des sessions d'e-TP.	BA ₂ BA ₃ BI ₁

Tableau 3.1 : Exigences principales à remplir par le système de configurateur.

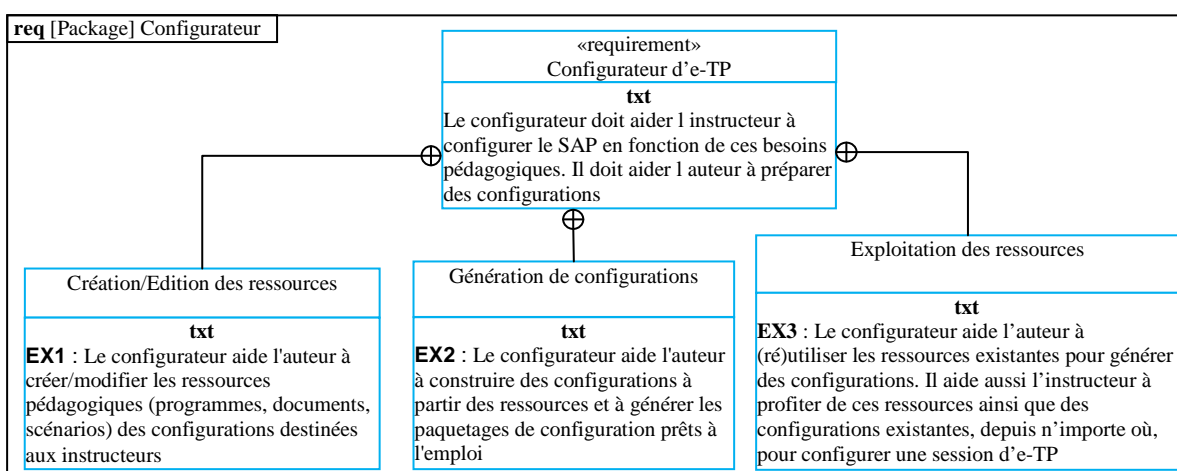


Figure 3.4 : Diagramme d'exigences du configurateur affichant les exigences principales du système.

Partons du diagramme de la Figure 3.4 pour élaborer une hiérarchie nous aidant à développer la structure du système (phase suivante). Nous reprenons chaque exigence principale pour identifier ses sous-exigences.

1. Sous-exigences de EX1 « Création/Édition des ressources » :

Cette exigence concerne l'auteur. Elle correspond à la phase préparatoire du processus de configuration d'une session d'e-TP. Il s'agit de créer et éditer non seulement les différents types de ressources pédagogiques, mais aussi le cycle de vie de ces ressources. Les sous-exigences de EX1 sont présentées dans le Tableau 3.2 et la Figure 3.5.

Exigence	Réf	Source	Description	Raisonnement	Origine
Sauvegarde des ressources	EX1 ₁	Auteur	Le configurateur aide l'auteur à mémoriser de manière pérenne les ressources créées	La disponibilité permanente des ressources pédagogiques simplifie leur réutilisation et enrichissement	BA ₁
Accès aux ressources	EX1 ₂	Analyse de l'existant	Le configurateur aide les auteurs à travailler sur des ressources	Toute modification d'une ressource, non finalisée, par d'autres auteurs	BE ₁

			communes sans conflit	risque de perdre l'objectif de ressource originale	
Gestion des permissions	EX1 ₃	Analyse de l'existant	Le configurateur aide les utilisateurs à accéder aux ressources avec des permissions garantissant leur sécurité	L'accès concurrent à une ressource par plusieurs auteurs sans les bons droits risque d'endommager le contenu de ressource	BE ₄ BE ₉
Vérification de l'état des ressources	EX1 ₄	Analyse de l'existant	Le configurateur aide l'auteur/instructeur à vérifier l'état des ressources (version actuelle, modifications, disponibilité pour utilisation, etc.)	L'utilisation d'une ressource pédagogique en cour de finalisation, ou bien la mauvaise version peut engendrer incohérence avec l'objectif du TP	BE ₂ BE ₃ BE ₅ BE ₆
Gestion des dépendances	EX1 ₅	Analyse de l'existant	Le configurateur aide l'auteur/instructeur à s'informer des impacts d'un changement de ressource sur les autres (scénarios, configurations)	La création des ressources s'enchaîne éventuellement pour générer une configuration. La chaîne de doit pas se détacher	BE ₇
Gestion de modification des ressources	EX1 _{5_1}	Analyse de l'existant	Le configurateur informe l'auteur/instructeur de l'impact d'une modification ou des répercussions d'une nouvelle version	La modification d'une ressource peut impliquer de modifier d'autres ressources	BE ₈
Gestion et mises à jour des ressources	EX1 _{5_2}	Analyse de l'existant	Le configurateur avertit l'auteur/instructeur de l'évolution d'une ressource créée en fonction de l'évolution à son tour des ressources qu'elle utilise	La mise à jour d'une ressource peut nécessiter de mettre à jour d'autres pour garder la cohérence entre elle	BE ₇ BE ₈

Tableau 3.2 : Liste des sous exigences de EX1 « Création/Édition des ressources ».

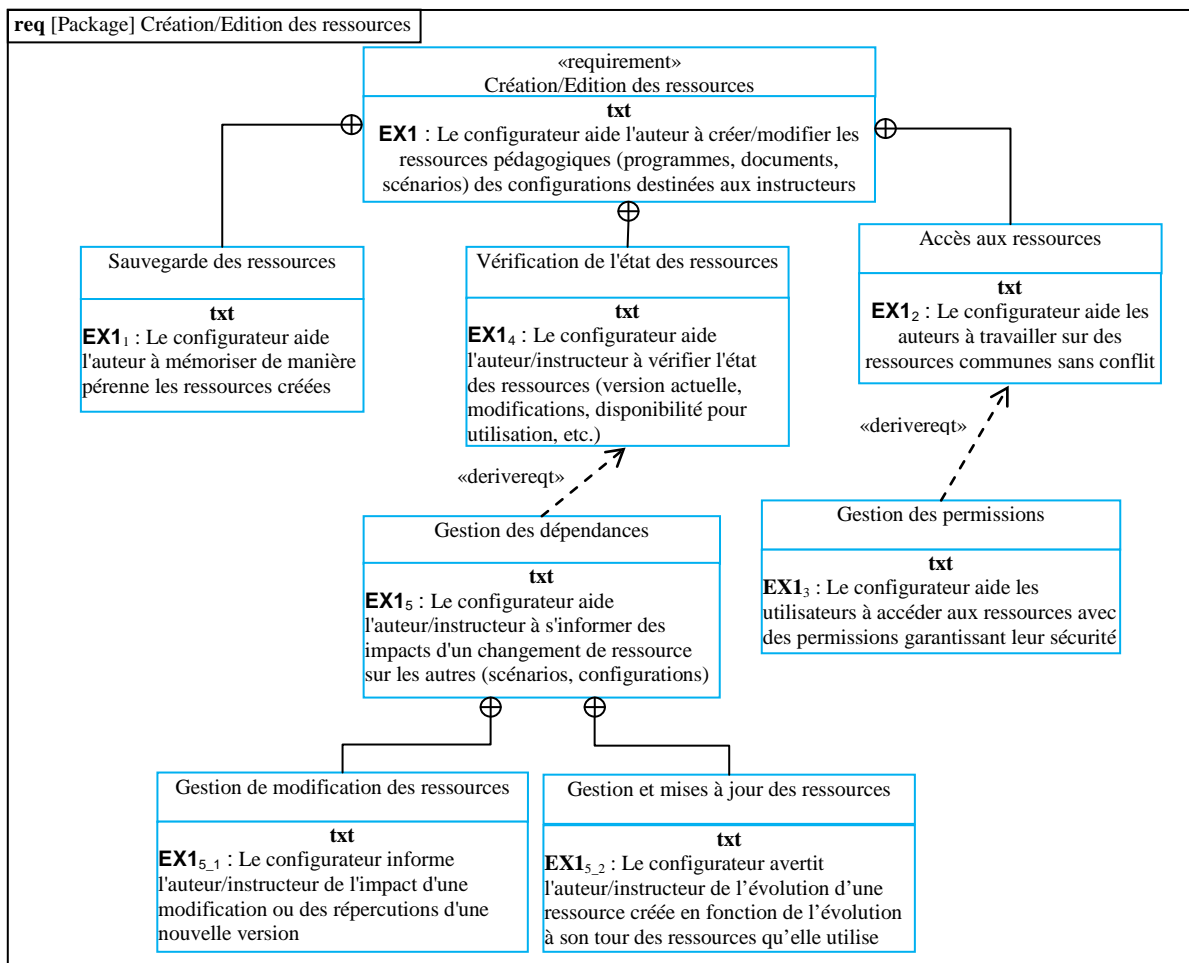


Figure 3.5 : Sous exigences de « Création/Édition des ressources ».

2. Sous exigences de EX2 « Génération de configuration » :

Cette exigence concerne l’auteur. Elle correspond à la phase préparatoire de configuration d’une session d’e-TP à partir des ressources disponibles. Cette consiste à assembler différentes ressources, et à les rendre opérationnelles dans la plate-forme d’e-TP.

Nous avons extrait dans le Tableau 3.3 des sous exigences clarifiant l’exigence principale **EX2**. La Figure 3.6 présente le diagramme correspondant.

Exigence	Réf	Source	Description	Raisonnement	Origine
Assemblage de ressources	EX2 ₁	Analyse de l’existant	Le configurateur aide l’auteur à assembler les ressources construisant une configuration	Une configuration est composée de ressources de types différents qui nécessitent d’être assemblées	BA ₄
Compilation de ressources	EX2 ₂	Analyse de l’existant	Le configurateur aide l’auteur à compiler les ressources d’une configuration	Les ressources non compilées d’une configuration ne peuvent pas être exploitées sur la plate-forme d’e-TP (ex : programme API sur SAP, scénarios pédagogiques sur LMS)	BE ₁₀

Tableau 3.3 : Liste des sous exigences de EX2 « Génération de configurations ».

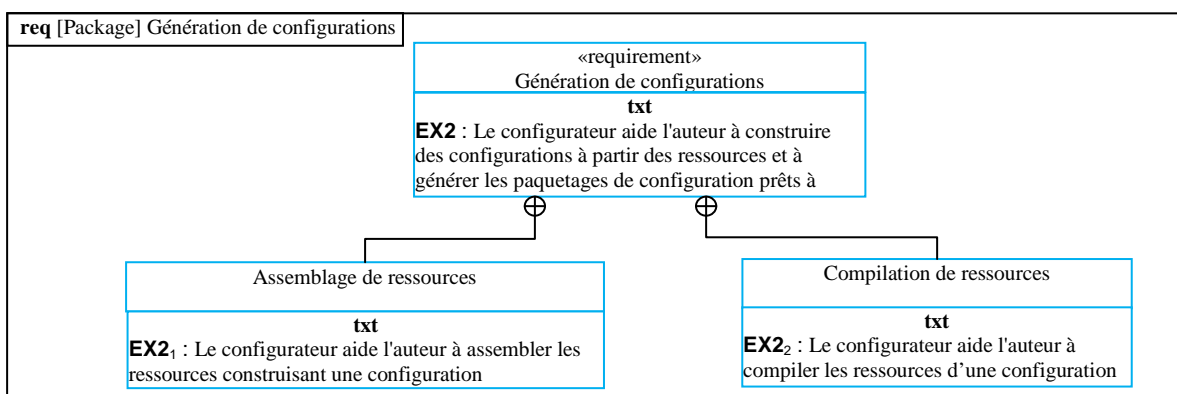


Figure 3.6 : Liste des sous exigences de EX2 « Génération de configurations ».

3. Sous exigences de EX3 « Exploitation des ressources » :

Cette exigence concerne l’auteur et l’instructeur. Elle correspond à la phase opératoire du processus de configuration d’une session d’e-TP. Il s’agit d’exploiter les ressources développées par les deux utilisateurs principaux selon différents objectifs :

- le configurateur doit permettre à l’auteur de rechercher/récupérer les ressources disponibles afin de les (ré)utiliser dans la construction de configuration selon des objectif pédagogiques données.
- il doit permettre à l’instructeur de rechercher/récupérer une configuration correspondante à des objectifs pédagogiques prédéfinis, puis à l’installer en un minimum de temps et en ne faisant pas appel au technicien.

Tout cela devra être fourni par le configurateur aux utilisateurs depuis n'importe où et sans se limiter à un système d'exploitation donné.

Le Tableau 3.4 explicite l'exigence principale **EX3**. La Figure 3.7 présente le diagramme correspondant.

Exigence	Réf	Source	Description	Raisonnement	Origine
Recherche de ressources et de configuration pour l'auteur	EX3 ₁	Auteur	Le configurateur assiste l'auteur dans sa recherche des ressources nécessaires pour configurer une session d'e-TP et dans sa recherche de configurations existantes en mettant à sa disposition des critères de sélection.	L'auteur doit trouver et choisir sans confusion les ressources appropriées pour créer une configuration, et doit pouvoir trouver des configurations qui l'intéressent	BA ₂
Recherche de configurations pour l'instructeur	EX3 ₂	Instructeur	Le configurateur assiste l'instructeur dans sa recherche de configurations existantes en mettant à sa disposition des critères de sélection.	La recherche des configurations selon des critères précis simplifie la sélection de celles qui correspondent aux besoins pédagogiques précis	BI ₁
Installation de configurations	EX3 ₃	Instructeur	Le configurateur doit assister l'instructeur dans l'installation d'une configuration d'e-TP en transférant les programmes sur les APIs et en le guidant dans la phase de configuration manuelle	L'instructeur doit préparer facilement la session d'e-TP à l'aide du configurateur, sinon il va hésiter à l'utiliser ultérieurement	BI ₂
Diminution du temps de configuration	EX3 ₄	Instructeur	L'instructeur bénéficie d'un temps réduit pour configurer une session d'e-TP	Le temps de (re)configurer une session d'e-TP doit être amélioré par rapport à la façon traditionnelle d'en faire	BI ₃
Indépendance l'instructeur/technicien	EX3 ₅	Analyse de l'existant	L'installateur minimise le degré de dépendance entre instructeur et technicien lors de la configuration d'une session d'e-TP	L'indisponibilité de technicien pour configurer une session d'e-TP peut empêcher son déroulement	BI ₄
Accès aux ressources sans contraintes	EX3 ₆	Analyse de l'existant	L'utilisateur peut accéder aux ressources du configurateur depuis n'importe quel ordinateur et sans s'inquiéter du système d'exploitation qu'il utilise	L'accès aux ressources restreint à certains ordinateurs et/ou certains systèmes d'exploitation limite l'utilisation du configurateur à certains personnels	BA ₂

Tableau 3.4 : Liste des sous exigences de EX3 « Exploitation des ressources ».

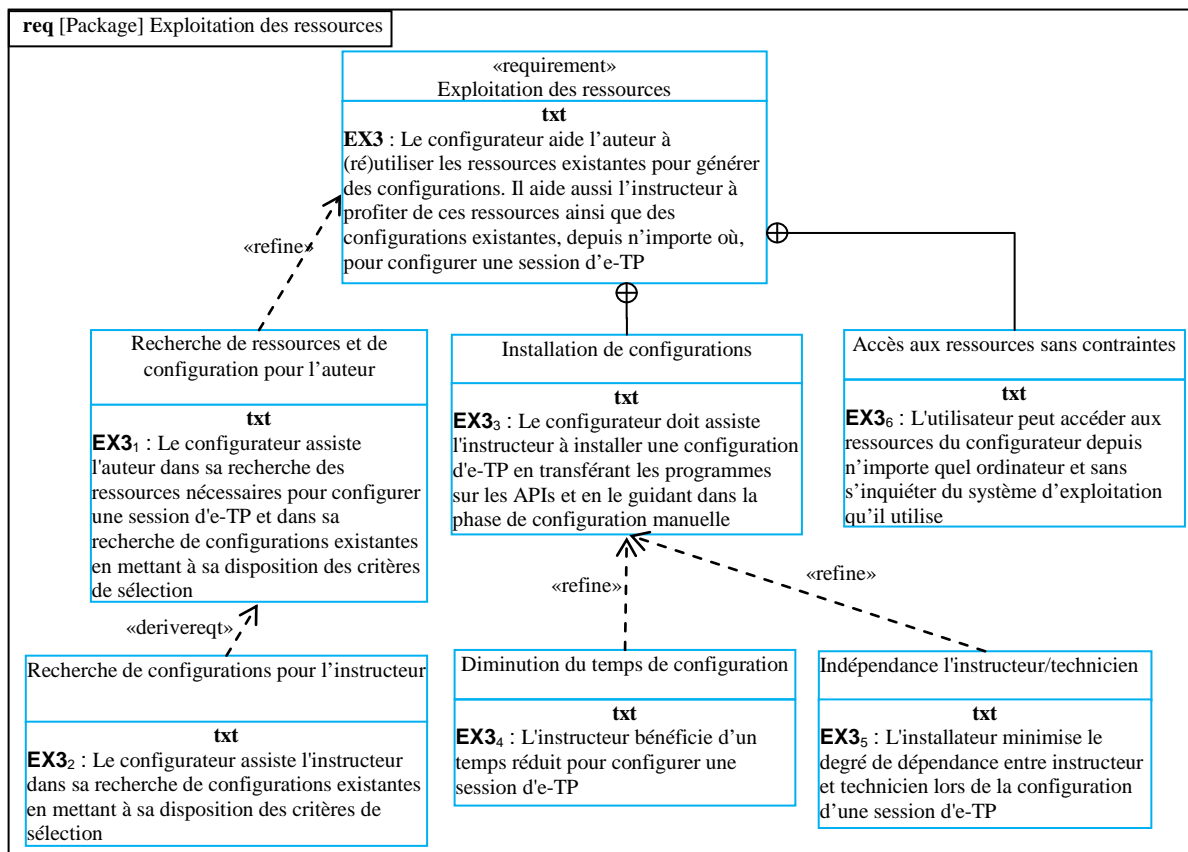


Figure 3.7 : Liste des sous exigences de EX3 « Exploitation des ressources ».

Les diagrammes d'exigences développés dans cette section décrivent la spécification, et seront utilisés dans la phase de conception du configurateur comme point de départ.

3.3.1.4 Notre engagement de réalisation

Compte tenu de la taille des exigences précédentes, précisons ici notre engagement :

1. Conception du configurateur :

Nous développons dans les sections suivantes la structure de ce système, ainsi que son comportement, en fonction des exigences du système. Le modèle ainsi obtenu satisfait **toutes** les exigences recensées indépendamment de toute technologie de réalisation (implémentation), laissant ainsi le développeur libre du choix des technologies de réalisation.

2. Réalisation d'un prototype de configurateur :

Le prototype se base sur le modèle. Les exigences, parmi celles recensées, que satisfait notre prototype sont représentées sous la forme d'un diagramme (voir Figure 3.8).

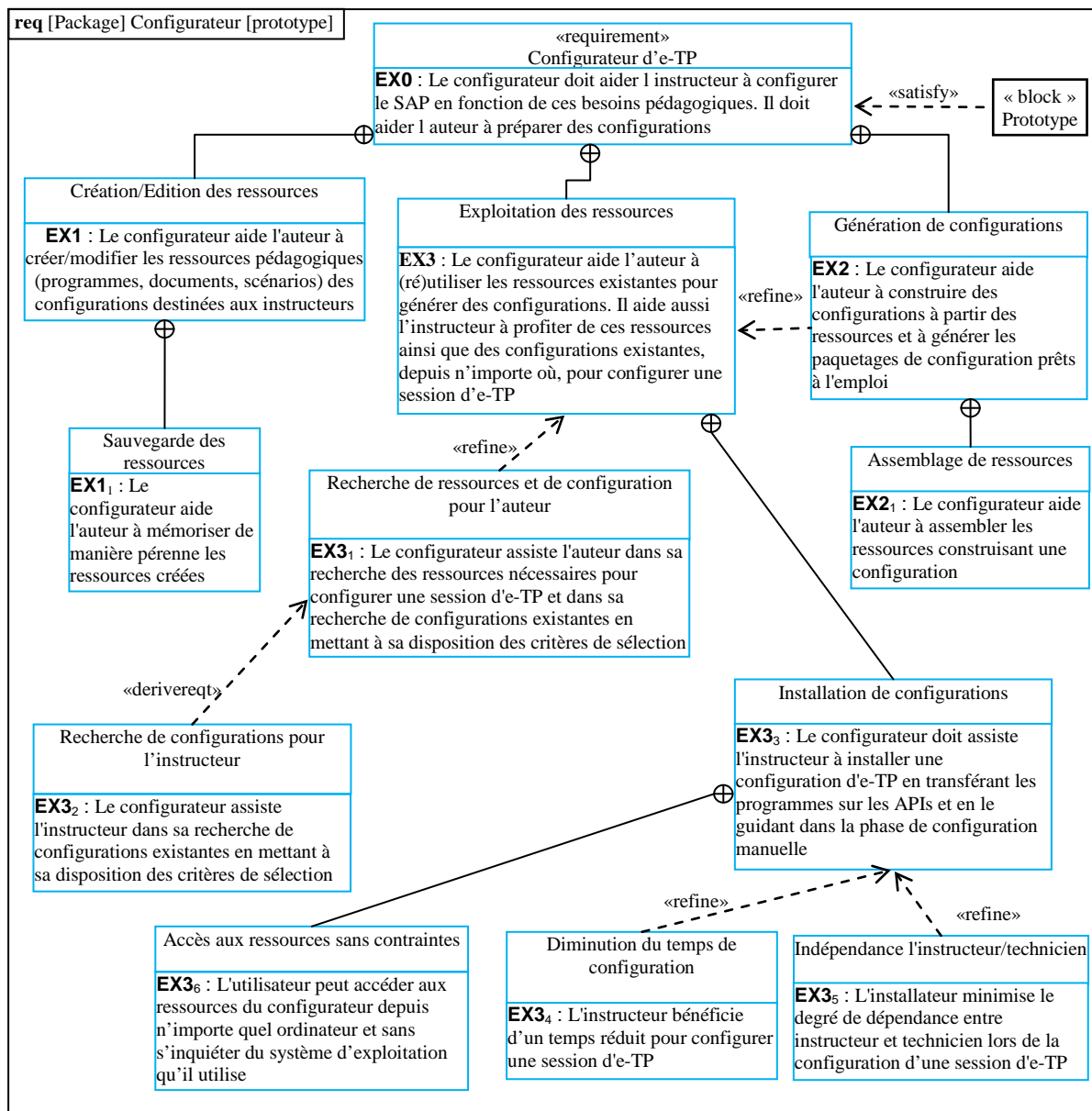


Figure 3.8 : Diagramme d'exigences du configurateur que le prototype satisfera.

3.3.1.5 Cas d'utilisation types

En fonction des exigences précédentes, on distingue deux phases d'utilisation de ce configurateur : la phase de création des ressources et des configurations (par l'auteur), et la phase de leur utilisation (par l'instructeur).

En partant de l'exigence **EX1** «Création/Édition des ressources» et en se basant sur les cas d'utilisation présentés dans la Figure 3.3, l'**auteur** peut à travers le système du configurateur **créer** : un scénario d'e-TP répondant à des objectifs pédagogiques, des documentations liées à ce scénario et les programmes à exploiter sur le SAP (si besoin avec l'aide d'un technicien) (voir Figure 3.9). L'auteur est ici l'acteur principal ; le technicien peut

intervenir dans le cas où l'auteur n'est pas spécialiste en automation. L'auteur se charge aussi dans cette phase de l'édition des ressources/configurations existantes afin d'en générer de nouvelles adaptées aux nouveaux objectifs pédagogiques.

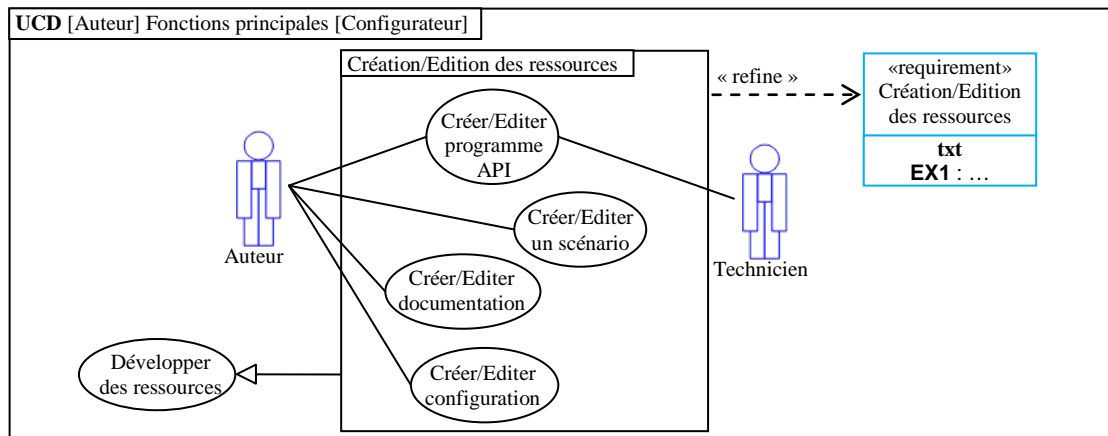


Figure 3.9 : Cas d'utilisation types - Phase de création des configurations.

En partant de l'exigence **EX3** « Exploitation des ressources », et en se basant sur les cas d'utilisation de la Figure 3.3, l'instructeur peut, lors de la phase d'utilisation des configurations (voir Figure 3.10) :

- choisir une configuration compatible avec ses objectifs pédagogiques à travers un moteur de recherche naviguant dans un catalogue de formations ;
- charger les programmes automatés correspondants sur le système automatisé ciblé ;
- charger les scénarios pédagogiques liés, sur un LMS (celui de son institution ou éventuellement un LMS spécifique) ;
- télécharger la documentation complémentaire : pour les opérations manuelles de configuration non automatisables et autres notices d'utilisation.

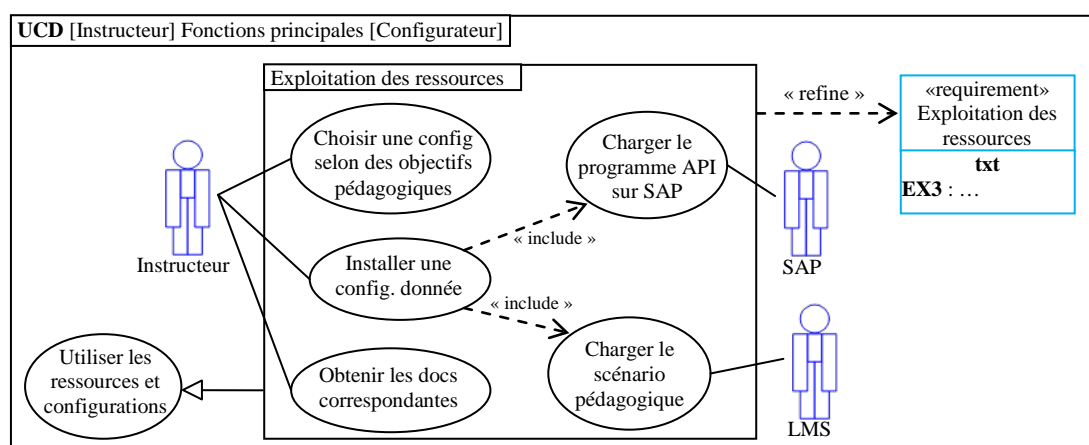


Figure 3.10 : Cas d'utilisation types - Phase d'utilisation des configurations.

Dans cette phase, le configurateur installe automatiquement les configurations d'une session d'e-TP préparées par l'auteur (téléchargement des programmes API sur les automates, téléchargement du scénario et de documentation pédagogique sur le LMS). Au cas où l'installation d'une configuration n'est pas automatisable, l'instructeur se charge de l'installer, en se faisant aider par un technicien si nécessaire, et en chargeant les programmes API sur la maquette pédagogique et le scénario d'apprentissage sur le LMS.

Cette logique implique donc l'intervention de plusieurs acteurs afin de préparer et dérouler une session d'e-TP (voir Tableau 3.5).

Acteur		Description de leur rôle
Humain	Auteur	Prépare, en amont, des configurations prêtes à l'emploi ; il est spécialiste du domaine d'application (ordonnancement, robotique, vision, MES, ...) mais pas nécessairement un spécialiste en automation.
	Instructeur	Choisit et implante une configuration selon ses objectifs pédagogiques.
	Technicien	Chargé de développer et maintenir les programmes API et d'assister les auteurs lors de la conception des configurations
Non-Humain	SAP	Le Système Automatisé de Production (SAP) et plus précisément son ou ses API et Interfaces Homme-Machine industrielles (panels graphiques tactiles par exemple) sont reprogrammés à chaque reconfiguration.
	LMS	Accueille les scénarios pédagogiques sélectionnés par l'instructeur et envoyés par le configurateur

Tableau 3.5 : Les acteurs principaux du système de configuration.

3.3.1.6 Le configurateur dans son environnement

Après présentation des besoins auxquels répond notre configurateur et avant formalisation du modèle du système. Il est important d'identifier dans l'environnement du système, ce qui peut interagir directement ou indirectement avec lui. Pour cela, nous utilisons un diagramme de type BDD (voir Figure 3.11).

L'environnement du configurateur est composé de deux éléments :

1. les utilisateurs: les auteurs et instructeurs sont les utilisateurs auxquels le configurateur est destiné. Le technicien est un utilisateur secondaire qui intervient lors de l'utilisation du configurateur par les deux autres ;
2. les éléments externes qui interagissent avec le configurateur : le SAP, le LMS qui gère l'exécution de scénarios pédagogiques transférés par le configurateur.

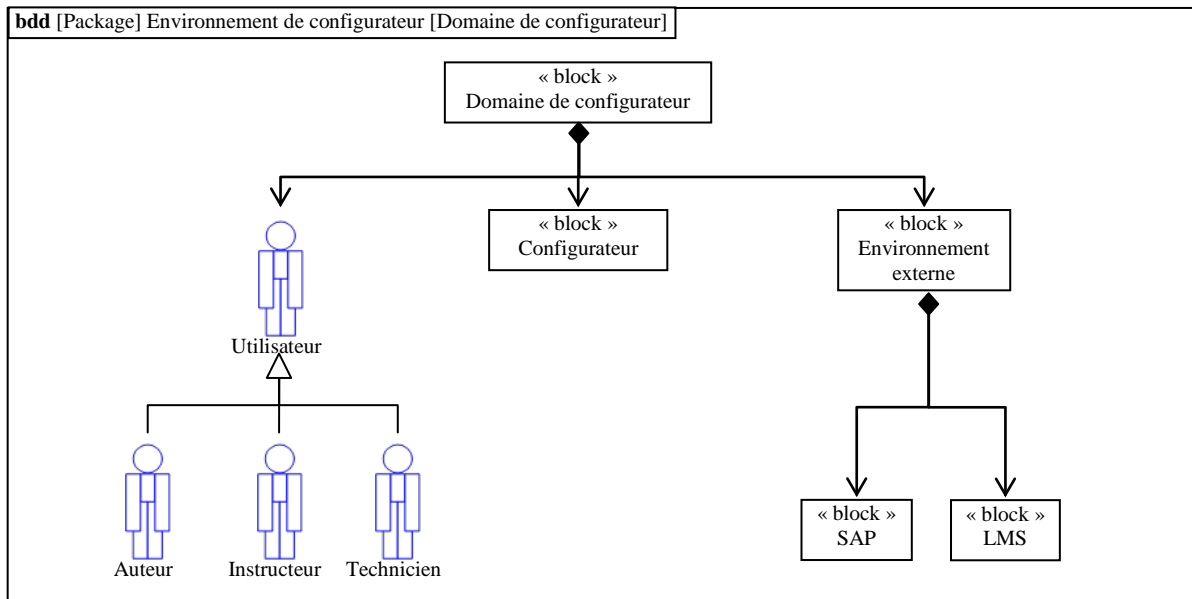


Figure 3.11 : Le domaine du configurateur, ses utilisateurs et son environnement.

Cette identification de domaine du configurateur nous aidera, dans la phase de modélisation suivante, à :

- affiner les rôles des différents acteurs ;
- identifier leurs interactions avec le système ;
- fournir une ligne conductrice pour distinguer les différentes parties du système à modéliser et leurs relations mutuelles.

3.3.1.7 Les données du configurateur en interface avec l'environnement

Partons d'une vision globale du configurateur comme une boîte noire (vision systémique).

Les données d'entrée **brutes** du configurateur sont :

- un ensemble de critères (de recherche) : objectifs pédagogiques⁵⁵, nom ou type de SAP, domaine d'expérimentation, etc. (voir Figure 3.12). Ces critères de recherche sont fournis par l'utilisateur (auteur et instructeur) sous forme de mot clés, catégories, sujets, SAP de certains types, etc. ces critères sont extensibles selon les besoins de l'utilisateur (par exemple : date de création, version, auteur(s), etc.) ;

⁵⁵ Objectifs pédagogiques tels que : « savoir extraire d'un SAP les données nécessaires au calcul de son TRS (Taux de Rendement Synthétique) », « Savoir mettre en place une traçabilité à partir d'étiquettes RFID », « Maîtriser le logiciel UNITY de Schneider Electric », etc.

- de « *ressources pédagogiques* » : documents bruts, tels que des sous programmes API, schémas, textes explicatifs, notices d'utilisation, scénarios pédagogiques, etc. (voir Figure 3.13).

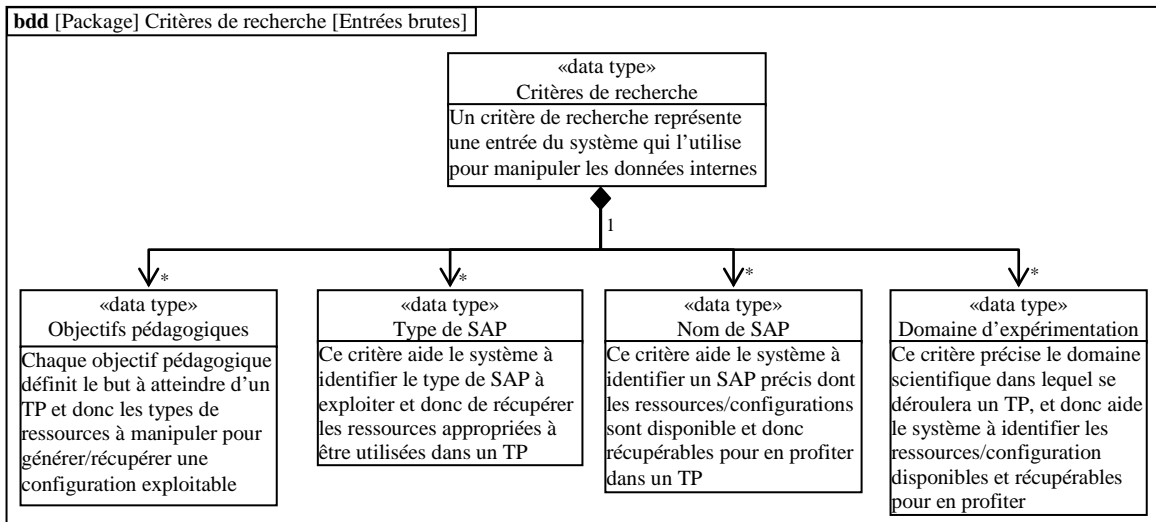


Figure 3.12 : Décomposition des critères de recherche.

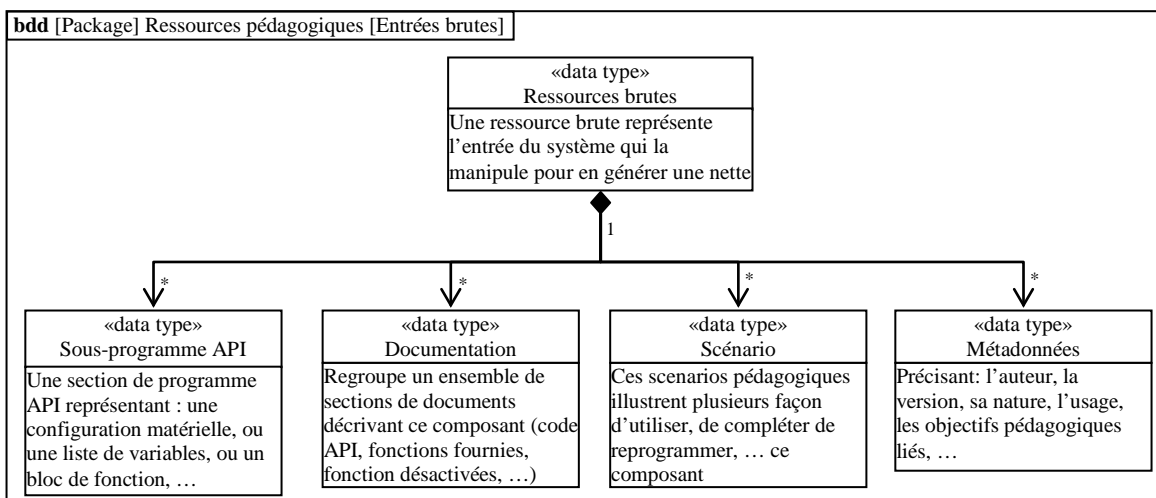


Figure 3.13 : Décomposition des types de ressources brutes.

Les données de sortie sont des configurations de sessions d'e-TP. Partant de la sous exigence **EX2₂** « Compilation de ressources », les ressources à exploiter sont disponibles sous forme compilée afin de les rendre directement exploitables dans un e-TP. Une configuration est un ensemble de ressources brutes manipulées et compilées pour générer d'autres ressources sous formes de *fichiers compilés* prêts à être exploités (voir Figure 3.14).

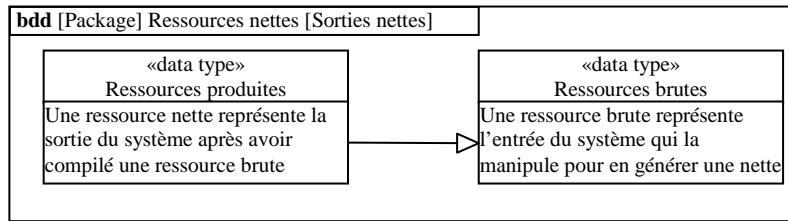


Figure 3.14 : Ressources produites.

La sortie du système, la configuration, est composée de documentations (exemples, notices, instruction), programmes API à télécharger manuellement sur le SAP, scénarios pédagogiques, qui sont tous compilés (voir Figure 3.15).

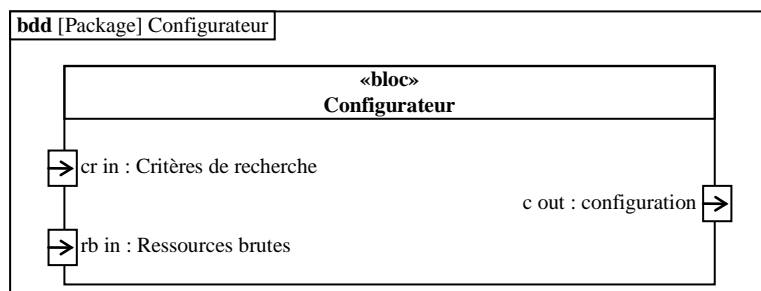


Figure 3.15 : Les entrées/sorties du configurateur.

3.3.1.8 Objectifs qualité du système

Le système proposé vise à :

- **OBJ1** : diminuer le temps de configurer un SAP **d’au moins 50%** ;
- **OBJ2** : améliorer la réutilisation des ressources d’e-TP de manière conduisant à diminuer le temps de sur la création de deux configurations de TP proches, **d’au moins 50%** ;
- **OBJ3** : améliorer l’autonomie de l’instructeur d’une session d’e-TP en réduisant sa dépendance à un technicien, par diminuer le nombre d’appels à un **technicien d’au moins 50%**.

3.3.1.9 Bilan sur le modèle CIM

Nous avons identifié dans les sections précédentes l’objectif général du système à modéliser, les besoins principaux auxquels il doit répondre et les cas d’utilisation types. Le développement des schémas précédents correspond à la phase de conception du modèle CIM de la démarche MDA, ce qui n’a impliqué aucune spécification d’une technologie cible ou moyen de mise en œuvre. Nous pouvons donc, à l’aide de ces diagrammes, avancer vers

l'étape suivante de la démarche MDA pour développer un modèle plus détaillé du système toujours indépendant de toute technologie.

3.3.2 Définition structurelle du système : modèle PIM

Nous abordons la structure du configurateur : quels composants faut-il faire apparaître ? Et comment ses composants communiquent-ils ? Cette phase de modélisation correspond au modèle PIM de la démarche MDA. Nous partons donc des besoins décrits précédemment pour proposer les briques constitutives de notre système.

3.3.2.1 Types de données internes

Partons de l'exigence « **EX3** » qui favorise la réutilisation des ressources pour différentes configurations. Une configuration contient des ressources « nettes » prêtes à être exploitées directement sur une plate-forme donnée (par exemple : un programme API au format propriétaire d'un automate, un scénario pédagogique au format propriétaire d'un LMS). Il est donc impossible de les réutiliser ailleurs. Notre vision consiste donc à séparer la configuration de travail, créée par l'auteur et basée sur des ressources brutes, de celle finale pour l'exploitation, utilisée par l'instructeur et basée sur des ressources nettes. Nous parlons ici d'une structure de configuration à deux états : **configuration générique** dont les ressources sont réutilisables (brutes), et **configuration spécifique** dont les ressources sont exploitables (nettes).

Cette structure « *Composant* » a pour rôle d'assembler plusieurs types de données brutes (cf. Figure 3.13). Elle se compose de (voir Figure 3.16) :

- sous-programmes API (une configuration matérielle, une liste de variables globales, un composant réutilisable (appelé DFB dans Unity), une section de code API, ...) stockés au format OpenPLC pour en favoriser la réutilisation. Étant donné que l'exigence **EX3** implique d'exploiter les ressources en favorisant leur réutilisation, grâce à cette décomposition du programme API en sous programmes l'auteur profite d'une ou plusieurs parties du programme API selon son objectif pédagogique ;
- un ensemble de scénarios pédagogiques élémentaires en liaison avec ces sous-programmes ;
- une documentation liée à l'explication et l'utilisation de ce composant ;

- des métadonnées indiquant l'auteur, la version, sa nature, des mots clefs, l'usage, ... des objectifs pédagogiques liés, ...

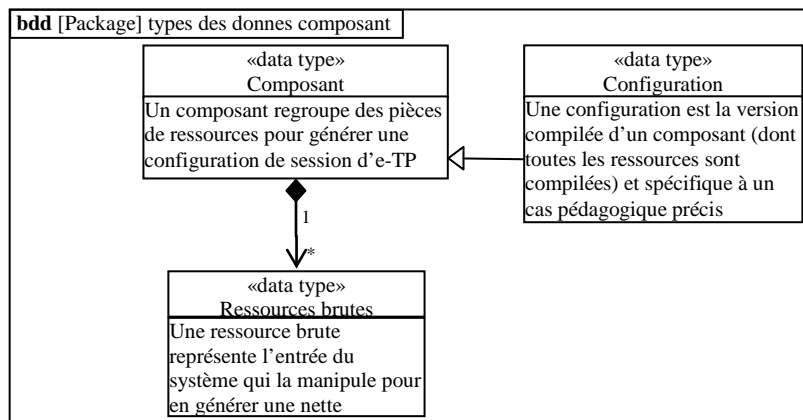


Figure 3.16 : Structure du « Composant ».

Un *composant* est un paquetage de ressources pédagogiques contenant des fichiers assemblés selon des objectifs pédagogiques, qui sont toutes éditables. Il représente une configuration générique (dont toutes les ressources sont éditables) qui peut être modifiée et/ou compilée (validée) afin de générer une configuration finale exploitable dans une session d'e-TP. Ainsi un composant doit être (ré)éritable et n'est pas associé à une installation (SAP, LMS, ...) spécifique (c'est en cela qu'il est considéré "générique"). Une *configuration* est, par contre, spécifique d'une installation et dans un état exploitable sur cette installation mais ne peut plus être éditée. Par exemple, pour une même ressource, un programme API dans un composant sera enregistré au format OpenPLC tandis que dans une configuration il sera dans un format propriétaire prêt à être téléchargé sur un API.

Des métadonnées sont associées à chaque ressource (voir Tableau 3.6) afin de les identifier, les catégoriser et lier des ressources différentes entre elles (par exemple, une documentation expliquant un sous-programme).

Champ	Description
Catégories	Les catégories auxquelles une ressource est associée
Objectifs pédagogiques	Les objectifs pédagogiques d'un TP auxquelles une ressource est associée
Type de SAP	Le type de SAP avec lequel une ressource est utilisée
Nom de SAP	Le nom du SAP avec lequel une ressource est utilisée

Tableau 3.6 : Les champs de métadonnées.

3.3.2.2 Généricité de configuration et réutilisation des ressources

Partant de l'exigence **EX3** qui implique d'exploiter les ressources en favorisant leur réutilisation, le configurateur la satisfait par le biais des composants (voir Figure 3.17).

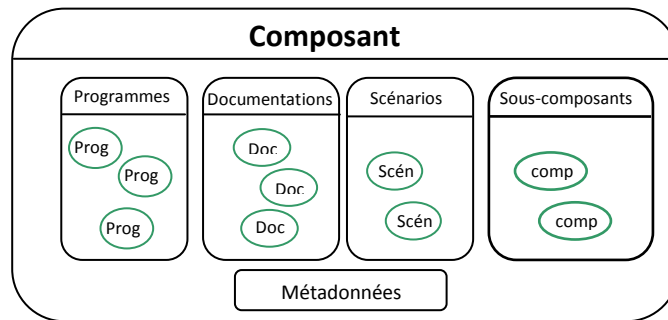


Figure 3.17 : Structure de composant (configuration générique).

Effectivement, la séparation entre la phase générique d'une configuration et la phase spécifique donne de la flexibilité à l'auteur de composant (configuration générique), en le structurant à partir des ressources pédagogiques, et en générant plusieurs versions, chacune adaptée à un cas d'utilisation ou servant un objectif pédagogique différent. La génération de différentes versions d'un composant d'automatisme se conforme aux techniques de POO (encapsulation et héritage). En effet, l'assemblage de plusieurs ressources qui servent un objectif défini dans une seule entité, le composant, ressemble au concept d'encapsulation des données adopté dans la POO. Du même, la modification d'un composant en ajoutant, supprimant ou modifiant ses ressources pour en générer une nouvelle version ressemble au concept d'héritage adopté aussi dans la POO.

Étant donné que la spécification du configurateur implique de gérer les dépendances entre les ressources (cf. Tableau 3.2, **EX1₂** et **EX1₅**), la génération d'une configuration spécifique pourra être problématique, vu que le composant d'origine est éditable à tout moment. La Figure 3.18 illustre le mécanisme adopté pour gérer les liens entre les différents éléments (ressources éditables, ressources compilées, composants, configurations) lors de génération de plusieurs versions d'un composant, puis la génération d'une configuration correspondante à chaque version. Effectivement, tandis que la compilation de ressources génériques d'un composant produit des ressources spécifiques pour une configuration, la modification d'une ressource générique (au sein du composant contenant), puis sa compilation va produire une autre version spécifique de cette ressource sans avoir d'effet sur la version déjà existante. De cette manière l'auteur ne se soucie plus d'associer une ressource générique à une version d'un composant, ni d'associer une ressource compilée à une version de configuration. C'est le configurateur qui se charge de gérer les liaisons entre les différents éléments, sans aucune intervention de l'auteur. Cette flexibilité d'édition libre des ressources et des configurations, fournie par le configurateur, améliore la réutilisabilité des ressources en omettant le souci de

produire des chevauchements de liens entre les configurations (génériques et spécifiques) et les ressources concernées.

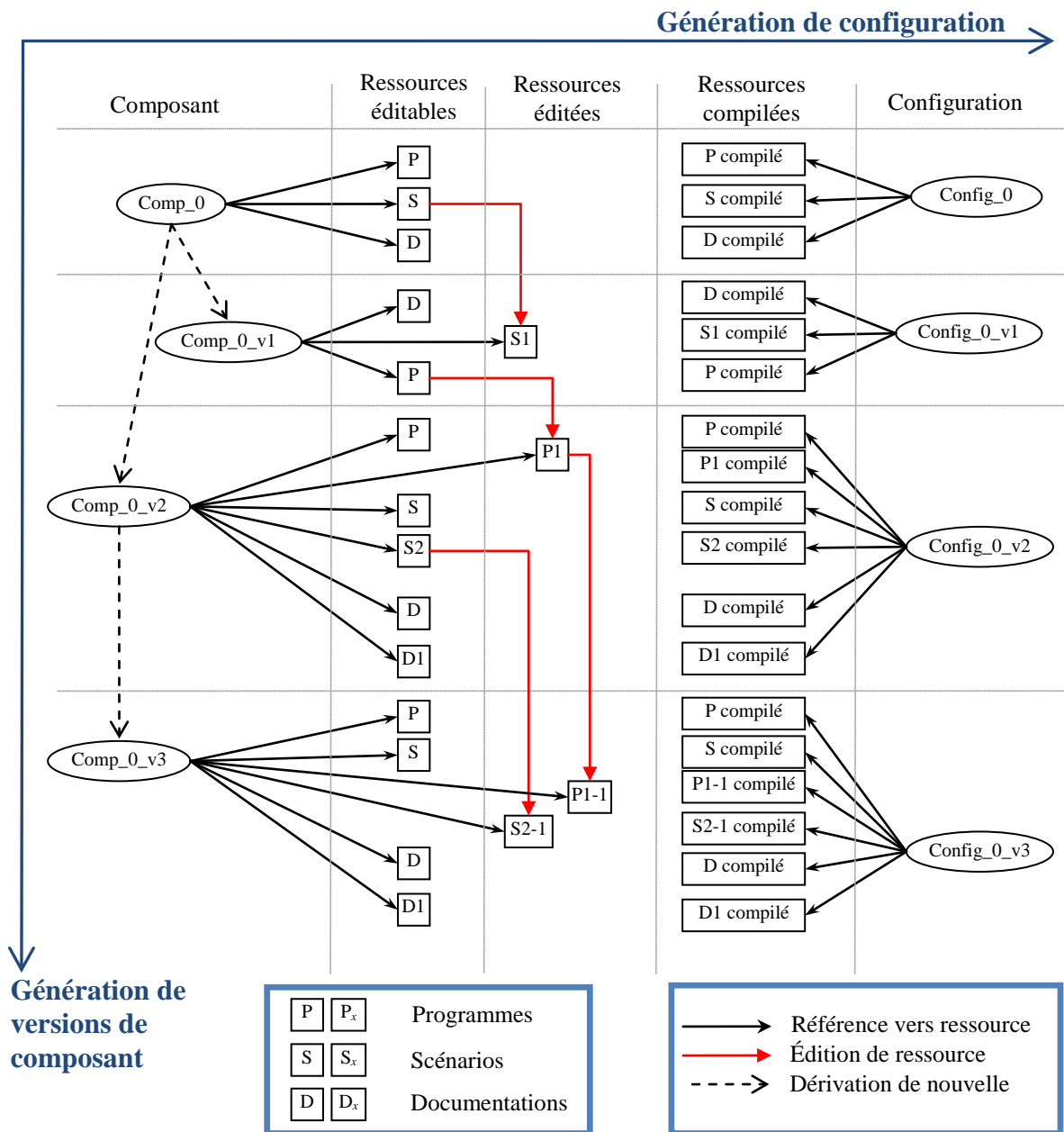


Figure 3.18 : Génération de versions d'un composant pour en générer plusieurs configurations.

3.3.2.3 Flux de données internes du configurateur

Conformément aux trois exigences principales du système (**EX1**, **EX2** et **EX3**), les utilisateurs ont trois actions types : création des ressources, génération de configurations et exploitation de ressources. Cela implique une interaction entre le système et son environnement. Le configurateur est utilisé lors des phases de création et d'exploitation de configuration. Les diagrammes d'activités de la Figure 3.19 illustrent les activités principales

associées au configurateur pour transformer les entrées (objectifs pédagogiques, ressources pédagogiques, critères de recherche, métadonnées) en sorties (composant, configuration) dans chacune des deux phases. Ces activités sont :

- création de ressources ;
- création de composant ;
- génération de configuration ;
- installation de configuration.

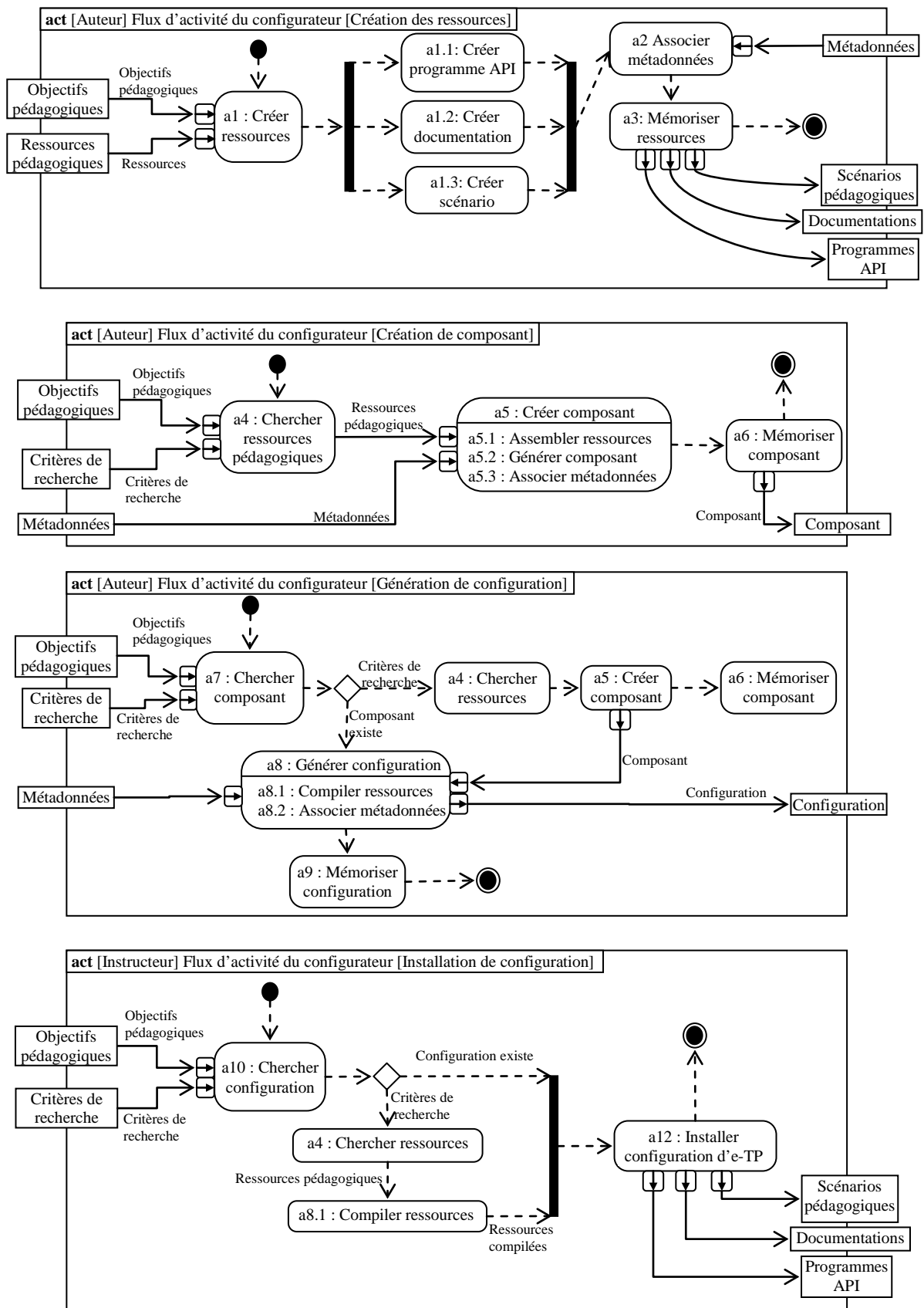


Figure 3.19 : Flux d'activités au sein du configurateur.

La Figure 3.20 présente une décomposition fonctionnelle selon un diagramme *bdd*. Cette décomposition n'indique pas pour l'instant quelle partie du système se charge de quelle activité.

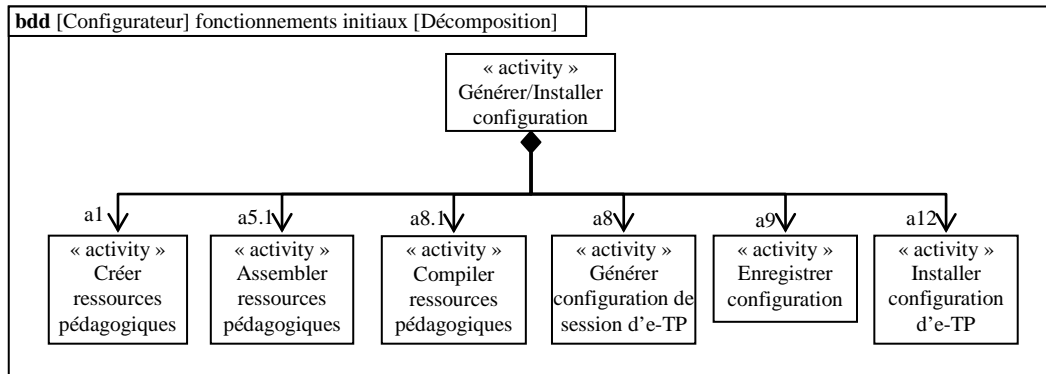


Figure 3.20 : Première décomposition fonctionnelle du configurateur.

3.3.2.4 Architecture du configurateur

Partant de l'exigence **EX1**, le système assure la création/édition de ressources pédagogiques à partir desquelles les auteurs gèrent la création des configurations. Ainsi, le « Configurateur » dispose d'un bloc « Bibliothèque des ressources pédagogiques » qui fournit à l'auteur des éléments pour créer, éditer et assembler des ressources pédagogiques lors de la phase de préparation, en fonction de ses objectifs (pédagogiques). Cette bibliothèque mémorise les ressources développées.

Pour répondre à l'exigence **EX2**, le système assure la génération de configurations à partir des ressources pédagogiques à travers la fonction « Générateur de configurations ».

Pour satisfaire les exigences **EX3** et **EX3₃**, deux fonctions (blocs) sont proposés :

- un bloc « Bibliothèque de configuration » qui stocke les configurations générées dans la phase de préparation et les met à la disposition de l'instructeur ;
- un bloc « Installateur de configuration » qui supervise l'installation d'une configuration pour l'instructeur.

La Figure 3.21 la structure du bloc « Configurateur ».

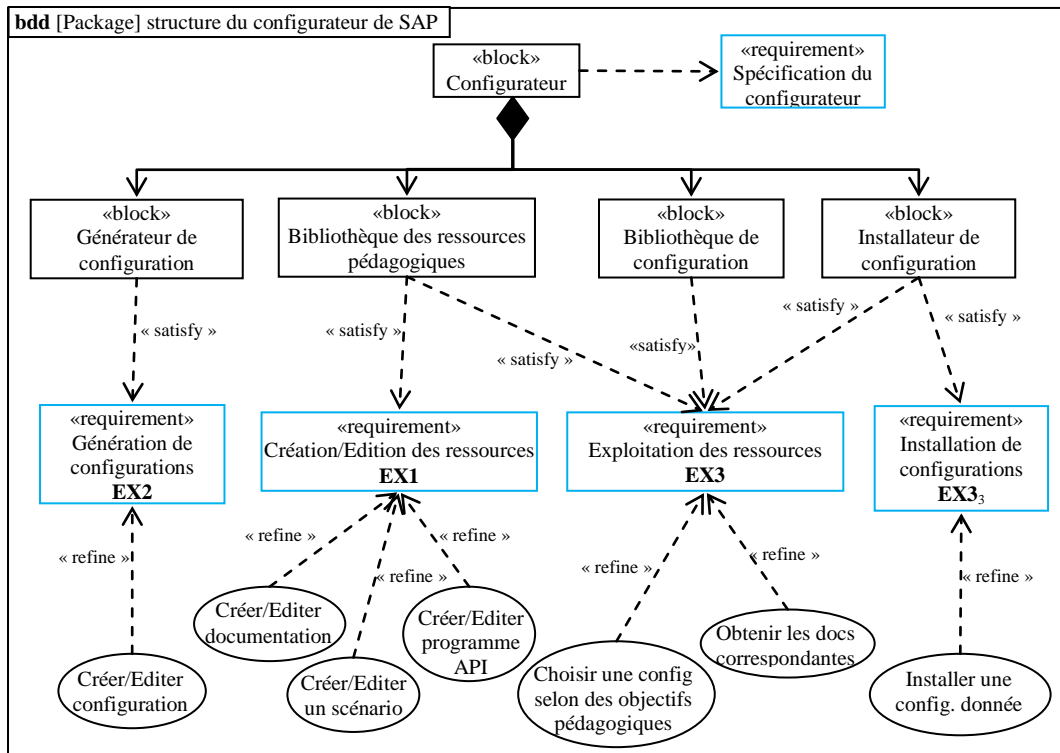


Figure 3.21 : Composants de l'environnement d'e-TP.

Selon cette structure, nous pouvons allouer les activités précédentes aux partitions du configurateur (voir Figure 3.22).

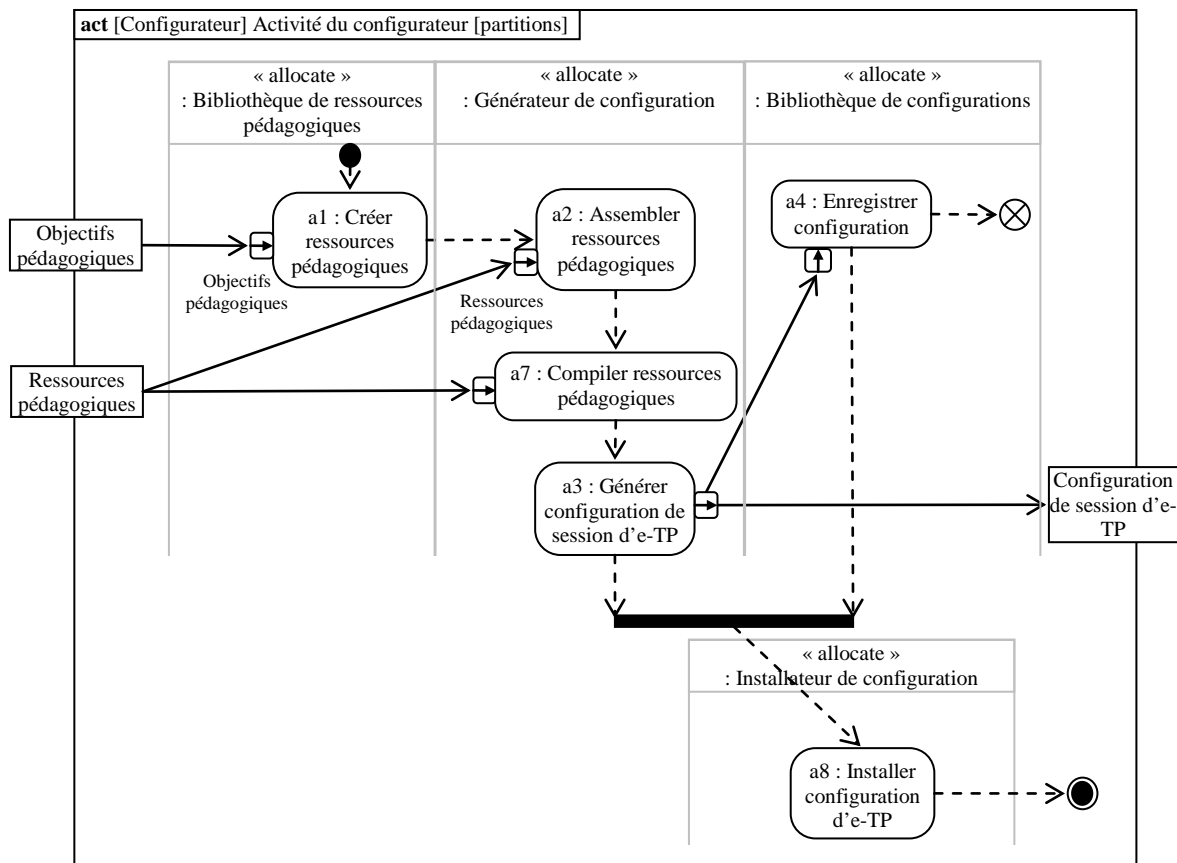


Figure 3.22 : Allocation des activités aux partitions de structure du configurateur.

Dans les sections suivantes, nous nous attachons à décrire la structure et les fonctions de chaque composant constituant le configurateur : « Bibliothèque de ressources pédagogiques », « Bibliothèque de configurations », « Génération de configuration » et « Installateur de configuration ».

3.3.2.5 Structure de la bibliothèque de ressources pédagogiques

Le bloc « Bibliothèque des ressources pédagogiques » représente un répertoire de ressources utilisables pour générer, par assemblage, des configurations de session d'e-TP.

Partant de l'exigence générale **EX1**, et en particulier de la sous-exigence **EX1₁**, cette bibliothèque dispose d'éléments facilitant la sauvegarde des ressources (scénarios pédagogiques, programmes API à télécharger sur des automates, documentations). Bien que les ressources à mémoriser dans cette bibliothèque soient interconnectées, leur réutilisation pourra être séparée. Par exemple : pour une session de TP quelconque un auteur crée des ressources (documentations, programmes API et scénarios pédagogiques) qui sont liées entre elles selon une catégorie, sujet, système, etc. Un autre auteur pourra réutiliser indépendamment les ressources qui l'intéressent pour son e-TP.

Pour une meilleure (re)exploitation des ressources (exigence **EX3**), ce bloc se subdivise en trois sous-blocs consacrés chacun à un type des ressources (programmes API, documentations et scénarios pédagogiques) (voir Figure 3.23). Ce sont les blocs :

- « **bibliothèque de scénarios** » : elle stocke les scénarios pédagogiques des sessions d'e-TP créés par l'auteur. Pour satisfaire l'exigence **EX3₁** « Recherche de ressources et de configuration pour l'auteur », les scénarios dans cette bibliothèque sont indexés à l'aide des métadonnées ;
- « **bibliothèque de programmes API** » : elle regroupe les programmes API créés par l'auteur (à l'aide d'un technicien si nécessaire). Pour satisfaire l'exigence **EX3₁**, les programmes dans cette bibliothèque sont indexés à l'aide des métadonnées;
- « **bibliothèque de documentation** » : elle regroupe des documents créés par l'auteur. Elle est utilisée par l'instructeur ou l'apprenant (documentation technique sur le fonctionnement d'un composant automate, notice d'installation d'une

fonctionnalité sur un SAP, etc.). Pour satisfaire l'exigence EX3₁, les documentations dans cette bibliothèque sont également indexées à l'aide des métadonnées.

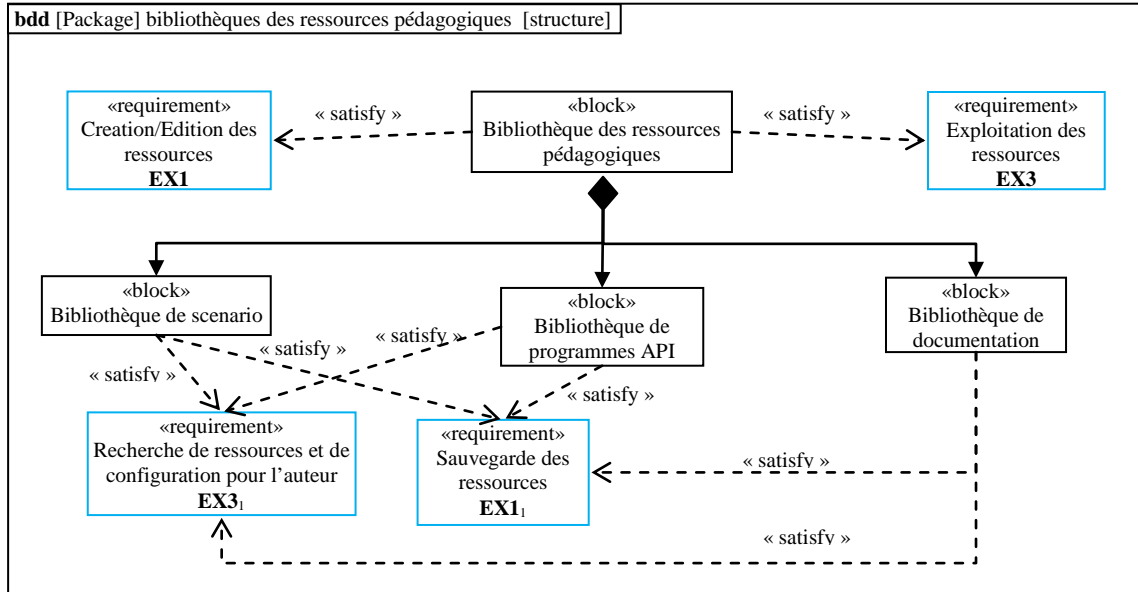


Figure 3.23 : Structure de la bibliothèque des ressources pédagogiques.

3.3.2.5.1 Entrées/Sorties et interconnexion des éléments de la bibliothèque

Les communications entre blocs sont représentées par des flux de données à travers des ports. Nous identifions, dans un *bdd*, les ports d'entrées et sorties de chaque sous bloc (voir Figure 3.24) en définissant le type de donnée manipulée à chaque port (voir Tableau 3.7).

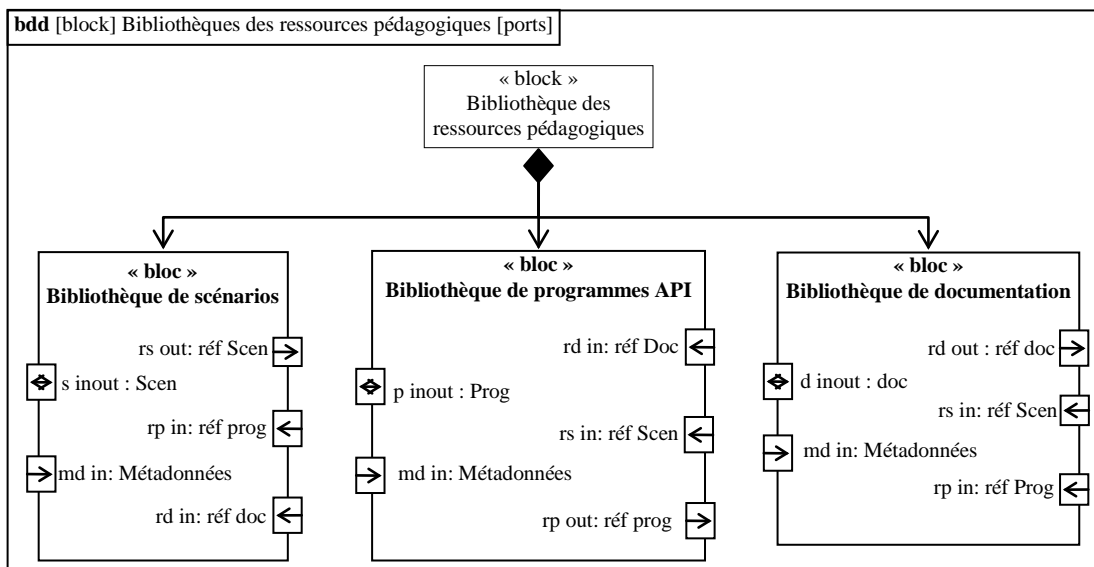


Figure 3.24 : Entrées et sorties des sous-blocs de la bibliothèque des ressources pédagogiques.

Bibliothèque de scénario	Bibliothèque de programmes API	Bibliothèque de documentation
- port <i>E/S</i> « <i>s</i> » pour recevoir/fournir un scénario	- port <i>E/S</i> « <i>p</i> » pour recevoir/fournir un programme API	- port <i>E/S</i> « <i>d</i> » pour recevoir/fournir une documentation
- port <i>E/S</i> « <i>md</i> » pour recevoir/fournir des métadonnées d'un scénario	- port <i>E/S</i> « <i>md</i> » pour recevoir/fournir des métadonnées d'un programme API	- port <i>E/S</i> « <i>md</i> » pour recevoir/fournir des métadonnées d'une documentation
- port <i>S</i> « <i>rs</i> » pour fournir la référence d'un scénario aux autres blocs	- port <i>S</i> « <i>rp</i> » pour fournir la référence d'un programme API aux autres blocs	- port <i>S</i> « <i>rd</i> » pour fournir la référence d'une documentation aux autres blocs
- port <i>E</i> « <i>rp</i> » pour recevoir la référence d'un programme API	- port <i>E</i> « <i>rs</i> » pour recevoir la référence d'un scénario	- port <i>E</i> « <i>rp</i> » pour recevoir la référence d'un programme API
- port <i>E</i> « <i>rd</i> » pour recevoir la référence d'une documentation	- port <i>E</i> « <i>rd</i> » pour recevoir la référence d'une documentation	- port <i>E</i> « <i>rs</i> » pour recevoir la référence d'un scénario

Tableau 3.7 : Ports associés aux sous blocs de la « Bibliothèque des ressources pédagogiques ».

Ces ports gèrent les différents flux de données utilisés pour réaliser les tâches adressées à la bibliothèque des ressources pédagogiques.

La Figure 3.25 présente l'interconnexion de ces fonctions sous la forme d'un diagramme *ibd*. Par exemple : l'insertion d'un élément de scénario se fait à travers le port « *s* » du bloc « bibliothèque de scénarios » et le port « *md* » reçoit les métadonnées indexant cet élément. Par contre, le port « *rs* » fournit la référence de cet élément aux autres bibliothèques pour l'associer à des éléments de programmes et documentations qui lui sont liés, et les ports « *rd* » et « *rp* » reçoivent les références des éléments de programmes et documentation liés à ce scénario.

Ce schéma de communication présenté pour le sous bloc « bibliothèque de scénarios » est valable pour les deux autres sous blocs (afin de ne pas surcharger cette figure, nous ne les avons pas représentés).

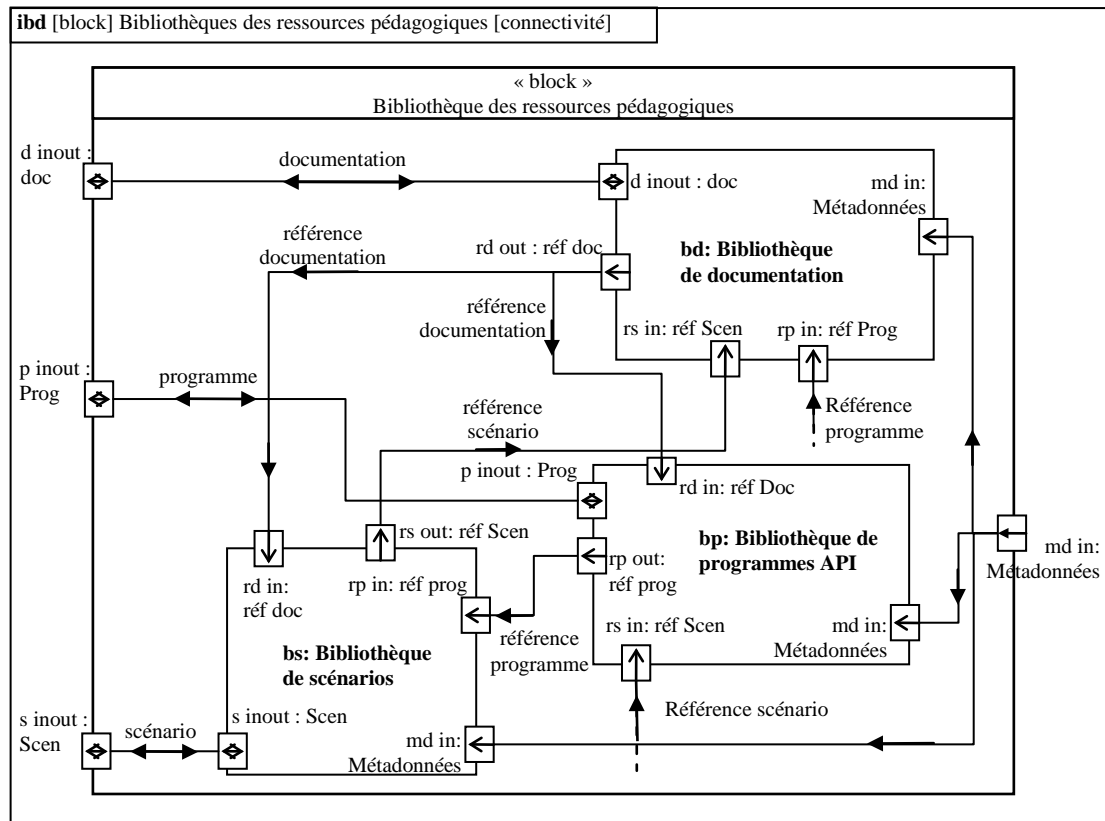


Figure 3.25 : Connexions entre les blocs du « Bibliothèque des ressources pédagogiques ».

3.3.2.6 Structure du générateur de configuration

Pour satisfaire l'exigence générale **EX2**, le configurateur est doté des fonctionnalités de création et d'édition des configurations à partir des ressources pédagogiques. Le bloc « Générateur de configuration », intégré à la structure du configurateur, y est consacré. Ce bloc représente le noyau fonctionnel du configurateur. En considérant les sous exigences **EX2₁** et **EX2₂**, ce bloc dispose des éléments structurels qui se chargent d'assembler et compiler des ressources de différents types afin de créer une configuration de session d'e-TP (voir Figure 3.26).

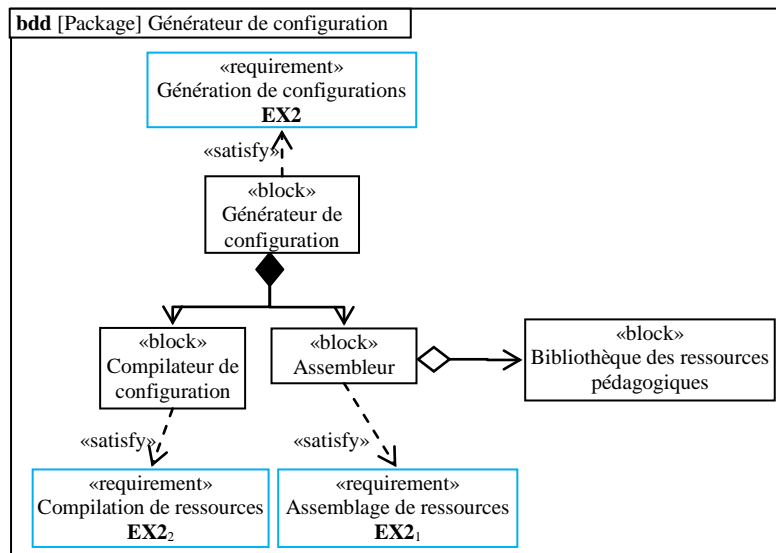


Figure 3.26 : Structure du générateur de configuration.

Étant donné que la génération de configuration se base sur l’assemblage et la compilation des ressources sauvegardées dans la « bibliothèque des ressources pédagogiques », ce bloc se compose de deux sous-blocs :

1. un bloc « **Assembleur** » : partant de la sous exigence **EX2₁**, ce bloc fournit à l’utilisateur les fonctionnalités nécessaires pour assembler plusieurs ressources par agrégation de leurs instances. Cet assembleur participe donc à la première étape de génération de configuration ;
2. un bloc « **Compilateur de configuration** » : partant de la sous exigence **EX2₂**, la tâche de ce bloc se base sur celle de l’assembleur et se charge de générer un paquetage de configuration.

Un diagramme *ibd* montre comment les fonctionnalités de génération sont interconnectées et un *bdd* identifie les ports d’entrées et de sorties gérant les flux de données de chaque sous bloc en définissant le type de donnée manipulé par chaque port (voir Figure 3.27, Tableau 3.8).

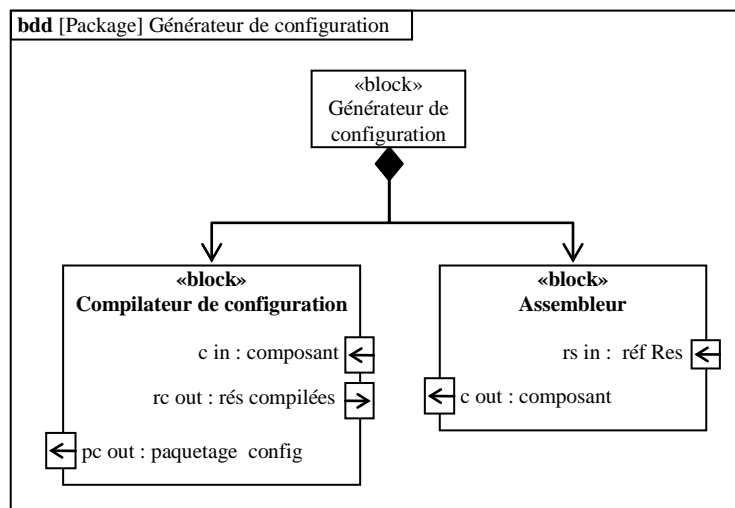


Figure 3.27 : Entrées et sorties des sous-blocs du générateur de configuration.

Assembleur	Compilateur de configuration
<ul style="list-style-type: none"> - port <i>E</i> « <i>rs</i> » pour recevoir les références des ressources à assembler - port <i>S</i> « <i>comp</i> » fournit un paquetage de ressources (depuis le bloc « Bibliothèque des ressources pédagogiques ») assemblées sous un composant 	<ul style="list-style-type: none"> - port <i>E</i> « <i>comp</i> » reçoit le composant contenant toutes les ressources à compiler - port <i>S</i> « <i>rc</i> » fournissant des ressources compilées - port de <i>S</i> « <i>conf</i> » qui fournit un paquetage de ressources compilées et assemblées sous une configuration

Tableau 3.8 : Ports associés aux sous blocs du « Générateur de configuration ».

La Figure 3.28 décrit l'interconnexion entre les sous-blocs du « Générateur de configuration ». Le port d'entrée « *rs* » reçoit les références des ressources à assembler depuis le bloc « Bibliothèque des ressources pédagogiques », en les fournissant au bloc « Assembleur » qui les assemble sous un composant. Puis, l'Assembleur fournit ce composant au bloc « Compilateur de configuration » via son port d'entrée « *comp* ». Après avoir manipulé et compilé les ressources du composant reçu, ce bloc enregistre les ressources compilées via son port de sortie « *rc* » dans le bloc « Bibliothèque des ressources pédagogiques », tout en générant une configuration contenant un paquetage de ressources compilées et prêtes à l'exploitation et en l'enregistrant dans le bloc « Bibliothèque de configuration » via son port de sortie « *conf* ».

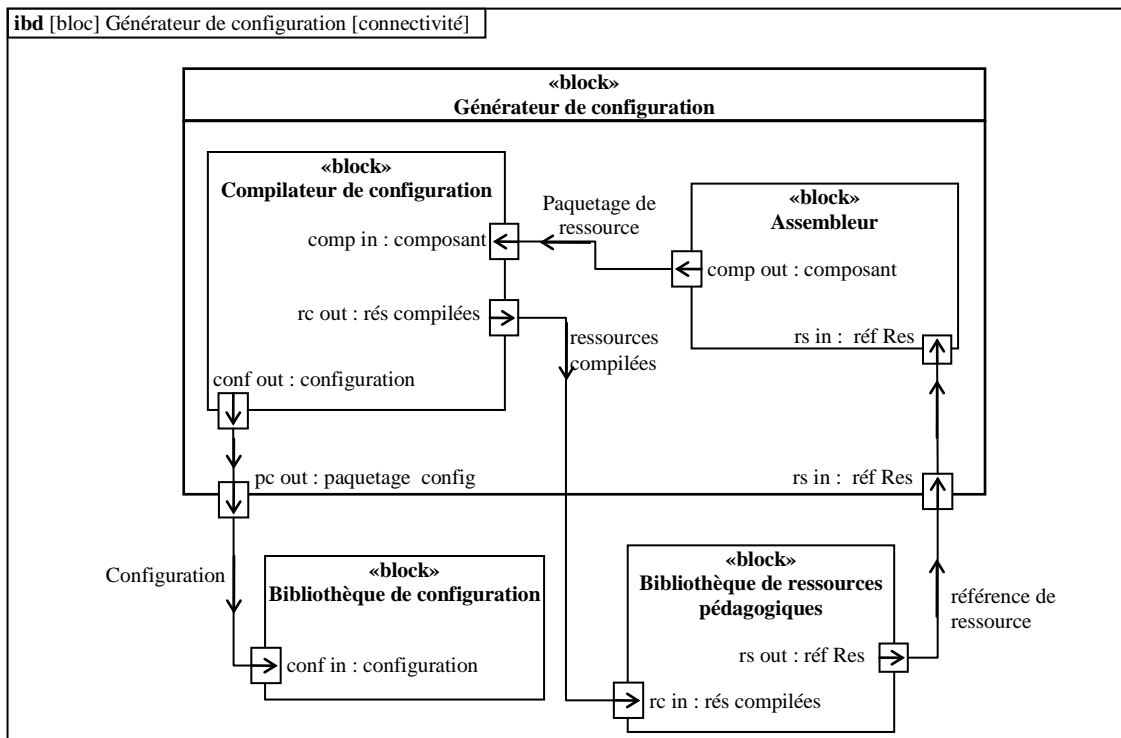


Figure 3.28 : Connectivité entre les blocs du « générateur de configuration ».

À l'aide du schéma de communication du bloc « Générateur de configuration », nous pouvons décrire les activités conduisant à réaliser les fonctionnalités adressées à ce bloc. L'assembleur prépare les éléments qui vont constituer une configuration. Ces éléments sont manipulés (complétés, modifiés, mis à jour, etc.) par l'auteur et/ou le technicien afin de les rendre exploitables dans une session d'e-TP.

Le processus de génération d'une configuration commence par l'assemblage, en récupérant les références des ressources choisies par l'auteur depuis la « Bibliothèque des ressources pédagogiques » (références de programmes API, de documentations et de scénarios pédagogiques). Puis, l'Assembleur assemble ces ressources dans un paquetage des ressources génériques « éditables » sous la forme d'un composant (voir Figure 3.29). Les ressources du composant sont éditées, éventuellement, par l'auteur et compilées dans des formats exploitables au sein d'un e-TP (par exemple : un programme au format OpenPLC est compilé dans un format propriétaire prêt à être téléchargé sur un API). La compilation de toutes les ressources d'un composant génère, au sein du bloc « Compilateur de configuration », un paquetage de ressources compilées sous une configuration. Une configuration se génère donc par la compilation des ressources d'un composant, et contient des ressources exploitables par l'instructeur sur un SAP dans une session d'e-TP (voir Figure 3.30). La configuration générée

est ensuite enregistrée dans la bibliothèque de configurations (cf. §3.3.2.7), et les ressources compilées sont enregistrées dans la bibliothèque de ressources pédagogiques.

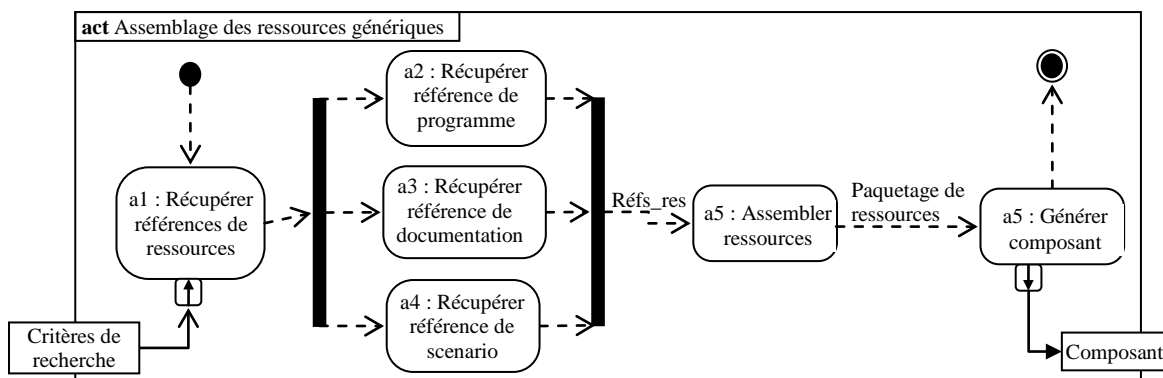


Figure 3.29 : Diagramme d'activité illustrant l'assemblage des ressources génériques par l'assembleur.

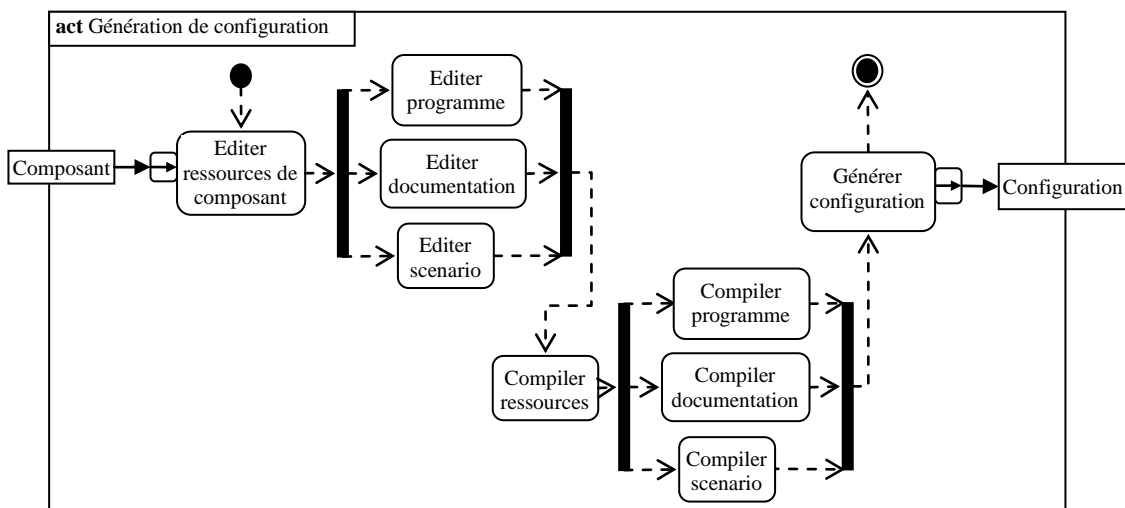


Figure 3.30 : Diagramme d'activité illustrant la génération de configuration par le générateur de configuration.

3.3.2.7 Structure de la bibliothèque de configurations

Jusqu'à ce point de conception, nous sommes partis d'une série d'exigences générales (EX1 à EX3) pour motiver la création de deux blocs essentiels de notre système, qui gèrent le passage des ressources de l'état brut (ressources disponibles) à l'état des ressources produites.

La bonne exploitation des configurations générées exige qu'elles soient organisées de manière similaire à la « Bibliothèque de ressources pédagogiques », afin de faciliter leur accessibilité par les instructeurs. Cela se fait dans la « Bibliothèque de configurations », qui est consacrée au stockage des configurations, à l'aide des métadonnées qui indexent les configurations sur mots clés, sujets, types de SAP, etc.

Néanmoins, contrairement aux ressources utilisables pour plusieurs objectifs et dans plusieurs contextes, les configurations correspondent à des cas spécifiques. L'organisation des configurations doit donc satisfaire le contexte pédagogique dans lequel elles seront exploitées. Cependant, elles pourront répondre à deux contextes d'utilisation:

- *contexte de machine* pour configurer une ou plusieurs parties d'un SAP, par exemple : dans un système de magasin rotatif (cf. §4.4.1), l'auteur prépare la configuration d'une partie de ce SAP contenant les programmes API qui fait fonctionner le magasin rotatif. Dans ce cas, cette configuration pourra être (ré)utilisée avec une autre configuration pour une autre partie du système ;
- *contexte d'e-TP* pour configurer un SAP selon scénario prédéfini et testé par l'auteur afin d'y dérouler une session d'e-TP. La configuration d'e-TP réutilise celles de la machine pour établir une relation entre les fonctions de la machine et le LMS.

Le bloc « Bibliothèque de configuration » a pour but de faciliter le stockage puis l'exploitation des configurations selon différents contextes d'utilisation (voir Figure 3.31). Elle sera utilisée pour enregistrer les configurations élaborées par l'auteur, et pour récupérer celles instanciées par l'instructeur. Ce bloc est composé de :

1. un bloc « **Bibliothèque de composants d'e-TP** » : il représente le répertoire des configurations génériques dont toutes les ressources sont éditables. Le composant (voir Figure 3.16 et Figure 3.17) dans cette bibliothèque représente l'état initial (générique) d'une configuration, et le passage à l'état exploitable se fait à travers le processus de génération de configuration ;
2. un bloc « **Bibliothèque de cas scénarisés** » : il représente le répertoire des configurations dont toutes les ressources sont compilées, testées et prêtes à l'emploi dans une session d'e-TP. La configuration dans cette bibliothèque est générée par la compilation de toutes les ressources d'un composant, et réutilise celles stockées dans la bibliothèque précédente ;
3. un bloc « **Bibliothèque de configurations machine** » : il représente les configurations d'une ou plusieurs parties d'un SAP. La configuration dans cette bibliothèque représente un ensemble de programmes API faisant fonctionner un ou plusieurs parties d'un SAP, indépendamment du contexte pédagogique. Il s'agit ici des configurations au niveau de la machine seulement.

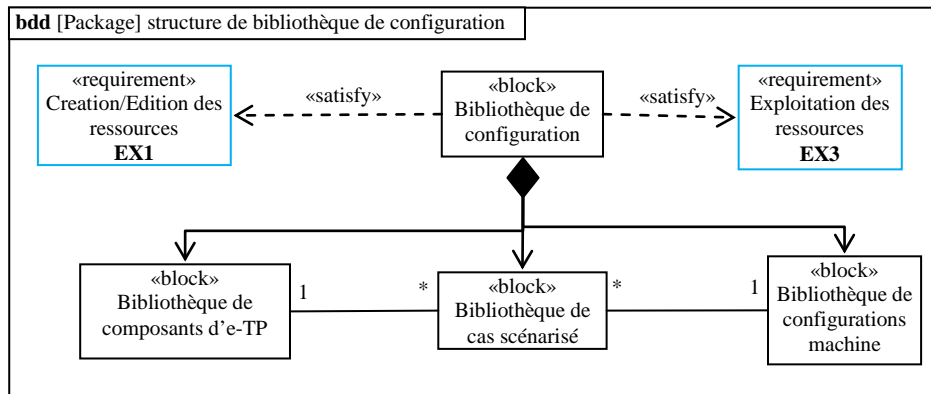


Figure 3.31 : Structure de la bibliothèque de configurations.

Afin que les sous blocs de la « Bibliothèque de configuration » réalisent les fonctionnalités de stockage, ils communiquent entre eux. Nous identifions, dans un *bdd*, les ports d'entrées et de sorties gérant les flux de données de chaque sous bloc en définissant le type de donnée manipulé par chaque port (voir Figure 3.32, Tableau 3.9).

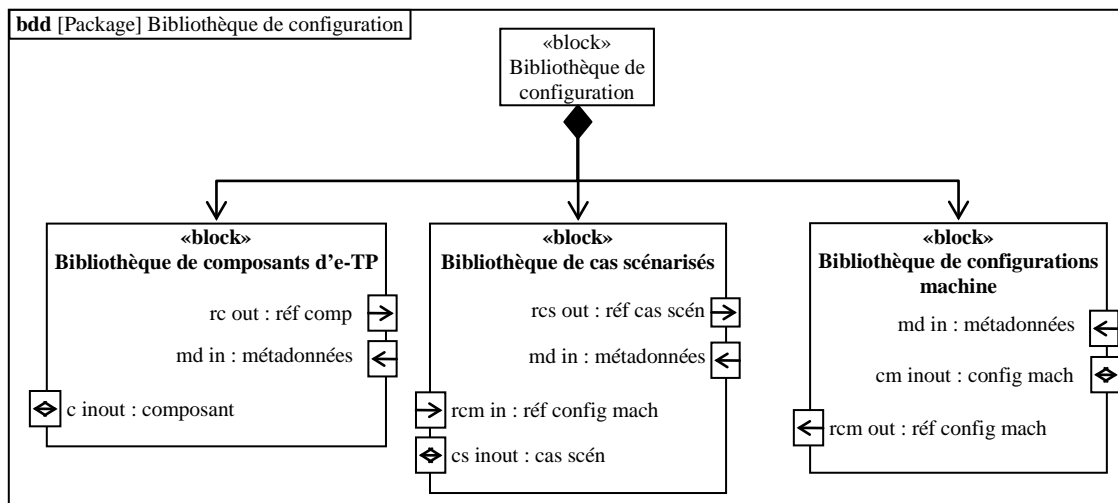


Figure 3.32 : Entrées et sorties des sous-blocs de la bibliothèque de configuration.

Bibliothèque de composants d'e-TP	Bibliothèque de cas scénarisés	Bibliothèque de configurations machine
<ul style="list-style-type: none"> - port <i>E/S</i> « <i>c</i> » pour recevoir et fournir un paquetage de composant créé par l'auteur - port <i>E</i> « <i>md</i> » pour recevoir des métadonnées indexant les composants insérées dans cette bibliothèque - port <i>S</i> « <i>rc</i> » fournissant la référence d'un composant 	<ul style="list-style-type: none"> - port <i>E/S</i> « <i>cs</i> » pour recevoir une configuration d'une ou plusieurs parties d'un SAP prête à l'exploitation dans e-TP créée par l'auteur, et récupérer une configuration par l'instructeur - port <i>E</i> « <i>md</i> » pour recevoir des métadonnées indexant les configurations insérées dans cette bibliothèque - port <i>S</i> « <i>rsc</i> » fournissant la référence de configuration d'un cas scénarisé dans cette bibliothèque - port <i>E</i> « <i>rcm</i> » pour recevoir la référence d'une configuration machine 	<ul style="list-style-type: none"> - port <i>E/S</i> « <i>cm</i> » pour recevoir une configuration de machine créée et compilée par l'auteur, et en récupérer par l'auteur aussi pour l'associer à un cas scénarisé - port <i>E</i> « <i>md</i> » pour recevoir des métadonnées indexant les configurations insérées dans cette bibliothèque - port <i>S</i> « <i>rcm</i> » pour fournir la référence d'une configuration de cette bibliothèque

Tableau 3.9 : Ports associés aux sous blocs du « Bibliothèque de configuration ».

Les configurations dans cette bibliothèque sont constituées, par assemblage, de ressources stockées dans la bibliothèque de ressources pédagogiques (cf. §3.3.2.5). Une configuration stockée dans la bibliothèque de configuration (que ce soit un composant d'e-TP, configuration de cas scénarisé ou bien celle de machine) ne stocke aucune ressource, mais des références vers des ressources stockées dans la bibliothèque des ressources pédagogiques. Les interconnexions des blocs de la bibliothèque de configuration sont décrites dans la Figure 3.33. Tous les blocs ont un port d'entrée/sortie dédié pour insérer/récupérer un paquetage de configuration selon le type de configuration à manipuler dans chaque bloc.

Le port d'entrée « *md* » associé à chaque bloc a pour objectif d'indexer les configurations enregistrées dans chaque sous bibliothèque. Par contre, le bloc « Bibliothèque de configurations machine » fournit la référence d'une de ses configurations via le port « *rcm* » au bloc « Bibliothèque de cas scénarisés » pour qu'elle puisse l'utiliser, alors que le bloc « Bibliothèque de cas scénarisés » fournit la référence d'une de ces configurations au bloc « Bibliothèque de composants d'e-TP » via le port de sortie « *rsc* ». La bibliothèque de cas scénarisés représente le répertoire dans lequel l'instructeur peut récupérer une configuration de session d'e-TP dont toutes les ressources sont exploitables. La séparation entre la configuration machine et la configuration d'un cas scénarisé d'e-TP, qui ne contient qu'une référence vers une configuration machine, a pour objectif de simplifier la génération de plusieurs versions de configuration de session d'e-TP à partir d'un composant (configuration générique) chacune différente en termes d'objectif pédagogique, mais similaire en termes de configuration de la machine, par exemple : dans le cas du magasin rotatif (cf. §4.4.1), la

même configuration du magasin pourra être utilisée dans un scénario de démonstration illustrant le fonctionnement du système, et dans un scénario d'exercice dans lequel on demande aux apprenants de modifier le programme existant et de développer ceux synchronisant un translateur avec le magasin rotatif.

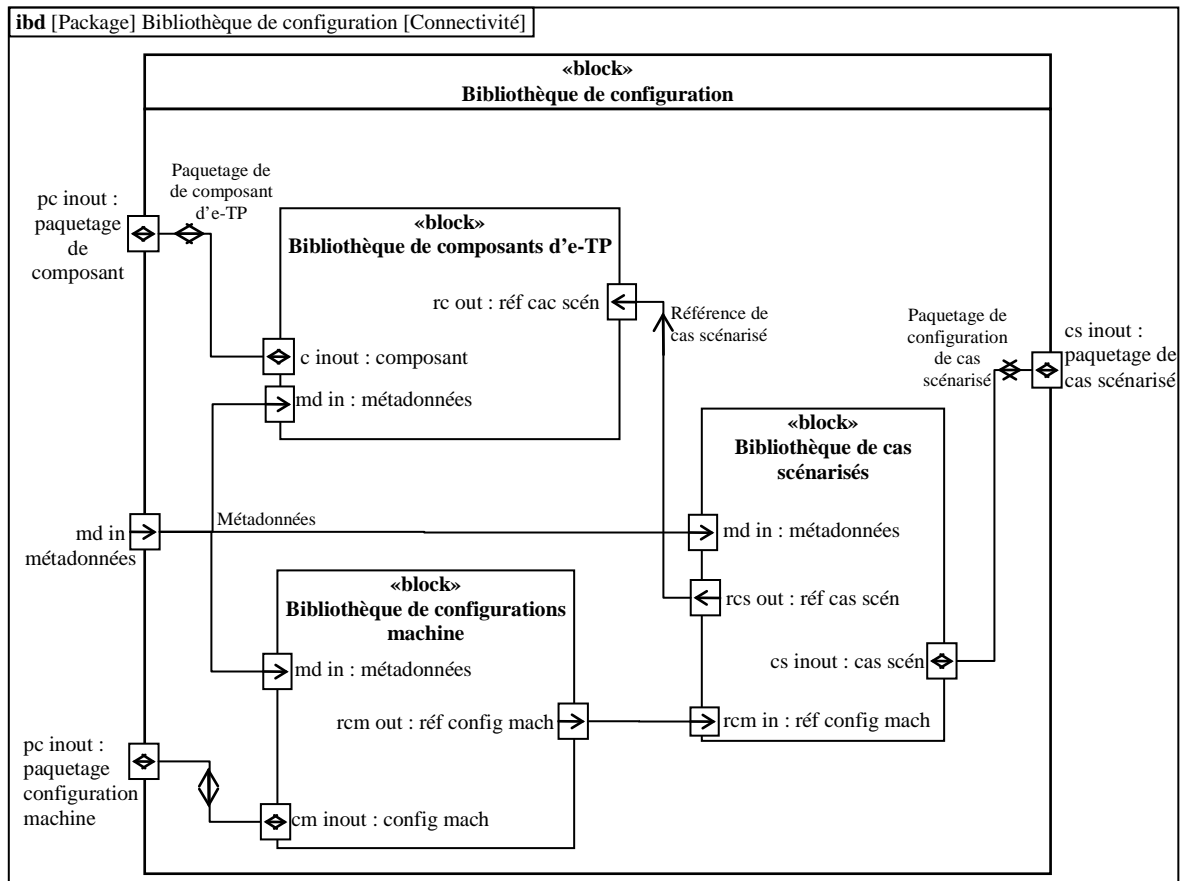


Figure 3.33 : Connexions entre les blocs de la « bibliothèque de configuration ».

3.3.2.8 Structure de l'installateur de configuration

L'installateur de configuration satisfait l'exigence **EX3₃** (cf. §3.3.1.1). Cet outil a pour objectif d'automatiser au maximum les opérations d'installation d'une configuration de session d'e-TP, de manière à satisfaire les sous-exigences **EX3₄** et **EX3₅**.

L'autonomie de l'instructeur dépend de la capacité du système à automatiser le téléchargement des programmes API sur les automates (composants), et la distribution des scénarios pédagogiques et documentation sur LMS. Si le configurateur ne peut pas automatiser certaines opérations, il avertit l'instructeur en lui fournissant les instructions nécessaires pour installer la configuration manuellement (par exemple : fournir un document d'instructions). Le bloc « Installateur de configuration » comprend deux sous blocs (voir Figure 3.34) :

- un sous-bloc « **installation automatique** » : satisfaisant l'exigence **EX3₅**. Il se charge des opérations exécutables automatiquement : le téléchargement des programmes API sur les automates via son sous bloc «Téléchargement sur SAP » et le téléchargement des ressources pédagogiques sur le LMS via son sous bloc « Téléchargement sur LMS » ;
- un sous-bloc « **Installation manuelle** » : satisfaisant l'exigence **EX3₄**. Il représente les opérations que l'instructeur doit réaliser manuellement lors de l'installation d'une configuration. L'installateur dans ce cas avertit l'instructeur si une configuration choisie a des opérations à réaliser manuellement.

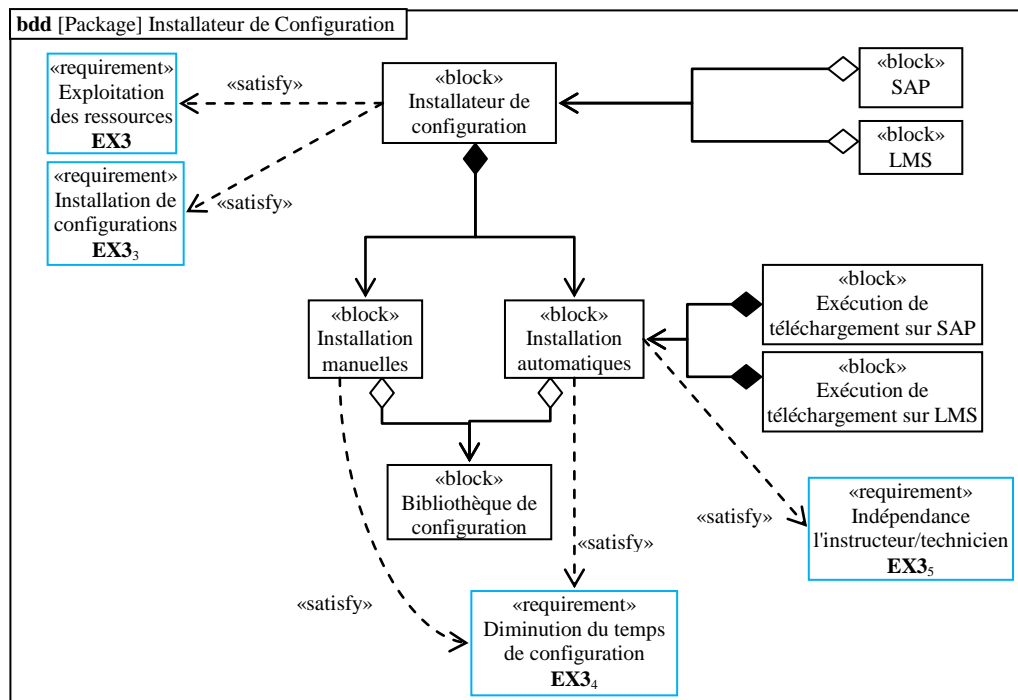


Figure 3.34 : Structure de l'installateur de configuration.

Ces deux sous blocs de l' « Installateur de configuration » communiquent, pour réaliser les fonctionnalités d'installation, à travers des ports d'entrées interconnectés (voir Figure 3.35, Tableau 3.10).

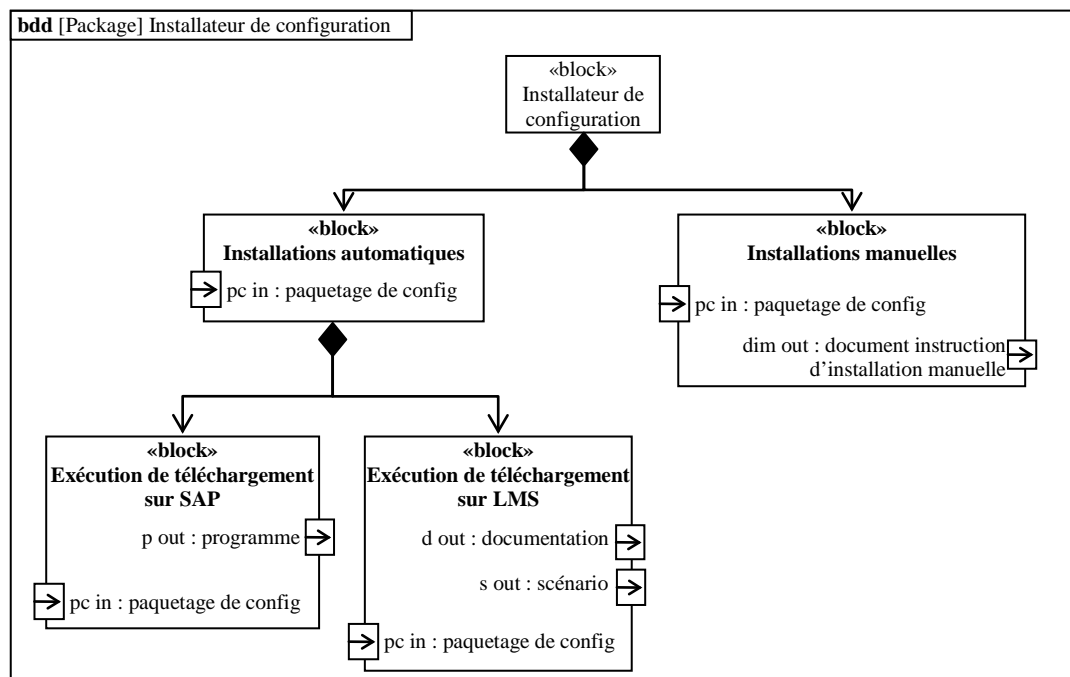


Figure 3.35 : Entrées et sorties des sous-blocs du générateur de configuration.

Installations automatiques		Installations manuelles
	<ul style="list-style-type: none"> - port <i>E</i> « <i>pc</i> » pour recevoir un paquetage de configuration d'e-TP à installer, et le rediriger vers les deux sous blocs - port <i>S</i> « <i>om</i> » pour fournir une liste des opérations d'installation à réaliser manuellement par l'instructeur 	<ul style="list-style-type: none"> - port <i>E</i> « <i>pc</i> » pour recevoir un paquetage de configuration d'e-TP à installer manuellement à l'aide d'une notice d'instructions associée à ce paquetage - port <i>S</i> « <i>dim</i> » pour fournir un document d'instructions sur l'installation manuelle du paquetage de configuration d'e-TP
Exécution de téléchargement sur LMS	<ul style="list-style-type: none"> - port <i>S</i> « <i>d</i> » pour fournir une documentation à installer sur LMS, extraite du paquetage de configuration - port <i>S</i> « <i>s</i> » pour fournir un scénario 	
Exécution de téléchargement sur SAP	<ul style="list-style-type: none"> - port <i>S</i> « <i>p</i> » pour fournir un programme API à installer sur un SAP, extraite du paquetage de configuration 	

Tableau 3.10 : Ports associés aux sous blocs de l' « Installateur de configuration ».

Cette connexion aide à automatiser, au maximum, la fonctionnalité d'installation d'une configuration (voir Figure 3.36). En effet, ce bloc reçoit un paquetage de configuration contenant toutes les ressources d'e-TP en le dirigeant vers le sous bloc « Installations automatiques ». L'installation se déroule donc en effectuant d'abord les opérations automatiques (téléchargement de : programme API sur l'automate, documentation et scénario pédagogique sur LMS). Puis, si le paquetage contient des ressources d'e-TP à installer manuellement, le bloc « Installations manuelles » fournit une liste des opérations manuelles via le port « *dim* ». Ce bloc se charge d'extraire un document d'instruction depuis le

paquetage de configuration sur les opérations à effectuer pour réaliser une installation manuelle des ressources d'e-TP (voir Figure 3.37).

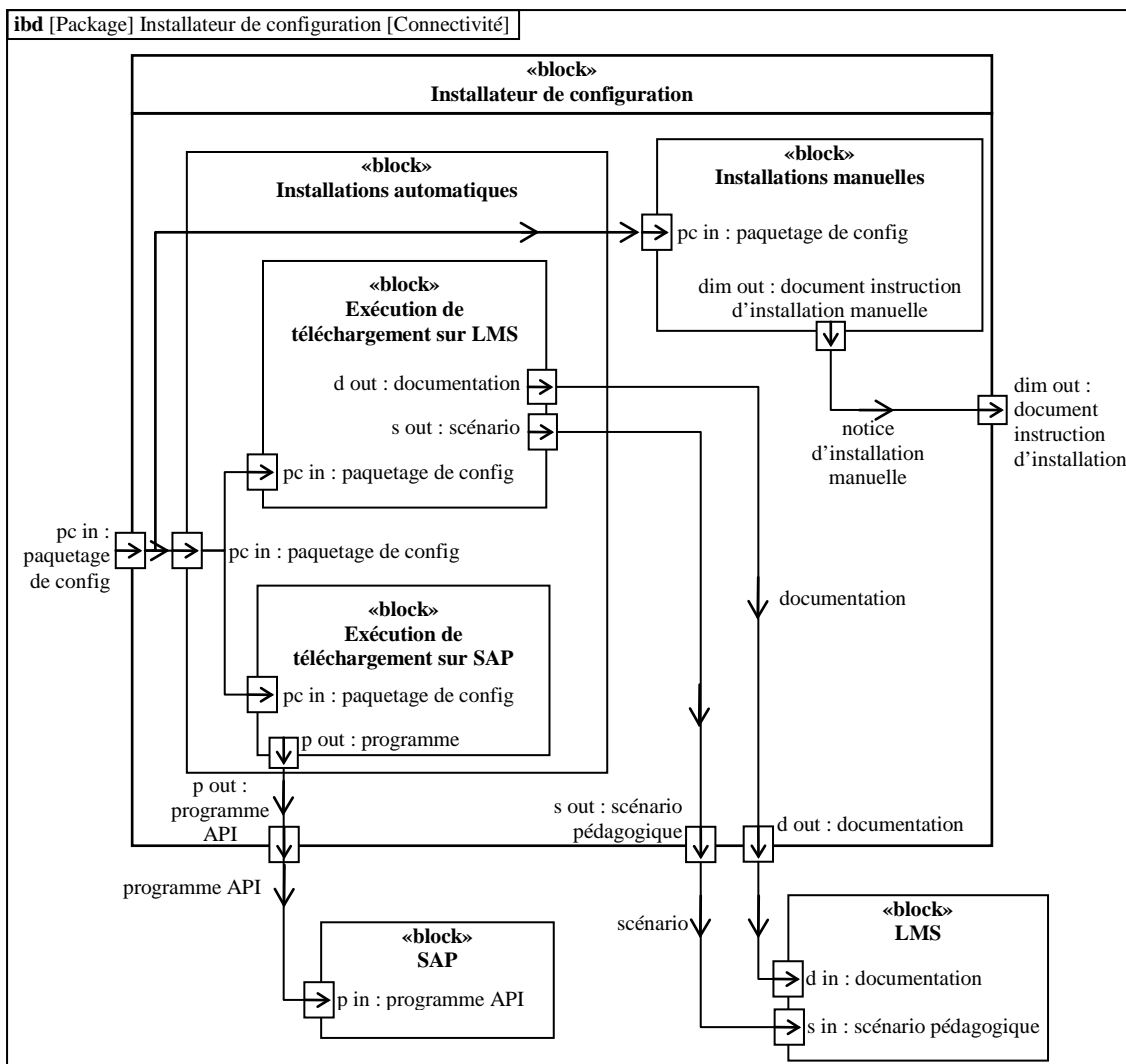


Figure 3.36 : Connexions entre les blocs de l' « Installateur de configuration ».

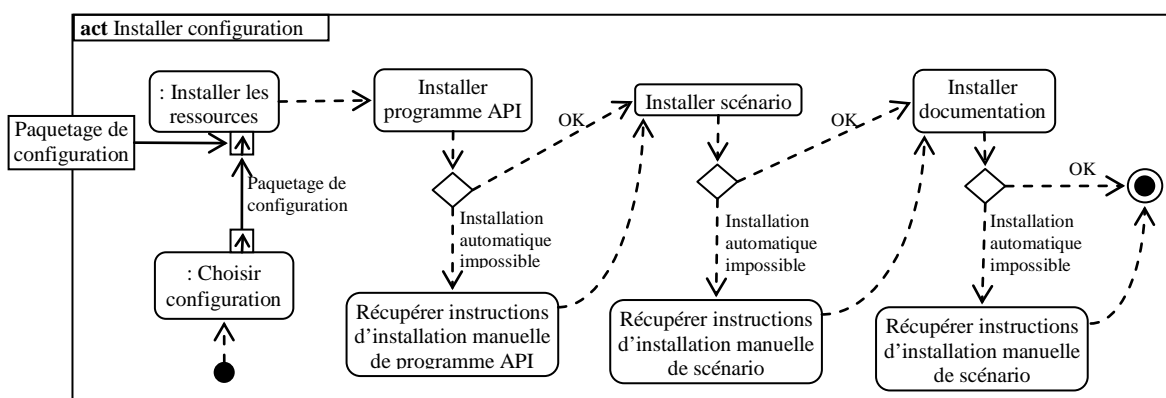


Figure 3.37 : Logique de fonctionnement de l'installateur de configuration.

3.4 Vérification de conformité du modèle en fonction de la spécification

Nous sommes partis des besoins initiaux des utilisateurs en les transformant en exigences de système.

Nous avons précisé les entrées et sorties du système et leur format (objectifs pédagogiques, programmes API, ...), ce qui nous a conduit aux flux internes : transformations successives nécessaires et données intermédiaires. L'ensemble de ces étapes est toujours filoguidé par ces exigences.

L'étape suivante consiste à vérifier si notre modèle se conforme à sa spécification. C'est-à-dire, pour le modèle PIM développé jusqu'à ce point, **vérifier** si le modèle **dispose** les éléments qui **peuvent satisfaire** chaque exigence énoncée du système visé. Une exigence non satisfaite par un (ou plusieurs) élément(s) du modèle pourra produire incohérence dans son implémentation, et engendrer une absence de fonctionnalité(s) requise(s) par les utilisateurs du système visé. À ce stade, il est important de noter que la vérification que nous faisons ici n'implique que de déterminer **qui** (élément du modèle) **fait quoi** (exigence). **La validation fonctionnelle du système sera, elle, décrite dans le chapitre suivant à travers des expérimentations.** L'activité marquée en rouge « Vérifier la conformité du modèle à sa spécification » dans la Figure 3.38 représente l'étape actuelle que nous développons dans cette section au cours du processus global de développement du configurateur.

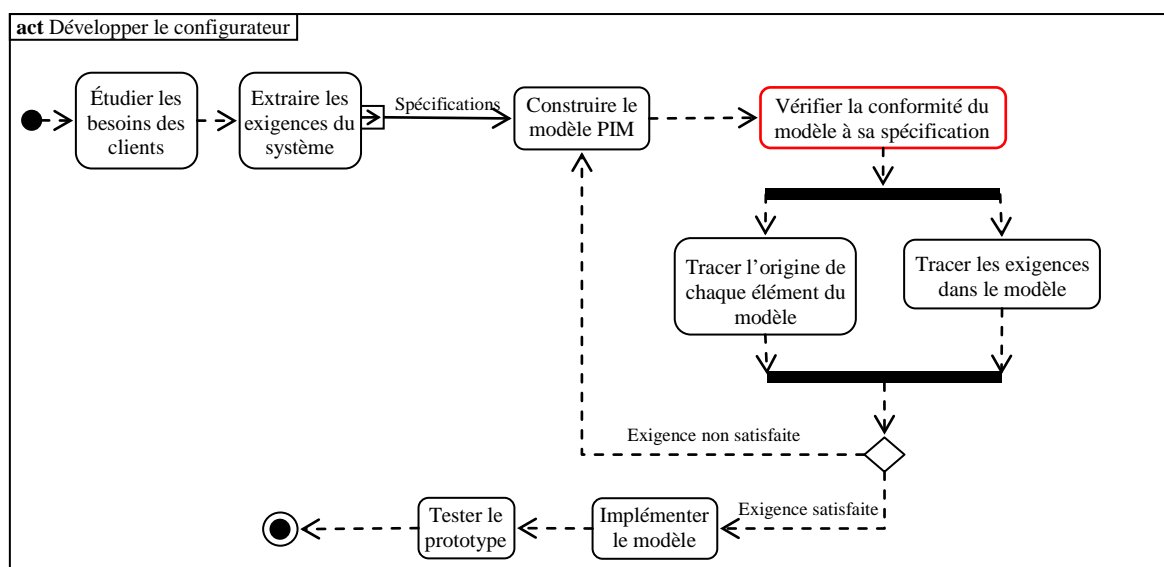


Figure 3.38 : Étapes de développement du configurateur.

La méthode employée est celle-ci fournie par SysML, la *traçabilité des exigences* (Requirements Traceability). Elle consiste à tracer les exigences à travers le modèle pour retrouver les éléments qui les satisfaites, grâce aux relations établies entre ces exigences et les éléments du modèle pendant le développement des différents diagrammes. Cela aide le concepteur et le développeur à comprendre le but, et donc le fonctionnement, de chaque élément en référant à son origine, tout en s'assurant ne pas rater d'implémenter des fonctionnalités répondantes à toute exigence lors de l'implémentation du modèle.

Dans le modèle, une exigence est satisfaite « *satisfy* » par un élément structurel (bloc), et/ou raffinée « *refine* » par un élément comportemental (cas d'utilisation). La traçabilité des exigences peut être effectuée à l'aide d'autres relations (« *derive* », « *satisfy* », « *verify* », « *refine* », « *trace* » et « *copy* ») [Friedenthal et al.-2008], mais ici nous avons limité la traçabilité à deux relations :

- la relation « *refine* » : le « raffinement » d'une exigence clarifie son sens ou son contexte. Cette relation réduit l'ambiguïté de l'exigence en l'associant à un autre élément du modèle qui la clarifie. Nous avons choisi de raffiner les exigences en les associant à des cas d'utilisation, chaque cas d'utilisation étant associé à un diagramme de séquence qui précise cette exigence. Ainsi, chaque exigence peut être tracée dans le contexte d'utilisation qu'elle implique.
- la relation « *satisfy* » : la satisfaction d'une exigence est apportée par un élément structurel du modèle. Nous avons tenté de satisfaire toutes les exigences en les associant à des blocs.

Terminons la phase de conception en constatant que le modèle *PIM* du configurateur dispose des éléments pouvant répondre à toutes les exigences énoncées, sans préciser comment faire (discuté dans le chapitre suivant), mais juste une **précision** de « **qui fait quoi** » grâce à une représentation tabulaire des relations mutuelles entre les exigences et les autres éléments du système. En effet, l'outil de conception utilisé, Artisan Studio Uno⁵⁶, dispose du concept « *Traceability Matrix* » qui est une matrice représentant les relations entre les différents éléments d'un modèle et ses exigences. La Tableau 3.11 liste les exigences définies pour notre système et leur satisfaction par des éléments du modèle développé. Cette matrice illustre la satisfaction des exigences par des éléments structurels et comportementaux.

⁵⁶ Voir le site web : <http://www.atego.com/products/artisan-studio-uno/>

Il est possible d'enrichir cette matrice en montrant les relations entre les exigences et les diagrammes d'activités, diagrammes de machines à état, etc. Mais pour ne pas surcharger la matrice, nous avons opté de ne pas la trop détailler en montrant la satisfaction de chaque exigence par un seul type d'éléments structurels (blocs) et un seul type d'éléments comportementaux (cas d'utilisation).

Dans le chapitre suivant, nous nous basons sur le modèle PIM pour implémenter un prototype du configurateur, puis nous le vérifions à travers des expérimentations.

Traces des exigences dans le modèle		Exigences																			
		Accès aux ressources	Recherche de configurations pour l'instructeur	Recherche de ressources et de configuration pour l'auteur	Gestion des dépendances	Vérification de l'état des ressources	Création/Édition des ressources	Sauvegarde des ressources	Gestion des permissions	Gestion de modification des ressources	Gestion et mises à jour des ressources	Exploitation des ressources	Installation de configurations	Génération de configuration	Indépendance l'instructeur/technicien	Diminution du temps de configuration	Assemblage de ressources	Compilation de ressources	Accès aux ressources sans contraintes		
Cas d'utilisation	Exigence	EX1 ₂	EX3 ₂	EX3 ₁	EX1 ₅	EX1 ₄	EX1	EX1 ₁	EX1 ₃	EX1 _{5_1}	EX1 _{5_2}	EX3	EX3 ₃	EX2	EX3 ₅	EX3 ₄	EX2 ₁	EX2 ₂	EX3 ₆		
		Gestion des ressources	✓																		✓
	Fonctions principales		✓	✓																	
	Gestion d'évolution des ressources		✓	✓	✓	✓														✓	
	Utilisation de configurateur		✓	✓											✓	✓				✓	
	Accéder aux ressources	✓	✓	✓																✓	
	Construire bibliothèque de composant						✓	✓									✓			✓	
	Donner permission aux ressources	✓																		✓	
	Donner permission de configuration								✓											✓	
	Modifier configuration									✓	✓								✓	✓	
	Modifier ressource									✓	✓									✓	
	Génération de configuration													✓						✓	
	Utiliser configurations de SAP											✓	✓							✓	
Blocs	Bibliothèque de configuration		✓	✓		✓	✓		✓	✓	✓	✓								✓	
	Bibliothèque de documentation	✓		✓	✓	✓	✓	✓												✓	
	Bibliothèque programmes API	✓		✓	✓	✓	✓	✓												✓	
	Bibliothèque de scénario	✓		✓	✓	✓	✓	✓												✓	
	Bibliothèques des ressources pédagogiques	✓		✓	✓	✓	✓		✓	✓	✓	✓								✓	
	Installeur de configuration												✓	✓							
	Générateur de configuration													✓			✓	✓		✓	
	Installations automatiques													✓	✓	✓					
	Assembleur													✓	✓	✓	✓	✓			✓
	Compilateur de configuration													✓	✓	✓		✓			✓

Tableau 3.11 : Matrice de traçabilité des exigences.

3.5 Conclusion sur la modélisation

Nous avons précisé dans cette partie, la structure globale du système en réponse aux besoins, ses principaux composants et l'utilisation d'un configurateur de SAP dans un contexte d'e-TP. Nous sommes partis de la spécification du configurateur pour justifier la structure du modèle développé, tout en le nourrissant avec les diagrammes comportementaux expliquant le fonctionnement de ses éléments. Ce modèle répond aux exigences définies au départ, du fait de la validation par traçabilité des exigences (voir Tableau 3.11). La matrice de traçabilité montre que toutes les exigences sont satisfaites (et éventuellement affinées) par un ou plusieurs éléments du modèle. Nous pouvons donc considérer que le modèle répond aux besoins attendus.

Le modèle élaboré correspond au modèle *PIM* de la démarche MDA. Les diagrammes développés dans cette phase peuvent être exploités, dans la phase suivante de cette étude, quelle que soit la plate-forme ou la technologie de réalisation adoptée. Nous allons dans le chapitre suivant nous attacher à décrire l'implémentation des fonctionnalités du système avec les choix de technologie et de la plate-forme.

4. Implémentation et validation expérimentale

Introduction

Ce chapitre présente l'implémentation du modèle du configurateur ainsi que les outils utilisés pour réaliser les expérimentations. Les fonctionnalités fournies par le configurateur sont illustrées à travers des expérimentations sur deux dispositifs :

- un magasin vertical de petite taille (échelle 1/5) ;
- le transfert libre de l'atelier de l'AIP Primeca de l'INSA de Lyon.

Ces expérimentations ont pour but de valider notre configurateur d'automatismes industriels, et démontrer sa valeur ajoutée dans un contexte pédagogique.

Une autre expérimentation, dans le domaine de l'informatique, est également présentée afin de démontrer l'extensibilité du configurateur dans d'autres domaines scientifiques. Nous terminons ce chapitre par une analyse des retours sur ces expérimentations.

4.1 Plate-forme d'implémentation

Nous présentons le processus d'implémentation du modèle du configurateur et les critères de choix de la plate-forme et de la technologie d'implémentation. Nous profitons des diagrammes élaborés dans le chapitre précédent pour implémenter les fonctionnalités de prototype du système.

Le configurateur est principalement destiné aux instructeurs et auteurs d'e-TP en automation. Ces utilisateurs appartiennent à plusieurs « écoles », chacune utilisant des ordinateurs et des systèmes d'exploitation différents (Windows, Linux et Mac OS) (cf. Tableau 3.4, « **EX3₆** »). Nous avons donc choisi de réaliser le configurateur sous la forme d'une application web accessible de partout et indépendante d'un système d'exploitation particulier.

Le configurateur doit permettre l'enregistrement de ressources de types différents (scénarios pédagogiques, documents PDF, programmes API, ... cf. §3.3.1.3 - **EX1₁**), leur indexation et leur traitement. Pour générer des configurations, nous avons donc choisi de nous greffer sur un système de gestion de contenus (CMS - *Content Management System*) existant. Ce choix nous libère de la programmation des fonctions de base (création, édition,

suppression de contenus, gestion des droits d'accès, ..., cf. §3.3.1.3 - **EX1**, **EX1₃**, **EX1₄**, **EX1₅**) d'une application web gérant différents types de documents avec différents utilisateurs, publics, ... afin de nous concentrer sur les fonctions spécifiques du configurateur.

Nous avons opté pour un CMS libre, en l'occurrence, Drupal⁵⁷, du fait de sa maturité (apparu en 2001, actuellement en version 7) et de son extensibilité (ajout aisé et bien documenté de nouvelles fonctions par programmation de modules développés en langage PHP).

Drupal est une application web open-source, installable sur de nombreux logiciels serveurs web (libres ou non) et compatible avec la plupart des moteurs de base de données actuels (libres ou non) (voir Figure 4.1). Ce choix initial laisse relativement ouvertes les possibilités d'installation par toute Direction des Systèmes Informatiques de tout établissement.

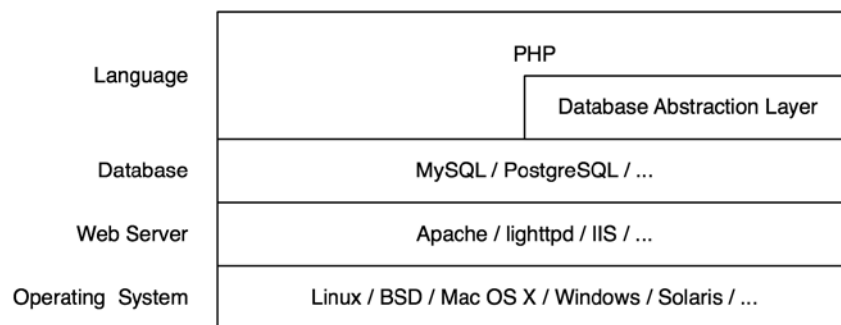


Figure 4.1 : "Technology stack" de Drupal⁵⁸.

Drupal est constitué d'un noyau fonctionnel chargé de réagir lors de la réception des requêtes et de bâtir la réponse en faisant appel aux fonctions de base (gestion des utilisateurs, droits d'accès, taxonomies, modèles, ...) et éventuellement aux modules d'extension [Vandyk-2010] (voir Figure 4.2).

⁵⁷ Voir <http://drupal.org/>

⁵⁸ Extrait de [Vandyk-2010] p.2.

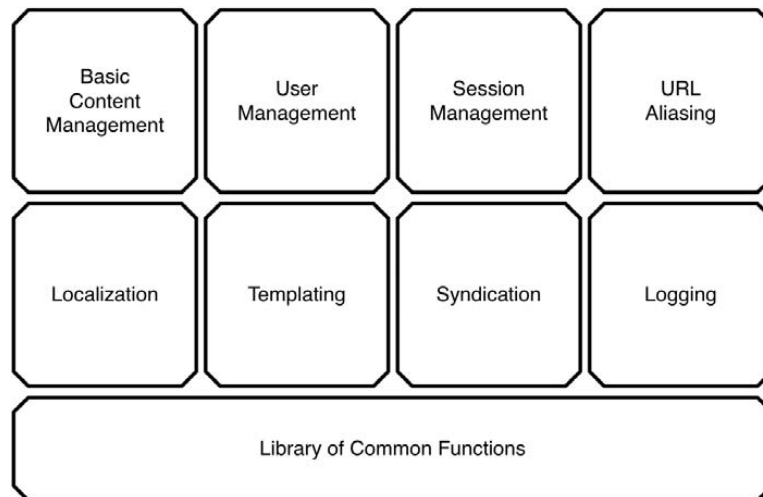


Figure 4.2 : Vue du noyau de Drupal⁵⁹.

À côté de l'adoption de Drupal pour développer la plate-forme de notre système de configurateur, nous avons adopté des outils auteurs pour la création/édition des ressources (programmes, documentation et scénarios pédagogiques). Ainsi Unity Pro (de Schneider Electric) a été utilisé pour la création/édition des programmes API, tandis que Opale⁶⁰ a été utilisé pour gérer des ressources pédagogiques. Opale est une chaîne éditoriale facilitant la production de ressources (cours, présentations, vidéos, exercices, ... à imprimer ou mettre en ligne sur un LMS) à but pédagogique. Les documents produits visent de nombreux usages:

- support visuel (écran et/ou imprimé) pour un cours en présentiel comme en auto-formation ;
- documentation (imprimée et/ou en ligne) ; formation à distance (évaluation en ligne) ;
- échange et partage de contenus pédagogiques entre enseignants et institutions.

Pour ce faire, la chaîne éditoriale Opale produit des supports : imprimables (ODT pour Open Office et PDF), web (HTML), web SCORM pour une diffusion sur les plates-formes de formation LMS (SCORM 1.2 et 2004) et diaporama (format HTML) pour la projection.

Le choix de logiciels libres non seulement facilite l'implémentation mais également est un gage de réutilisabilité de cette plate-forme. Seul Unity déroge à cette règle du fait que nous n'avons pas connaissance d'un logiciel libre aussi mature pour le développement de programmes automatés.

⁵⁹ Extrait de [Vandyk-2010] p.3.

⁶⁰ Voir : <http://scenari-platform.org/>

4.2 Réalisation d'un prototype

Cette section présente le prototype de configurateur réalisé afin de fournir aux auteurs et instructeurs en automatisme les fonctionnalités les aidant à configurer des sessions d'e-TP.

4.2.1 Fonctions principales

Dans le chapitre précédent nous avons présenté les activités adressées au configurateur dans un diagramme d'activité (voir Figure 3.22). Le flux de ces activités décrit les fonctions globales du système que nous implémentons dans notre prototype.

Partant de la spécification du configurateur et en réponse à l'exigence principale **EX_1** « Création/Édition des ressources » (voir Tableau 3.2), la fonction de manipuler chaque type de ressources est adressée à un logiciel dédié : Unity est utilisé pour manipuler des programmes API, et Opale est utilisé pour manipuler toute sorte de documentations ainsi que les scénarios pédagogiques. La version actuelle du configurateur ne fournit pas des fonctionnalités natives pour manipuler ces ressources directement via son interface, il est donc indispensable de profiter des logiciels externes pour atteindre cet objectif. Le stockage des ressources ainsi que leur indexation sont adressés à Drupal.

En réponse à l'exigence principale **EX_2** « Génération de configuration » (voir Tableau 3.3), Drupal se charge de réaliser la fonction d'assemblage des ressources puis de générer une configuration. La compilation des ressources s'adresse à nouveau aux logiciels externes (à Unity pour compiler les programmes API, et à Opale pour compiler les documentations et les scénarios pédagogiques).

Finalement et en réponse à l'exigence **EX_3** « Exploitation des ressources » (voir Tableau 3.4), la recherche des ressources par l'auteur et la recherche des configurations par l'instructeur s'adressent à Drupal. La fonction d'installation d'une configuration par l'instructeur s'adresse à Drupal.

Hormis le traitement du contenu de chaque type de ressources (création/édition et compilation des programmes API, documentations et scénarios), l'application web basée sur Drupal fournit toutes les fonctions requises dès le stockage des ressources à l'installation d'une configuration (cf. §3.3.2.4 , voir Figure 3.20 et Figure 3.22).

4.2.2 La structuration d'une configuration

Par soucis de clarté, nous présentons la façon dont les fonctions proposées au chapitre précédent sont réalisées dans l'ordre chronologique d'utilisation.

Créer une nouvelle configuration d'e-TP implique de dérouler deux phases : une phase générique où toutes les ressources d'une configuration restent réutilisables par d'autres auteurs, et une phase spécifique où toutes les ressources d'une configuration sont directement exploitables par les instructeurs. Cette séparation a pour but de satisfaire l'exigence « **EX3** » de la spécification du système (cf. §3.3.2.1) qui implique d'améliorer la réutilisation des ressources dans un contexte de leur exploitation. Plus précisément :

- **dans la phase de création d'une configuration générique** sont créés des « composants » représentant une configuration générique d'une session d'e-TP. L'auteur crée d'abord les programmes API à l'aide d'un logiciel externe (par exemple : PL7Pro cf. §4.4.1, Unity cf. §4.4.2) en les sauvegardant au format éditable (OpenPLC). Puis, il crée aussi les documentations et le scénario pédagogiques à l'aide d'un logiciel externe (Opale dans notre cas) en les sauvegardant dans un format ouvert et éditable (format XML adopté par Opale). L'auteur télécharge ensuite ces ressources sur l'application web en associant une description et des catégories à chacune (comme illustré dans l'utilisation du configurateur, cf. §4.3.1).

Via l'application web, la création du composant se fait par l'assemblage des ressources que l'auteur choisit dans une liste (organisée en classes). Le composant généré est enregistré dans le configurateur en tant que configuration générique dont toutes les ressources sont éditables ;

- **pendant la phase de génération d'une configuration spécifique** : l'auteur compile, à partir d'une configuration générique, les ressources associées pour générer une version spécifique à une session d'e-TP. Après avoir téléchargé un paquetage contenant tous les fichiers associés au composant choisi, l'auteur fait appel à des logiciels externes (les mêmes utilisés pour leur création) pour les éditer si nécessaire puis les compiler. Dans notre cas, l'édition et la compilation des programmes API se fait via Unity (ou PL7Pro), et l'édition/compilation des documentations et scénario pédagogique se fait via Opale.

Une fois la compilation terminée, les fichiers compilés sont rassemblés dans un paquetage que l'auteur télécharge sur le configurateur. Ce dernier extrait le contenu de ce paquetage en les enregistrant dans la bibliothèque de ressources pédagogiques, puis une configuration ayant la même structure de composant utilisé est créée avec des liens vers les nouvelles ressources. De cette manière, chaque ressource générique du composant original se voit associée à une version compilée.

Un exemple illustre ce principe (voir Figure 4.3) : sur le système de magasin vertical (cf. §4.4.1), nous avons développé deux configurations génériques « *Comp-Magasin-Vertical 1* » et « *Comp-Magasin-Vertical-Translateur 1* ». Ces deux configurations partagent un même élément de programme : « *Programme-Base-Magasin 1* ». La génération d'une configuration spécifique à partir de ces deux configurations génériques implique de compiler toutes leurs ressources de manière cohérente (en ôtant d'éventuels conflits de versions compilées). En réponse aux exigences **EX1_{5_1}** et **EX1_{5_2}** (cf. §3.3.1.3), nous avons mis en place le processus suivant :

- la compilation du composant « *Comp-Magasin-Vertical 1* » va générer une configuration spécifique « *Config-Magasin-Vertical 1* » (cf. §4.4.1.4) liée à la version compilée du programme « *Programme-Base-Magasin 1* » ;
- la compilation du composant « *Comp-Magasin-Vertical-Translateur 1* » créera une configuration « *Config-Magasin-Vertical-Translateur 1* » liée à une autre version compilée du programme « *Programme-Base-Magasin 1* » ;
- le configurateur relève ce conflit et le résout en demandant à l'auteur/instructeur de choisir l'une ou l'autre de ces versions compilées ;
- la version choisie de ce programme compilé est alors clonée et associée à la configuration spécifique résultante.

La modification d'une configuration dont le composant d'origine partage une (ou plusieurs) même ressource avec d'autres composants n'implique pas de modifier les configurations générées par la compilation de ces composants. Ainsi, le partage d'une même ressource compilée entre plusieurs configurations (composants compilés) reste un choix laissé au libre arbitre de l'auteur, avec prudence.

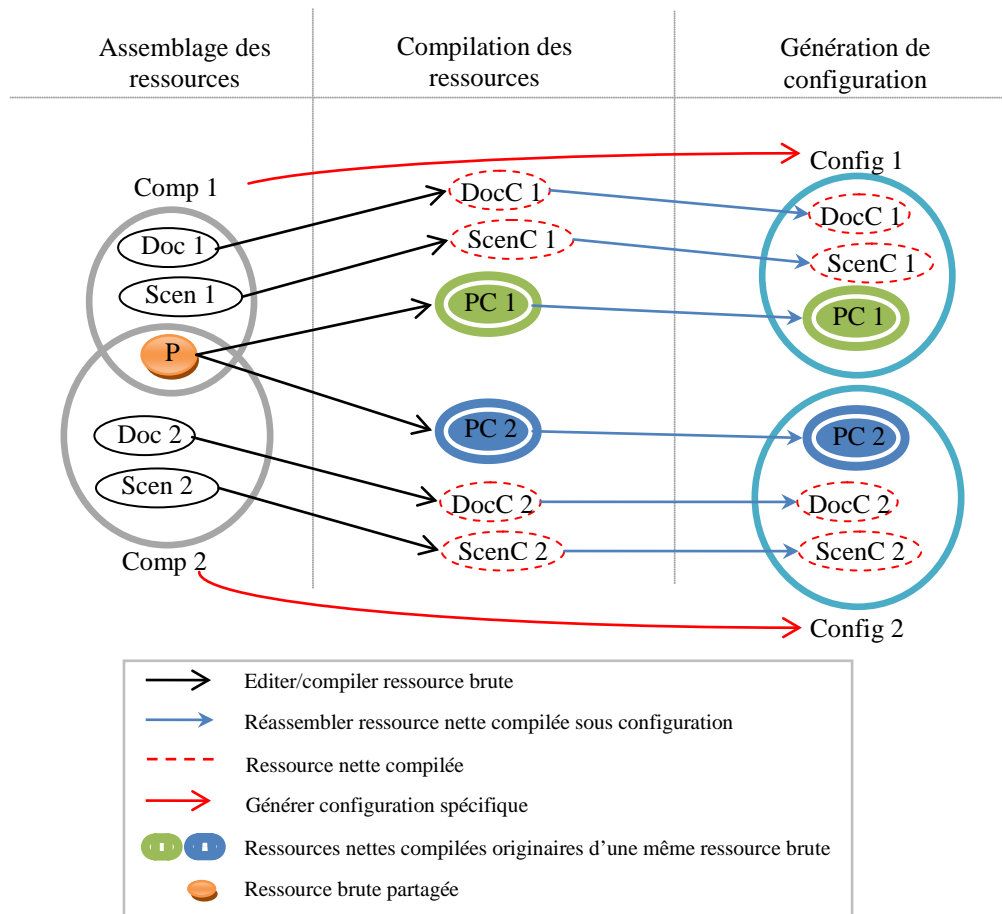


Figure 4.3 : Passage de configuration générique à configuration spécifique.

Remarque : certaines ressources ne nécessitent pas d'être compilées, elles peuvent être utilisées telles quelles (par exemple : document sur l'initialisation d'une machine au format PDF).

4.2.3 Catégorisation et indexation des ressources

En réponse aux exigences **EX3₁** « Recherche de ressources et de configuration pour l'auteur » et **EX3₂** « Recherche de configurations pour l'instructeur », le configurateur incite l'auteur à catégoriser les ressources à ajouter dans la bibliothèque de ressources pédagogiques (documentation, scénario pédagogique ou programme API), afin d'assurer leur traçabilité et simplifier leur recherche. Le configurateur fournit également à l'auteur la possibilité de créer ses propres catégories auxquelles les éléments ajoutés appartiennent, si les catégories existantes ne correspondent pas à ses intérêts. Cette catégorisation de ressources facilitera la création d'un composant (configuration générique) en regroupant toutes les ressources se conformant à des critères donnés (pédagogiques ou techniques).

Dans notre cas, les ressources sont à exploiter essentiellement sur des SAP pour servir un ou plusieurs objectifs pédagogiques. Il en résulte deux indicateurs pour ces ressources : le type de composant d'automatisation à exploiter et l'objectif pédagogique. La catégorisation elle-même des ressources est assurée via trois ontologies : « Perspective composant », « Perspective objectifs pédagogiques » (qui fournissent une recherche des ressources basée sur type et discipline), et « Perspective système réel » (qui fournit une recherche des ressources en se basant sur des mots clefs).

4.2.3.1 Perspective « composant »

Cette ontologie fournit une hiérarchie de composants d'automatisation (*parts*) susceptibles d'être présents dans un système automatisé (par exemple, dans un système de convoyage on peut trouver des composants tels que des vérins pneumatiques, des capteurs de présence, des moteurs d'entraînement, etc.). Cette hiérarchie définit un groupe de classes pour la recherche de ressources à base de composants d'automatisation.

4.2.3.2 Perspective « système réel »

Cette ontologie est une hiérarchie de balises qui représentent une ou plusieurs parts des systèmes dans un atelier ou un laboratoire ou (par exemple : magasin vertical#1, vérin#2, poste d'assemblage#4, système#1, etc.). Elle définit des instances de classes dans « Perspective de composant ». Par exemple : dans un atelier, le vérin d'un magasin vertical sera référencé par la balise « *vérin#1* » de l'ontologie « Perspective système réel », et sera au même temps référencé par la classe « *vérin* » extrait de l'ontologie « Perspective composant ». Cette ontologie associe une ressource à un système ou sous-système.

4.2.3.3 Perspective des objectifs pédagogiques

Cette ontologie fournit des termes généraux correspondant à des thèmes pédagogiques (supervision, ordonnancement, robotique, MES, etc.). Chaque terme peut avoir une ou plusieurs références vers des classes de l'ontologie « Perspective composant », ce qui classe les ressources sur des critères techniques et pédagogiques retrouvables. Par exemple : le terme (MES) dans l'ontologie des objectifs pédagogiques des références vers la classe (*Capteur de présence*) et la classe (*Poste de transfert libre*) de la « Perspective composant » (cf. §4.4.2 : TP sur le système du transfert libre).

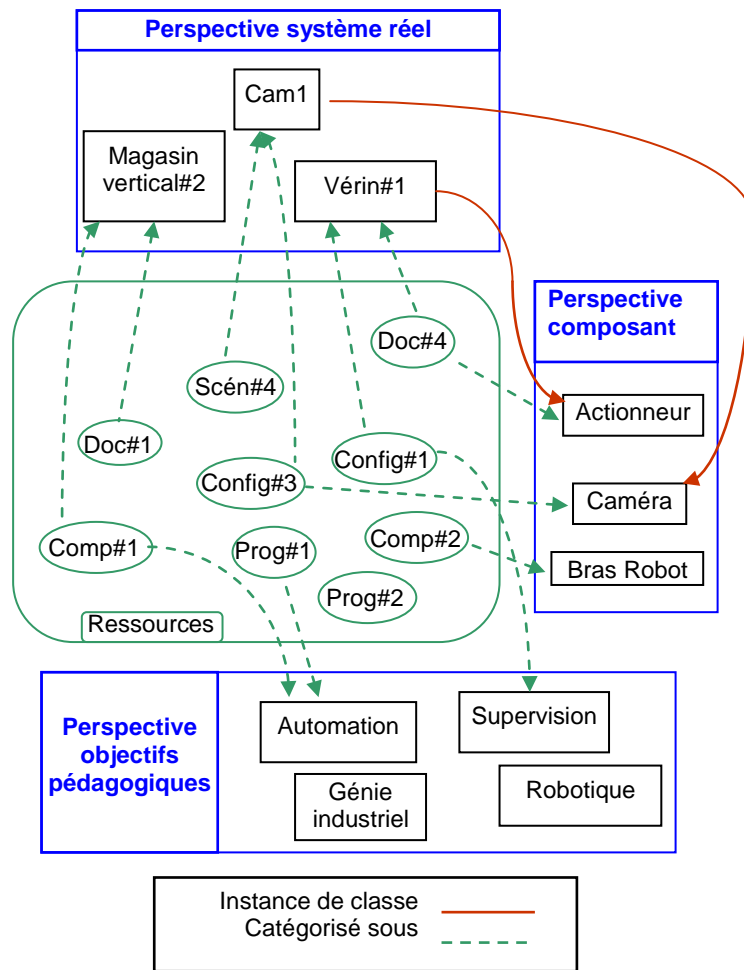


Figure 4.4 : Taxonomies.

Le schéma de catégorisation des ressources (voir Figure 4.4) donne la possibilité d'effectuer une recherche selon trois points de vue :

1. une recherche par **mot clef** basée sur la « Perspective système réel ». Par exemple, la recherche des ressources selon le mot « Vérin » va récupérer une liste de résultats comme : « vérin#1 », « vérin#2 », « vérin pneumatique », etc. sans que l'utilisateur ait besoin de préciser ni le type de SAP cible, ni le contexte de leur utilisation. La composition de plusieurs mots clefs est possible ;
2. une recherche **sémantique** basée sur :
 - une recherche par **type de composant** qui récupère toutes les ressources associées à une classe de composants. Par exemple : « vérin », « bras robot » ;
 - une recherche par **objectif pédagogique** qui récupère toutes les ressources associées à un thème donné, en classant les résultats de recherche par type de

composant. Pour ce faire, nous avons fait appel aux fonctions de taxonomie de Drupal (voir Figure 4.5).

Ces fonctionnalités analysent la requête (textuelle) de l'utilisateur en extrayant des mots clefs (ou des compositions de mots clefs), correspondant à ceux des deux ontologies.

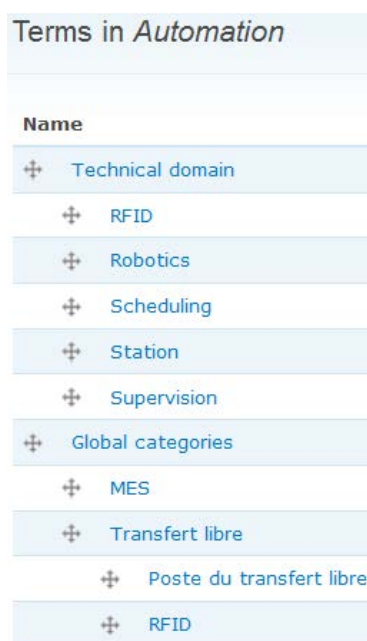


Figure 4.5 : Exemple des termes correspondant à des thèmes pédagogiques.

La possibilité de chercher des ressources selon une discipline, un mot clé ou un sujet aide à vérifier s'il existe dans la base de données des ressources potentiellement utiles : ressources issues d'autres configurations, réalisées par un collègue, documents que l'auteur ne dispose pas, programmes nécessaires (pour la session) que l'auteur n'a pas développé lui-même, ...).

4.2.4 Réutilisabilité des ressources

L'un des besoins/objectifs identifiés (cf. §3.3.1.1 : **BA₃** et cf. §3.3.1.3 : **EX3**) est d'améliorer la réutilisabilité des ressources exploitées au sein des différents TP afin de réduire le temps de préparation des sessions et de configuration. La séparation entre la phase générique d'une configuration et la phase spécifique sert l'objectif de réutilisabilité des ressources. Cette séparation est implémentée dans le prototype du configurateur où l'auteur :

- soit utilise un composant déjà développé en le compilant pour générer une configuration spécifique ;

- soit le modifie ;
- soit construit son propre composant à partir des ressources disponibles, ou après les avoir créées, puis le compile pour générer la configuration souhaitée.

L'indexation des ressources à l'aide des trois ontologies (cf. §4.2.3) participe donc à la réutilisabilité des ressources, en fournissant à l'utilisateur le moyen de chercher des ressources préexistantes selon différents critères de recherche.

4.3 Utilisation du configurateur

Cette section illustre le fonctionnement du prototype de configurateur dans trois contextes : pendant la création des ressources, pendant la génération de configuration et pendant l'installation de configuration de session d'e-TP.

4.3.1 Préparation des ressources

Rappelons ici les cas d'utilisation type présentés dans §3.3.1.5 , notamment la phase de création de configuration (voir Figure 3.9). Les ressources pédagogiques (documentations et scénarios) créées par Opale sont une arborescence de fichiers, notamment des fichiers de description du contenu, dans leur format natif, enregistrés au format *XML*. Ce format est utilisé par l'auteur dans la phase éditable (générique) de création de ressources. Dans leur version compilée, le format utilisé peut être *ODT*, *PDF*, *HTML*, selon les préférences de l'instructeur. Les programmes API développés sont enregistrés au format OpenPLC, basé sur un codage en XML, qui simplifie l'assemblage des sous programmes. Le format d'une version compilée d'un programme dépend du logiciel externe que l'utilisateur adopte pour ce but (par exemple : Unity ou PL7Pro).

La création d'une configuration suit deux étapes :

A. Création des ressources de base :

- création des catégories ;
- création des programmes API ;
- création des documentations pour les apprenants et les tuteurs ;
- création des scénarios d'apprentissage.

B. Création de configuration :

- création de composant assemblant toutes les ressources (fichiers) nécessaire pour une session d'e-TP ;

- compilation des ressources du composant assemblé puis génération de la configuration finale de la session d'e-TP.

L'application web développée pour dérouler cette tâche dispose d'une page d'accueil⁶¹ regroupant toutes les opérations que l'utilisateur peut effectuer.

La première opération consiste donc à créer des catégories de ressources (voir Figure 4.6).

Figure 4.6 : Page de création des catégories.

Les autres types de ressources (programme, documentation, scénario) ont une structure identique :

- nature du contenu porté : programme, documentation ou scénario ;
- intitulé de la ressource ;
- catégories auxquelles elle appartient ;
- description textuelle de la finalité de cette ressource ; contenu sous forme de fichier attaché dans un format éditable (l'outil d'édition dépendant de la nature de cette ressource).

⁶¹ Page d'accueil : <https://liris.cnrs.fr/~sarnous/configurator/drupal-6.22-v2/?q=node/2680>

La Figure 4.7 donne un exemple de création de programme. On y trouve les différents champs à remplir par l'utilisateur. La page de création des autres éléments (documentation et scénario) est identique.

Note: the fields followed by the (*) are required and must be filled

Program name *: Programme Poste 1 (compatible avec différentes versions de Unity Pro)

Select The categories to which this program can belong to *:

- MES
- RFID
- Transfert libre
- Poste du transfert libre

Program description

Ce programme est à télécharger sur le poste 1 dans un système du transfert libre. A cet élément est associé un le fichier de ce programme avec l'extension (*.sta) qui compatible avec différentes versions du logiciel Unity Pro. Ce type de fichier permet à se connecter à un API (PLC) en ouvrant le projet avec une version plus récente du Unity Pro.

Select the file of your program to upload *: C:\Users\Saher\Desktop Browse...

Note: the file name must not include the following characters: é, Ê, è, È, î, Î, ù, Û, ô, Ô

Simple description of the file: Programme Poste 1 (version *.sta compatible avec diffèrentes versions de Unity Pro)

Create program element

Figure 4.7 : Page de création d'un élément de programme.

La même approche est suivie pour la création des composants (configurations génériques), avec une différence : le composant ne contient que des références vers des éléments de ressources et/ou vers d'autres composants, et la création de composant n'implique l'utilisation d'aucun fichier source. L'utilisateur est d'abord amené à préciser les catégories décrivant ce composant (exemple : Figure 4.8), puis il est dirigé vers une page affichant les listes de ressources toutes filtrées selon les choix des catégories dans l'étape précédente (voir Figure 4.9).

Figure 4.8 : Sélection des catégories correspondant au composant avant sa création.

Assembled component name Poste 1 MES

Assembled component description
Ce composant rassemble les ressources liées au poste 1 dans le système du transfert libre

Add student documentations:

For Student	For Tutor	Description	Related Categories
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Description du Poste 1 du système du transfert libre [+info]	Related Categories (MES,Transfert libre,Poste du transfert libre)
<input type="checkbox"/>	<input type="checkbox"/>	Description du Poste 2 du système du transfert libre [+info]	Related Categories (MES,Transfert libre,Poste du transfert libre)
<input type="checkbox"/>	<input type="checkbox"/>	Description du Poste 3 du système du transfert libre [+info]	Related Categories (MES,Transfert libre,Poste du transfert libre)
<input type="checkbox"/>	<input type="checkbox"/>	Description du Poste 4 du système du transfert libre [+info]	Related Categories (MES,Transfert libre,Poste du transfert libre)
<input type="checkbox"/>	<input type="checkbox"/>	Description du Poste 6 du système du transfert libre [+info]	Related Categories (MES,Transfert libre,Poste du transfert libre)
<input type="checkbox"/>	<input type="checkbox"/>	Structure de poste du transfert libre [+info]	Related Categories (MES,Transfert libre,Poste du transfert libre)
<input type="checkbox"/>	<input type="checkbox"/>	Vue globale du système du transfert libre [+info]	Related Categories (MES,Transfert libre,Poste du transfert libre)

Add programs:

<input checked="" type="checkbox"/>	Programme Poste 1 (compatible avec différentes versions de Unity Pro) [+info]	Related Categories (MES,Transfert libre,Poste du transfert libre)
<input type="checkbox"/>	Programme Poste 2 (compatible avec différentes versions de Unity Pro) [+info]	Related Categories (MES,Transfert libre,Poste du transfert libre)
<input type="checkbox"/>	Programme Poste 4 (compatible avec différentes versions de Unity Pro) [+info]	Related Categories (MES,Transfert libre,Poste du transfert libre)
<input type="checkbox"/>	Programme Poste 6 (compatible avec différentes versions de Unity Pro) [+info]	Related Categories (MES,Transfert libre,Poste du transfert libre)
<input type="checkbox"/>	Programme Poste 7 (compatible avec différentes versions de Unity Pro) [+info]	Related Categories (MES,Transfert libre,Poste du transfert libre)
<input type="checkbox"/>	Programme Poste 7 (incompatible différentes versions) [+info]	Related Categories (MES,Poste du transfert libre)

Add learning scenarios:

<input type="checkbox"/>	Scénario de démonstration du système du transfert libre [+info]	Related Categories (MES,Transfert libre,Poste du transfert libre)
--------------------------	---	---

[Create assembled component](#)

[Create resources](#) [Edit resources](#) [Create configurations](#) [Configurations library](#) [Search](#)

Figure 4.9 : Page de création d'un composant.

Ce composant représente une configuration générique à compiler pour générer la configuration correspondant à une session d'e-TP. L'utilisateur peut à tout moment modifier les éléments de ressources de base et les composants sous forme de **révisions** (mémorisation des différentes versions d'un même élément : configurations génériques et composants).

4.3.2 Phase de génération de configuration

La création d'une configuration se base essentiellement sur les composants déjà créés. Une configuration est un ensemble de fichiers compilés prêts à fournir aux apprenants des documentations et scénarios pédagogiques via un LMS, et à initialiser un SAP.

Le diagramme de séquence de la Figure 4.10 illustre le processus de création d'une nouvelle configuration. L'auteur cherche à travers le moteur de recherche du configurateur, le composant approprié (voir Figure 4.11), puis en télécharge le contenu (sous forme de paquetage ZIP généré automatiquement). Ce mode de transfert sert à transmettre les fichiers source aux différents logiciels d'édition (Opale, Unity) puis les fichiers compilés de ces éditeurs au configurateur.

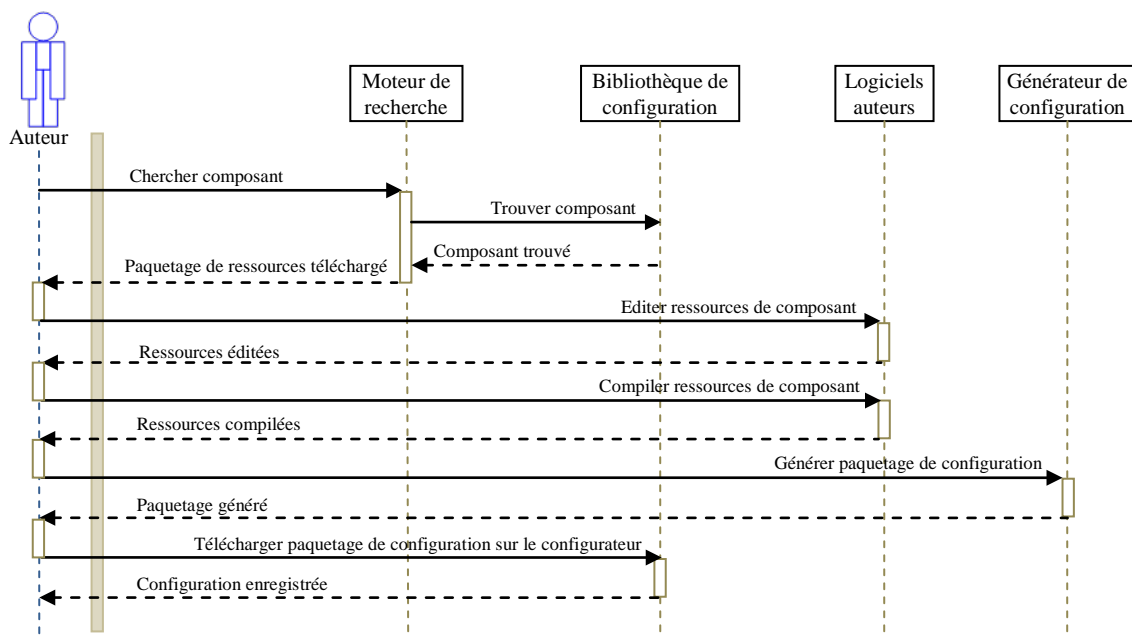


Figure 4.10 : Diagramme de séquence illustrant le processus de création de configuration.

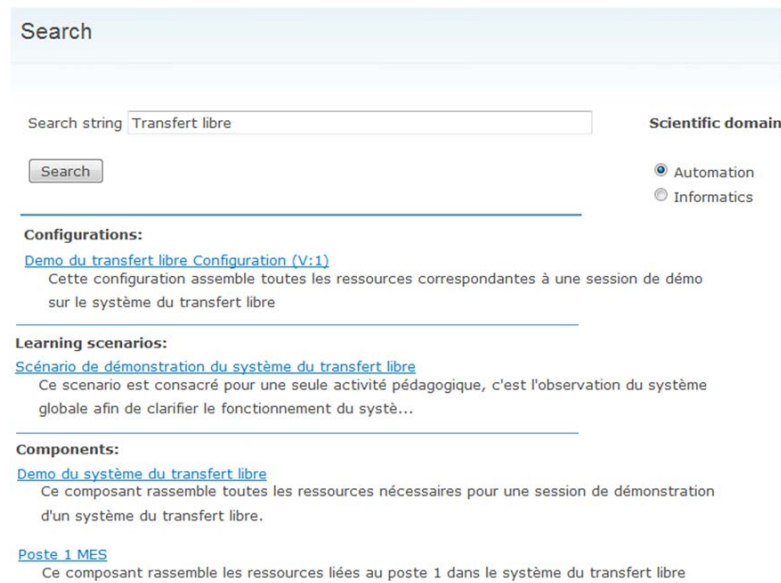


Figure 4.11 : Écran affichant les résultats de recherche lancée par l'utilisateur.

Avec la version client-serveur d'Opale il est possible d'automatiser l'échange de ces fichiers avec le configurateur (ce qui éviterait cette manipulation lourde), mais nous n'avons pas eu le temps de développer cette option et l'usage d'Opale en version locale est suffisant pour valider les concepts de ce projet.

Ce fichier ZIP (*assembled_component_archive.zip*) contient les éléments suivants :

- un dossier nommé *temp-configs* : regroupe tous les fichiers des ressources listées dans ce composant, qui nécessitent d'être compilées (voir Figure 4.12) ;
- une application (*Config_Launcher*) aidant à assembler les pièces de documentations et de scénarios pédagogiques dans Opale ;

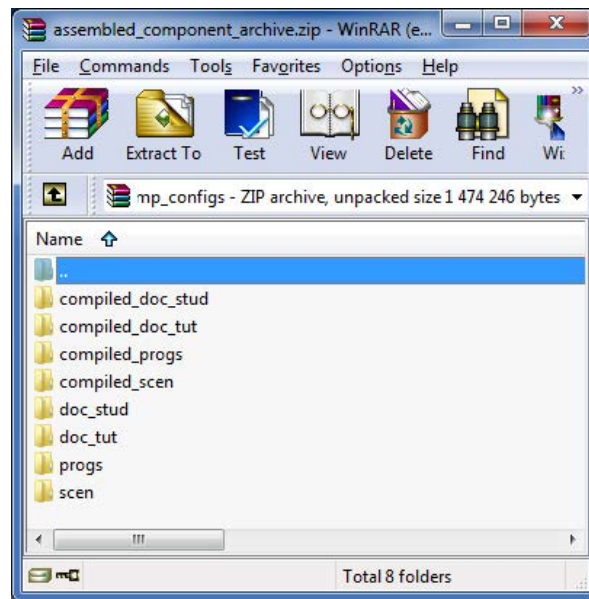


Figure 4.12 : Contenu du fichier ZIP téléchargé contenant les ressources à compiler.

Config_Launcher.exe est un outil aidant l’auteur à assembler localement des pièces de documentations ainsi que des pièces de scénarios téléchargés. Via cet outil, l’utilisateur retrouve une liste des fichiers sources dans le fichier ZIP ordonnés selon leurs types (documentation tuteur, documentation étudiant, scénario pédagogique, programme). Cet outil est capable de traiter tous les fichiers XML (qui se trouvent dans le fichier ZIP téléchargé) décrivant une ou plusieurs hiérarchies de ressources documentaires (documentations et scénarios) créées par Opale, et de générer un paquetage Opale rassemblant toutes ces ressources. L’auteur n’a qu’à compiler ces ressources assemblées via Opale pour en générer une version compilée. Cette application permet aussi de lancer les programmes API téléchargés en les ouvrant dans Unity pour les compiler.

Après avoir compilé toutes les ressources, l’auteur les met dans le paquetage ZIP qu’il charge dans le configurateur. Celui-ci associe ce paquetage à une configuration désormais stockée dans sa bibliothèque de configurations. À ce stade, l’auteur n’intervient plus et la configuration de la session d’e-TP est prête à l’emploi.

4.3.3 Phase d’exploitation

Avant de lancer le TP, l’instructeur cherche la configuration désirée dans le configurateur, puis télécharge un paquetage de configuration (fichier ZIP) contenant toutes les ressources nécessaires (voir Figure 4.13), et notamment l’application *Config_Launcher* dont le rôle est de télécharger les programmes API (sur les automates), le scénario pédagogique et les

documentations sur le LMS. Cette application n'est pas indispensable mais simplifie ces tâches techniques qui nécessiteraient une « qualification » de l'instructeur.

Si le paquetage dispose d'une notice sur des opérations à effectuer manuellement par l'instructeur, le configurateur l'avertit de sa présence en fournissant un lien vers cette notice.

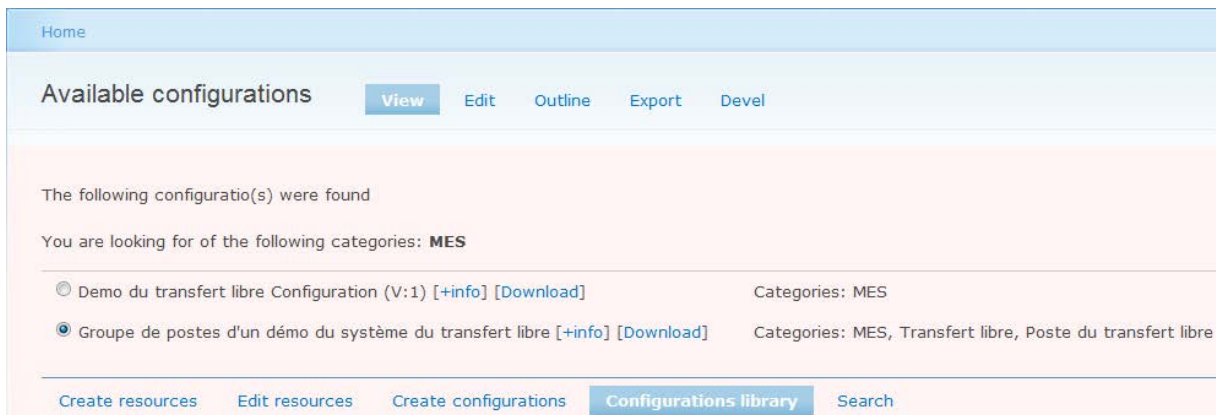


Figure 4.13 : Liste de configurations enregistrées dans la bibliothèque de configuration.

Le téléchargement du scénario pédagogique sur un LMS est optionnel ; l'instructeur peut toujours imprimer des versions (papier) accessibles depuis le configurateur. *Config_Launcher* affiche une liste de tous les programmes API, documentations et scénarios pédagogiques de la configuration téléchargée. L'instructeur peut alors ouvrir les programmes API dans le logiciel Unity Pro qui se charge de les télécharger sur les automates idoines.

Dans l'idéal, cette fonctionnalité de téléchargement devrait être exécutée par *Config_Launcher* mais dans le cadre de cette expérimentation, nous n'avons pas accès au protocole de transfert. De même pour le transfert des scénarios pédagogiques vers un LMS.

4.4 Application sur des environnements de TP en automatique

Cette section présente des exemples d'utilisation de notre configurateur afin de prouver son applicabilité dans des environnements d'e-TP en automatique. Nous présentons le protocole d'expérimentation et les objectifs qui valideront nos modèles et notre approche.

4.4.1 Cas du « Magasin Vertical »

Il s'agit d'un système automatisé appartenant à une plate-forme de TP d'Automatique au sein du Département « Génie Industriel » de l'INSA. Après une sommaire description, nous précisons en quoi il permet de mettre en lumière l'intérêt de ce travail de recherche via l'expérimentation et ses résultats (publié dans [Arnous et al-2012]).

4.4.1.1 Description du magasin vertical

Le magasin vertical est composé de (voir Figure 4.14) :

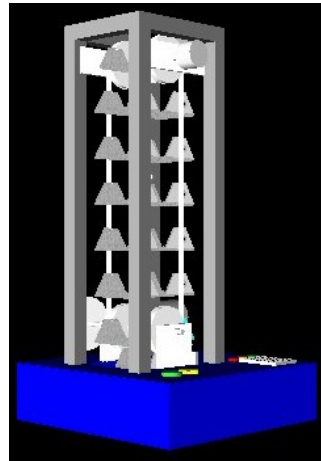


Figure 4.14 : Le magasin vertical.

- une **partie opérative** constituée de deux sous-systèmes :
 - un ensemble de nacelles entraînées par une courroie, pouvant accueillir chacune une pièce métallique cubique (voir Figure 4.15 - A, B) ;
 - un translateur rotatif pour le (dé)chargement de ces pièces via la porte du magasin (voir Figure 4.15 - C). Il s'agit d'un bras manipulateur à deux degrés de liberté (une rotation autour d'un axe vertical et une translation dans le plan horizontal). Ce translateur rotatif est commandé par un second API ;
- une **partie commande** constituée d'un coffret électrique industriel comportant les alimentations et les protections électriques ainsi que l'automate programmable TSX Micro de Schneider Electric.



A : Magasin vertical



B : Simulateur du magasin vertical



C : Bras rotatif avec les pièces métalliques

Figure 4.15 : Le magasin vertical et le bras rotatif avec les pièces métalliques.

Ce magasin vertical peut être utilisé sans le bras rotatif. Il stocke des pièces qui sont introduites et retirées manuellement via une porte située à l'entrée droite du magasin (voir Figure 4.16). Pour introduire (respectivement retirer) une pièce dans le magasin, il faut donc positionner une nacelle vide (respectivement pleine) en face de la porte, ouvrir la porte et placer (respectivement enlever) la pièce sur la nacelle. Un moteur entraîne en rotation une poulie crantée. Des nacelles portant des pièces sont fixées sur cette poulie et se déplacent verticalement dans le magasin. Utilisé avec le bras rotatif, l'échange des pièces a lieu via la porte de gauche du magasin.

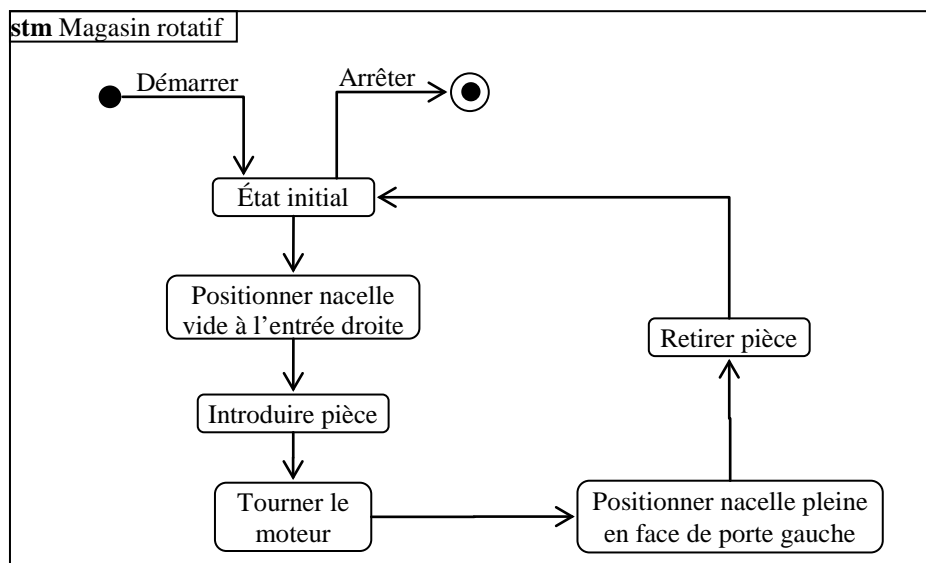


Figure 4.16 : Machine à état illustrant le fonctionnement du magasin rotatif.

Les objectifs pédagogiques des sessions organisées via ce dispositif varient entre l'apprentissage de la programmation d'un système automatisé, la maîtrise de l'interconnexion sûre entre deux systèmes automatisés indépendants et la maîtrise d'incidents.

Cette première expérimentation sert à justifier l'utilisation du configurateur sur un système (relativement simple), que ce soit en termes de création de ressources réutilisables que d'amélioration du temps de (re)configuration. Ce système étant modulaire, il peut démontrer la valeur ajoutée des fonctions de conception de configurations par réutilisation/assemblage.

L'outil de développement de programmes automate est ici PL7Pro de Schneider Electric (Unity sera utilisé dans la prochaine expérimentation). Le fait de pouvoir travailler avec un logiciel différent démontre l'indépendance du configurateur vis-à-vis des logiciels auteurs.

4.4.1.2 Expérimentation

Cette section présente la démarche d'expérimentation suivie, en utilisant le configurateur pour générer des activités de TP sur le magasin vertical.

Nous avons demandé à l'auteur de développer des configurations permettant aux étudiants de réaliser les activités suivantes :

- 1- programmer le magasin vertical seul (première activité) ;
- 2- programmer le magasin vertical en le synchronisant avec le translateur (deuxième activité).

Puis nous lui avons demandé de chercher (puis installer) la configuration de chaque activité de TP (installer les programmes API, et télécharger le scénario pédagogique ainsi que la documentation sur un LMS).

4.4.1.3 La préparation des ressources

Voici la démarche suivie par l'auteur :

1- Création des catégories décrivant les ressources :

La première opération consiste à créer des catégories pour mieux identifier les ressources insérées ultérieurement (programmes, documentations et scénarios pédagogiques) (cf. §4.3.1). Les catégories créées correspondent à chacune des 3 parties du dispositif :

- **Système séquentiel**: pour référencer les ressources utilisées par la programmation séquentielle du système ;
- **Magasin rotatif** : pour référencer les ressources utilisées pour le magasin rotatif ;
- **Translateur** : pour référencer les ressources concernant un poste du translateur.

La première catégorie est orientée « Perspective des objectifs pédagogiques » (cf. §4.2.3.3) tandis que les deux suivantes suivent la « Perspective de composant » (cf. §4.2.3.1).

2- Création des éléments de programmes API

Deux éléments de programmes API ont été créés par l’auteur :

- pour le magasin vertical : « **Programme-Base-Magasin 1** » (voir Annexe C) qui code, dans un Grafcet maître, l'enchaînement des principaux modes de marche du système : Etat initial, Prise d'Origine Machine, Mise à jour et Fonctionnement normal (voir Figure 4.17),
- pour le fonctionnement du translateur : « **Programme-Translateur 1** » suivant le même schéma.

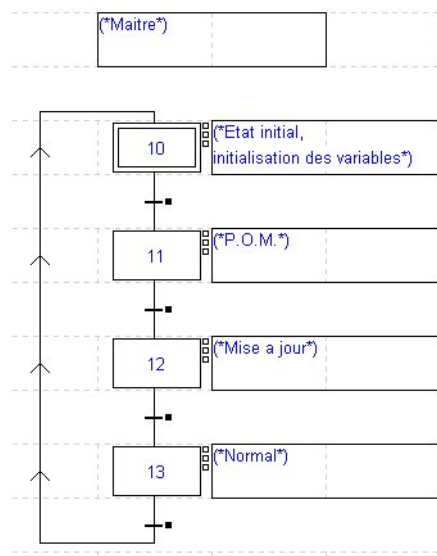


Figure 4.17 : Grafcet maître (pour le magasin vertical).

Lors de la création (insertion) des éléments de programmes dans le configurateur, l’auteur les a associés aux catégories créées lors de l’étape précédente en fournissant une description

pour chaque élément et des fichiers (un pour chaque élément de programme) au format « FEF ».

3- Création des éléments de documentation

L'auteur a développé des documents à fournir aux étudiants pendant ce TP, en utilisant Opale :

- description générale du système (magasin + traducteur) : structure, contexte d'utilisation, cas d'utilisation ;
- description de la partie opérative du traducteur: description de chaque capteur et actionneur ;
- description de la partie opérative du magasin : description de chaque capteur et actionneur ;
- description de la partie commande : description de l'API, du câblage ;
- notice explicative concernant le principe de codage des touches du clavier ;
- liste des 6 modules d'entrées/sorties installés sur l'API Magasin ;
- liste des XX modules d'entrées/sorties installés sur l'API Traducteur ;
- description du module d'entrées/sorties TOR TSX DMZ 28DR ;
- description du module d'acquisition TOR TSX DEZ 12D2 ;
- description du module d'acquisition analogique TSX AEZ 801 ;
- description du module de sorties analogiques TSX ASZ 401 ;
- description de l'algorithme d'initialisation du magasin ;
- description de l'algorithme d'initialisation du traducteur ;
- description du Grafset Maître ;
- description de l'algorithme de P.O.M. du magasin ;

- description de l'algorithme de P.O.M. du translateur ;
- description de l'algorithme de mise à jour des données (initialisation du tableau de données « quelle pièce dans quelle nacelle ») du magasin ;
- description de l'algorithme de mise à jour des données (initialisation du tableau de données « quelle pièce à quel emplacement ») du translateur ;
- description du fonctionnement normal attendu du système (magasin + translateur).

Lors de la création (insertion) dans le configurateur, l'auteur a associé à chaque élément de documentation, une ou plusieurs catégories créées lors de la première étape et y a ajouté, pour chacun, une description. Le fichier source, associé à chaque élément est au format XML (il s'agit généralement d'une arborescence de fichiers de types différents, PNG/JPG/etc., décrite dans un fichier XML). Le fait de créer autant de ressources avec une granularité fine facilite leur réutilisation.

4- Création des éléments de scénarios

L'auteur a développé, avec Opale, un scénario pédagogique pour chacune des deux sessions de TP.

L'objectif du premier scénario « *Scénario-Magasin 1* » est de guider les étudiants pour réaliser l'algorithme du chargement/déchargement manuel des pièces métalliques (voir Annexe C-3).

L'objectif du deuxième scénario « *Scénario-Magasin-Translateur 1* » est de guider les apprenants pour synchroniser le magasin vertical avec le translateur qui réalisera le chargement/déchargement automatique des pièces métalliques (voir Annexe C-4).

Les éléments de scénario insérés dans le configurateur sont associés aux catégories créées dans la première étape. Comme les documentations générées par Opale, les fichiers source de scénarios téléchargés sur le configurateur sont une arborescence de fichiers de types différents décrite dans un fichier XML.

5- Création des éléments de composant

Après avoir créé (inséré) les ressources dans le système du configurateur, l'auteur a créé un composant (configuration générique) pour chaque session de TP. Le premier composant « *Comp-Magasin-Vertical 1* », associé aux catégories « *Magasin rotatif* » et « *Système séquentiel* » cf. §4.4.1.3 - 1, regroupe toutes les ressources à exploiter dans la première session de TP. Il est composé des ressources :

- un élément de programme « *Programme-Base-Magasin 1* » ;
- un élément de scénario « *Scénario-Magasin 1* » ;
- les éléments de documentations créés dans la troisième étape.

Pour la deuxième session, l'auteur a profité du composant créé pour la première session qui regroupe toutes les documentations liées au système ainsi que l'élément de programme de base du magasin vertical. Il a seulement créé un deuxième composant « *Comp-Magasin-Vertical-Translateur 1* », associé aux catégories « *Magasin rotatif* » « *Translateur* » et « *Système séquentiel* » cf. §4.4.1.3 -1. Il comporte :

- un composant « *Magasin-Vertical 1* », mais en supprimant après importation le lien vers le scénario pédagogique associé à ce composant ;
- un élément de programme « *Programme-Translateur 1* » ;
- un élément de scénario « *Scénario-Magasin-Translateur 1* ».

Ces deux composants représentent deux configurations génériques, exploitables dans une session de TP après compilation des ressources associées à chacun. Rappelons que ces deux composants ne contiennent que des références vers des ressources (adresses des éléments de programmes, documentations et scénarios). La modification de leur structure n'implique que d'éditer ces références (supprimer ou modifier l'adresse d'élément faisant partie de la structure du composant).

Les fichiers sources, que l'auteur a insérés dans le configurateur en tant que ressources, sont déjà préparés par l'auteur. L'insertion de ces éléments dans le configurateur (éléments de programmes, documentations et scénarios) a requis **une heure**. La création du premier composant a requis **5 minutes**, la création du deuxième n'a requis que **2 minutes** du fait de la

réutilisation directe de l'élément de programme et des pièces de documentation associés au premier composant.

4.4.1.4 Génération d'une configuration

Dans cette phase, l'auteur a compilé les configurations des deux sessions de TP sur le magasin vertical. Pour la première session et via la page de création de configuration (du configurateur), l'auteur a sélectionné la catégorie « *Magasin rotatif* » (cf. §4.4.1.3), pour filtrer les contenus de la bibliothèque de composants. Dans la liste des composants affichés, il a choisi le composant « *Comp-Magasin-Vertical 1* », puis téléchargé le paquetage de tous les fichiers sources associés à ce composant (un fichier ZIP).

La compilation des fichiers a suivi les étapes suivantes :

- 1- Compiler le programme :** importation dans PL7Pro et génération d'une version exécutable sur API.
- 2- Compiler les documentations :** assemblage par l'auteur de tous les fichiers de documentation dans Opale, puis génération d'une version diffusable (au format *HTML*).
- 3- Compiler le scénario pédagogique :** ouverture du fichier de scénarios avec Opale, édition du contenu pour l'adapter au contexte du TP, puis génération d'une version diffusable (au format *HTML*). La génération automatique n'a pas été programmée dans le cadre de ces expérimentations faute de temps mais il serait intéressant dans un produit fini que cette génération soit déclenchée automatiquement.

Après avoir compilé toutes les ressources, l'auteur les a remis dans le fichier *ZIP* en le téléchargeant dans le configurateur, qui a enregistré ce paquetage (*ZIP*) dans sa bibliothèque de configurations. Le configurateur s'est chargé de récupérer tous les fichiers compilés dans ce paquetage et les a insérés dans la bibliothèque des ressources pédagogiques.

La structure de la configuration « *Config-Magasin-Vertical 1* » insérée dans le configurateur est similaire à celle du composant original (qui contient des ressources de type : programme, documentation et scénario). Cette configuration est automatiquement associée aux mêmes catégories que le composant original (« *Magasin rotatif* » et « *Système séquentiel* »).

Pour la configuration de la **deuxième session**, l'auteur a choisi dans la page de création de configuration la catégorie « *Translateur* » pour filtrer les composants enregistrés. Dans la liste des composants affichés, il a choisi le composant « *Comp-Magasin-Vertical-Translateur 1* », puis téléchargé le paquetage de tous les fichiers sources associés à ce composant. Pour générer cette configuration, l'auteur a suivi la même chronologie que pour la première, en compilant tous les fichiers du paquetage téléchargé y compris la recompilation du programme « *Programme-Base-Magasin 1* ». Cette configuration est automatiquement associée aux mêmes catégories que le composant original (« *Magasin rotatif* » « *Translateur* » et « *Système séquentiel* »).

Le temps requis pour générer la première configuration (compiler des fichiers sources, les remettre dans un fichier ZIP, télécharger sur le configurateur et insérer la description de cette configuration) a requis **15 minutes**. La réutilisation des ressources de la première configuration a réduit le temps de génération de la deuxième configuration à **8 minutes**.

Ce timing est améliorable avec une meilleure intégration entre les logiciels de développement (notamment Opale) et le configurateur.

Les configurations sont alors prêtes à être utilisées par un instructeur lors d'une session de TP. À ce stade, l'auteur n'intervient plus et la configuration de la session d'e-TP sera prête à l'utilisation.

4.4.1.5 Installation et utilisation d'une configuration

Avant de lancer la première session de TP, l'instructeur repère la configuration désirée dans le configurateur, selon le mot clé « Magasin rotatif ». Puis il télécharge le paquetage « Config-Magasin-Vertical 1 » (fichier ZIP) contenant tous les fichiers ressources compilés.

Après décompression des fichiers, l'instructeur charge le programme API dans le logiciel PL7Pro, puis télécharge ce programme sur l'automate du magasin. Il télécharge manuellement les documentations et le scénario pédagogique sur Moodle (la version récente du configurateur lors de cette expérimentation n'a pas permis d'effectuer un téléchargement automatique des programmes API ni des documentations et scénarios, l'instructeur a dû faire ces opérations manuellement). À ce stade, la première session de TP sur le magasin vertical est configurée, et les étudiants peuvent commencer à réaliser les activités de cette session selon le scénario pédagogique proposé.

Pour la **deuxième session**, l'instructeur repère une configuration selon le mot clé « Translateur ». Puis, il récupère le paquetage de configuration « Config-Magasin-Vertical-Translateur 1 », filtré selon cette sélection de catégorie (cette configuration est aussi affichable selon le mot clé « Magasin rotatif » comme elle est associée à une catégorie portant le même nom). Puis, il suit la même procédure d'installation que pour la première session.

Pour l'**instructeur**, la procédure suivie pour configurer les deux sessions de TP était la même, la seule différence étant les mots clés utilisés pour chercher les configurations. Les contenus différents de configurations des deux sessions n'impliquent aucune différence sur la façon d'utiliser le configurateur lui-même. À partir du moment où la session est installée, le configurateur n'a plus d'utilité ; l'instructeur supervise directement ou via le LMS les activités proposées dans le scénario pédagogiques de chaque session.

4.4.1.6 Gains et limites

Le développement des différentes ressources (programmes, documents et scénarios) est un travail à réaliser que l'auteur utilise ou non le configurateur. Nous n'avons donc mesuré que le temps de préparation de ces ressources au sein du configurateur. Ce temps de préparation, pour le magasin vertical et le translateur, est monté à une heure. Cette durée est plus longue qu'une solution simple de copier/coller que l'auteur peut adopter quand il travaille sans envisager la réutilisabilité de ses ressources par d'autres collègues.

Pour les configurations, la création du premier composant (configuration générique) a requis **5 minutes** et le gain s'est illustré lors de la création du deuxième qui a requis **2 minutes**. À propos des configurations compilées, la première a été créée au bout de 8 minutes alors que la deuxième n'a requis que 10 minutes. Le temps de créer une configuration est plus élevé que celui de créer un composant. Cette différence (pour le bénéfice du composant) s'est expliquée par la nécessité d'utiliser des logiciels extérieurs (Opale et Unity Pro) pour compiler les ressources.

Par contre, l'expérimentation a révélé que son utilisation limitée à quelques cas n'encourage pas l'auteur/instructeur à profiter du configurateur, compte tenu du temps requis pour créer les différents éléments de ressources. Selon les personnes interrogées, la solution de copier/coller est plus pratique malgré le fait que le configurateur puisse représenter un système d'archivage intéressant pour d'autres collègues qui travaillent également sur le même système (dans d'autres départements ou pendant d'autres trimestres). L'absence d'un

prototype de la fonction installateur du configurateur ne nous a pas permis d'évaluer s'il y avait un gain de temps dans la phase de configuration du TP.

4.4.2 Cas du « Transfert libre »

Nous avons présenté dans le deuxième chapitre le transfert libre (cf. §2.6.2). Ce SAP, à caractère pédagogique, est utilisé par les établissements partenaires de l'AIP Primeca RAO afin de réaliser des travaux pratiques en :

- automatique industrielle (programmation d'API, de robots industriels) ;
- informatique industrielle (vision industrielle, MES) ;
- gestion de production (ordonnancement, traçabilité).

Nous avons choisi ce système comme terrain d'expérimentation afin de prouver l'applicabilité de notre configurateur sur un système complexe et de mesurer la valeur ajoutée qu'il fournit au niveau de la réalisation de sessions et de leur configuration.

4.4.2.1 Description du transfert libre

Ce SAP, à vocation pédagogique, produit des assemblages de pièces de couleur sur un support métallique. Chaque support est convoyé par une palette qui avance par friction sur des bandes rugueuses en perpétuel mouvement. Chaque palette est équipée d'une étiquette électronique, autrement dit une mémoire non volatile stockant son numéro d'identification et d'autres informations nécessaires à la production.

Ce SAP est constitué de 6 postes dont deux robotisés, placés autour d'un convoyeur en boucle (voir Figure 4.18). Selon les besoins pédagogiques, ce SAP peut être utilisé soit poste par poste de manière indépendante, soit globalement (tous les postes sont utilisés pour la production), soit partiellement (certains postes coopèrent tandis que d'autres sont autonomes). Dans le cas des projets MES délivrés aux apprenants des départements Génie Électrique et Génie Industriel de l'INSA de Lyon, où un MES supervise le SAP, les postes ont les rôles suivants :

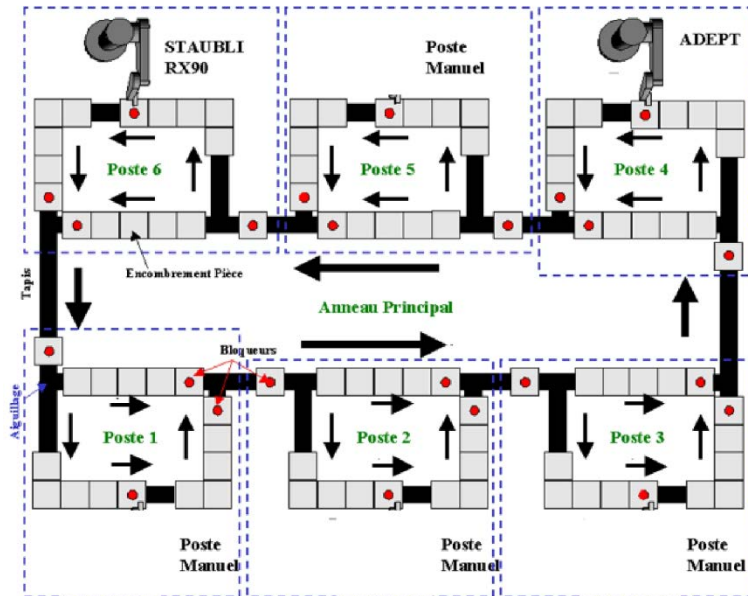


Figure 4.18 : Représentation schématique du système de transfert libre.

- Poste 1 : poste de lancement et de réception des palettes pour production (palettes impaires) ;
- Poste 2 : poste de lancement et de réception des palettes pour production (palettes paires) ;
- Poste 3 : poste de secours en cas d'indisponibilité d'un des postes 4 ou 6 ;
- Poste 4 : poste d'assemblage robotisé (SCARA Adept One) pour poser ou retirer des pièces noires (voir Figure 4.19 (A)) ;
- Poste 6 : poste d'assemblage robotisé (Staubli RX90) pour poser ou retirer des pièces rouges et vertes (voir Figure 4.19 (B)) ;
- Poste 7 : poste de supervision des précédents postes :
 - gestion des énergies (moteurs des convoyeurs, alimentation pneumatique de la ligne) ;
 - gestion de l'arrêt d'urgence de la ligne ;
 - gestion des modes de fonctionnement.

Le poste 5 est un poste de contrôle automatisé par vision industrielle (In-Sight 1000) qui n'est pas utilisé dans le cadre de ce projet

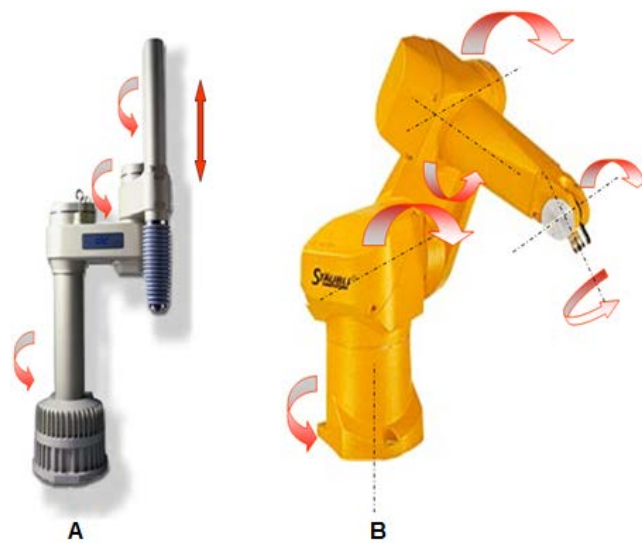


Figure 4.19 : (A) Le robot ADEPT (poste 4), (B) Le robot STAUBLI (poste 6).

Chaque poste dispose d'une zone d'entrée placée sur le convoyeur central, d'une zone de travail déportée du convoyeur central et d'une zone de sortie permettant aux palettes de transiter vers le poste suivant dans la boucle (voir Figure 4.20). Dans la zone d'entrée, une butée arrête les palettes à leur arrivée afin de lire leur étiquette numérique. Selon son avancement dans la production, la palette bloquée est libérée et est soit envoyée vers le poste de travail, soit laissée sur le convoyeur. Dans la zone de travail, la palette est immobilisée afin de réaliser l'opération du poste (ajout d'une pièce par exemple) puis relâchée. La zone de sortie a pour unique objectif d'éviter toute collision entre les palettes sortant du poste et celles transitant sur la boucle centrale du convoyeur.

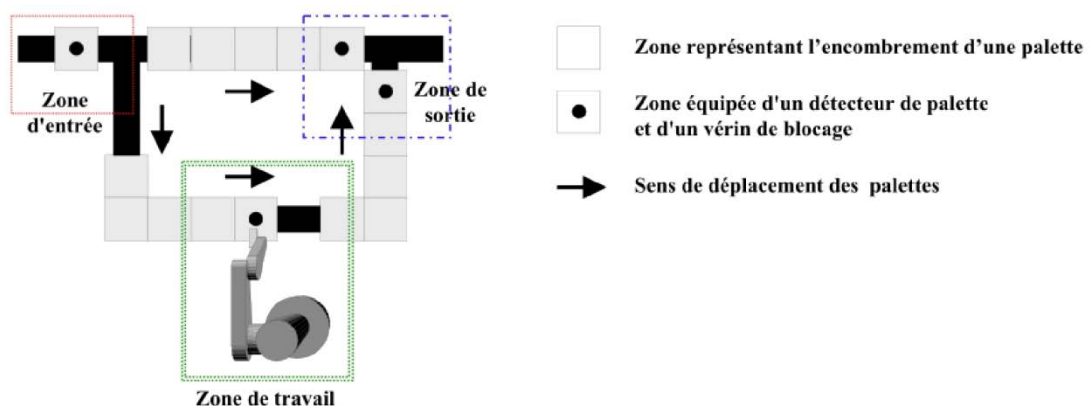


Figure 4.20 : Schéma d'un poste de travail.

4.4.2.2 Objectifs pédagogiques du projet MES et son déroulement

Le projet MES a pour objectif de former les apprenants à concevoir et réaliser le prototype d'un MES (le système d'information de l'atelier) collectant les données de production de tout ou partie du système de production, analysant la performance, contrôlant la production et supervisant le système. Dans notre cas, le système de production (le transfert libre) est opérationnel et la mise en place du MES nécessite, après étude de ce système, de modifier les programmes API afin que le MES puisse échanger les informations avec cette commande.

Le projet est réalisé par groupes de cinq ou six apprenants, chaque groupe réalise indépendamment une application MES répondant à l'ensemble du cahier des charges sur trois points :

- 1- la supervision du système ;
- 2- le contrôle de la production ;
- 3- l'analyse de la performance.

Le cahier des charges définit les besoins du client et correspond à une vision externe du système à concevoir. Il est principalement constitué de cas d'utilisation ainsi que de récapitulatifs sur les données à mettre en place. Les cas d'utilisation sont conçus pour la mise en œuvre sur l'atelier de l'AIP. Le système à concevoir n'est pas seulement un MES, mais il faut aussi mettre à jour les programmes automate de l'atelier pour que le MES puisse s'y greffer.

L'expérimentation s'articule autour d'une démonstration du fonctionnement globale du système. L'objectif de la séance de démo est de familiariser les étudiants avec le système pour qu'ils puissent dans les séances suivantes étudier l'existant, modéliser l'application et la faire fonctionner. Les ressources à utiliser dans cette expérimentation pour configurer le système du transfert libre sont réutilisables pour d'autres TP (par exemple : cf. §4.4.3). Cela nous aidera à démontrer la valeur ajoutée de notre configurateur en ce qui concerne Cette disponibilité, ce partage des ressources, ces possibilités de réutilisation et cette réduction du temps de configuration qui en résulte, sont des vraies valeurs ajoutées de notre configurateur.

4.4.2.3 Scénario pédagogique témoin: production autonome d'assemblages

Le SAP est opérationnel, et son utilisation (démarrage/arrêt du SAP, lancement/réception de production) a lieu à travers l'IHM du pupitre de supervision, sans l'aide d'un MES. Dans ce

scénario pédagogique, l'instructeur déroule la démonstration et les apprenants se familiarisent avec le système en observant son fonctionnement global.

4.4.2.3.1 Besoins pour le déroulement d'une session de projet

Le concepteur (auteur) de cette session de TP doit préparer en amont les ressources suivantes :

- les programmes API : une série de programmes (un par poste) de démonstration du fonctionnement du système complet ;
- les programmes des robots ;
- les documents fournis aux apprenants :
 - une description du système global ;
 - une description détaillée de l'aspect matériel/logiciel ;
 - le mode d'emploi du SAP via l'IHM (les apprenants auront à l'utiliser pour lancer leurs propres campagnes de mesure) ;
 - le cahier des charges s à remplir (fonctions qu'ils devront développer) ;
- les documents fournis à l'instructeur : une notice sur l'emplacement (sur le réseau informatique) des programmes API, IHM et robots.

Ce SAP étant utilisé pour différents enseignements (quatre TP⁶² se déroulent sur le système du transfert libre), pour différents publics et par différents enseignants (environ 11 enseignants chaque année), des programmes API existent déjà mais malgré un effort d'harmonisation, ils sont très hétérogènes et dispersés. Etant donnée la complexité du système, à la création d'un tel enseignement, l'automaticien n'a d'autre choix que de partir de programmes existants et de les modifier selon ses besoins.

⁶² Les TP qui se déroulent sur le système du transfert libre sont : TP **d'Automatique** « Pilotage d'automate programmable à l'aide de l'environnement de développement Unity PRO (Schneider) », TP **Robotique** « Mise en œuvre d'un cycle de travail robotisé avec synchronisation avec l'automate du poste du transfert adjacent », TP **Ordonnancement** « Essais de différents algorithmes d'ordonnancement des palettes sur l'ensemble du transfert libre », TP **MES** « La ligne de transfert fonctionne dans son ensemble (7 API) et simule un atelier flexible (fonctionnement en boucle). Les apprenants développent un projet de contrôle et de suivi de fabrication en communiquant avec les automates depuis un logiciel de MES ».

Par chance, dans ce cas précis, les programmes ont été organisés de manière modulaire (programmation par composants d'automatismes, sections dédiées à des fonctions indépendantes, ...), les modifications sont donc facilitées mais ce n'est pas toujours le cas. Lorsque des modifications ont lieu ultérieurement sur le SAP (changement de capteur, débogages, ...), il faut modifier tous les programmes liés au système, analyser chaque programme pour déterminer les impacts qu'engendre chaque modification d'une partie du code. C'est un travail fastidieux qui risque de se répéter régulièrement car un SAP évolue régulièrement (du fait de pannes ou simplement de besoins nouveaux à remplir). De plus, tous les documents explicatifs doivent être mis à jour simultanément. En bref, la réutilisation de ces programmes est très difficile à mettre en œuvre, encore plus à faire vivre dans le temps.

Nous distinguons trois étapes pour la réalisation de cette session de TP :

1. la conception/réalisation des programmes API, textes de TP, corrigés, ... ;
2. la préparation du TP avant la séance ;
3. le déroulement de la séance.

4.4.2.3.2 Conception/réalisation des ressources

Il faut :

- **(re)concevoir et réaliser les programmes API** nécessaires pour le fonctionnement des différents postes du système (heureusement, les programmes des robots existent déjà). Cela implique qu'un automaticien intervienne dans ce processus si l'auteur n'est pas lui-même spécialiste en automatisme ;
- **préparer les documentations** pour les apprenants et l'instructeur de la session de TP. Ici, la documentation étant fortement liée à l'objectif pédagogique du TP, l'auteur est nécessairement spécialiste du sujet (automatisme, gestion de production, ...). Un automaticien peut si besoin également intervenir ;
- **préparer le scénario de démonstration** du fonctionnement du système.

L'effort clef est la conception/réalisation des programmes API, avec l'aide d'un automaticien en tenant compte des objectifs pédagogiques spécifiques du TP.

Le premier inconvénient de cette façon de faire est que la configuration élaborée (l'ensemble des programmes, documents et scénarios pédagogiques) n'est pas utilisable sur des systèmes similaires (similaires en terme de fonctionnalités) alors que l'objectif pédagogique n'est pas spécifique à ce SAP, seuls les détails d'implémentation le sont.

4.4.2.3.3 La préparation du TP

Le SAP étant utilisé par plusieurs établissements partenaires, les TPs tournent à tour de rôle et chaque instructeur doit préparer le SAP pour son propre TP avant la séance. Ce travail peut prendre jusqu'à 15 minutes en fonction de l'état dans lequel il trouve le SAP à son arrivée. Il doit alors télécharger les programmes API et robots, préparer le SAP (remettre les palettes dans une configuration propre).

L'étape de préparation nécessite d'avoir un minimum de connaissances du SAP. C'est faisable la plupart du temps par l'instructeur (d'autant plus s'il est automaticien) mais dans certains cas, par exemple quand l'instructeur n'est pas du tout à l'aise avec le SAP, la tâche de préparation incombe au responsable de la plate-forme. Cela implique donc de le faire tôt le matin même, tard la veille au soir, ou pendant la pause méridienne, ce qui implique parfois de devoir prendre en charge des heures supplémentaires.

4.4.2.4 Utilisation du configurateur lors de la phase de préparation

Les étapes suivies par l'auteur sont :

1- Création des catégories décrivant les ressources à construire :

La première opération consiste à créer des catégories auxquelles les ressources (programmes, documentations et scénarios pédagogiques) insérées plus tard seront rattachées (cf. §4.3.1). Pour ce TP, l'auteur a créé les catégories suivantes :

- MES : pour référencer les ressources pouvant être utilisées dans un contexte de TP en MES ;
- Transfert libre : pour référencer les ressources pouvant être utilisées dans un contexte de TP sur le système du transfert libre ;
- Poste du transfert libre : pour référencer les ressources qui concernent un poste dans un système de transfert libre.

2- Création des éléments de programmes

Cinq programmes source ont été créés par l'auteur, chacun pour l'un des postes 1, 2, 4, 6 et 7 (« *Programme-poste n* » où *n* est : 1, 2, 4, 6, 7). Lors de la création (insertion) des éléments de programmes, l'auteur les a associés aux catégories qu'il a créées dans l'étape précédente. Les fichiers téléchargés sur le configurateur (un fichier de programme sources pour chaque élément de programme) sont au format Unity « *.sta ».

3- Création des éléments de documentation

L'auteur a développé des pièces de documentations à fournir aux étudiants pendant ce TP, en utilisant Opale. Lors de la création (insertion) des éléments de documentation dans le configurateur, l'auteur a associé chaque élément créé aux catégories créées dans la première étape. Le fichier source, associé à chaque élément, à télécharger sur le configurateur est au format « *.xml ». Les pièces de documentation créées sont :

- Architecture globale (6 postes + 1) ;
- Architecture générale de chaque poste (boucle principale et dérivation) ;
- Architecture détaillée de chaque poste (capteurs et actionneurs) ;
- Liste et emplacement des cartes E/S spécifique à chaque API ;
- Paramétrage des communications inter-automate ;
- Entrées et sorties spécifiques à chaque poste ;
- Notice boitiers de lecture écriture des étiquettes des palettes (L/E) ;
- Paramètres de configuration des boitiers L/E pour chaque poste ;
- Organisation des données stockées dans les étiquettes ;
- Description externe des DFB de gestion des vérins ;
- Fonctionnement des DFB de gestion des vérins⁶³ ;

63 Nous avons créé deux documents séparés sur la description et le fonctionnement des DFB de gestion des

- Description externe des DFB de L/E ;
- Fonctionnement interne des DFB de L/E ;
- Description externe des DFB de gestion entrée/poste/sortie ;
- Fonctionnement des DFB de gestion entrée/poste/sortie.

4- Création de l'élément de scénario

L'auteur a développé un scénario pédagogique simple « *Scénario-MES 1* », avec Opale. L'élément de scénario inséré dans le configurateur est associé aux catégories créées dans la première étape, et le fichier source téléchargé est un fichier ZIP contenant une arborescence de fichiers de types différents, PNG/JPG/etc., décrite dans un fichier XML.

5- Création des composants

Comme tous les éléments de ressources sont créés (insérés) dans le configurateur, l'auteur commence à créer un composant par poste (« *Comp-poste n* » où *n* est : 1, 2, 4, 6,7). Chacun représente une configuration éditable (générique) pour le poste concerné. Les cinq composants sont associés aux catégories créées dans la première étape. Chaque « *Comp-poste n* » créé a une structure simple :

- un élément de programme correspondant au poste (« *Programme-poste n* » ;
- des éléments de documentation de chaque poste :
 - architecture générale du poste « *x* » (boucle principale et dérivation) ;
 - architecture détaillée du poste « *x* » (capteurs et actionneurs) ;
 - liste et emplacement des cartes E/S spécifique à l'API « *x* » ;
 - paramétrage Global Data spécifique à l'API « *x* » ;
 - données d'E/S spécifiques au poste « *x* » ;
 - paramètres de dialogue avec boîtiers L/E pour poste « *x* ».

vérins pour pouvoir créer ultérieurement des TP où on demande aux apprenants de programmer ces DFB en ne leur donnant que les données d'E/S.

Chacun de ces composants représente en soi une configuration générique du poste concerné.

Enfin vient la configuration globale du système. L'auteur construit un composant « *Comp-MES-Démo* » constitué des éléments suivants :

- les cinq composants créés ci-dessus ;
- un élément de scénario (« *Scénario-MES 1* » créé dans l'étape 4) ;
- les éléments de documentation suivants :
 - architecture globale (6 postes + 1) ;
 - notice boitiers L/E ;
 - organisation générale données étiquettes ;
 - description externe des DFB de gestion des vérins ;
 - fonctionnement des DFB de gestion des vérins ;
 - description externe des DFB de L/E ;
 - fonctionnement des DFB de L/E ;
 - description externe des DFB de gestion entrée/poste/sortie ;
 - fonctionnement des DFB de gestion entrée/poste/sortie.

Le dernier composant créé par l'auteur représente la configuration générique de TP. L'étape suivante consiste à compiler les ressources de ce composant afin de générer une configuration spécifique exploitable dans un TP.

4.4.2.5 Utilisation du configurateur pour générer une configuration spécifique

Via la page de création de configuration (du configurateur), l'auteur a sélectionné les catégories « *MES, Transfert libre, Poste du transfert libre* », cf. §4.4.2.4 - 1, pour filtrer les contenus de la bibliothèque de composants selon cette sélection (lors de l'expérimentation, cette bibliothèque ne contient que les composants créés dans la phase précédente, et la sélection d'une seule de ces catégories conduisait à la même liste des composants affichés).

Dans la liste des composants affichés, il choisit le composant correspondant à ses objectifs (le dernier composant créé dans la dernière étape « *Comp-MES-Démo* »), puis télécharge le paquetage de tous les fichiers sources associés à ce composant, avec l'application « *Config_Launcher.exe* ».

La compilation des fichiers pour les rendre exploitables dans un TP est passée par les étapes suivantes :

- 1- Compilation des programmes :** ouverture puis compilation de tous les fichiers de programmes API avec Unity Pro ;
- 2- Compilation des documentations :** à l'aide de l'outil fourni par le configurateur « *Config_Launcher.exe* », l'auteur assemble tous les fichiers de documentation en un seul qu'il ouvre avec Opale, avant de générer une des versions *HTML* et *PDF* de documentation ;
- 3- Compilation du scénario pédagogique :** même procédure que pour l'item précédent.

Après avoir compilé toutes les ressources, l'auteur les remise sous forme de fichier *ZIP* dans le configurateur, qui enregistre ce paquetage dans sa bibliothèque de configurations en tant que nouvelle configuration « *Config-MES-Démo 1* ». Le configurateur, en insérant tous ces fichiers compilés dans la bibliothèque des ressources pédagogiques, leur associe automatiquement les catégories créées à la première étape (cf. §4.4.2.4 - 1). À ce stade, l'auteur n'intervient plus et la configuration de la session d'e-TP est prête à l'utilisation par l'instructeur.

4.4.2.6 Utilisation du configurateur lors de la phase d'exploitation

Avant de lancer le TP, l'instructeur cherche la configuration désirée sur le configurateur, selon les mots « *MES* » et « *Démo* ». La liste des configurations générée par le moteur de recherche affiche la configuration « *Config-MES-Démo 1* » (qui correspond aux mots de recherche), et l'instructeur télécharge le paquetage de cette configuration qui contient toutes les ressources (compilées par l'auteur à l'étape précédente) nécessaires, accompagnées de l'application (*Config_Launcher.exe*) qui sera nécessaire pour télécharger les programmes API sur les automates et les documentations et scénario pédagogique sur LMS.

À l'aide de cette application, l'instructeur lance les programmes de postes (1, 2, 4, 6 et 7) chacun dans une instance du logiciel Unity Pro, et télécharge manuellement ces programmes

sur les API (pour les mêmes raisons explicitées dans §4.4.1.5). Il télécharge également les documentations et le scénario pédagogique sur le LMS. À ce stade, la session de TP sur le système du transfert libre est configurée, et les étudiants commencent à élaborer les activités de cette session selon le scénario pédagogique proposé.

4.4.2.7 Gains et limites

Avant de créer des éléments dans le configurateur, l'auteur possède tous les fichiers sources nécessaires pour cette expérimentation. Leur développement n'est donc pas spécifique et n'est pas pris en compte. Dans ces conditions, le temps total pour l'insertion des ressources dans le configurateur a duré **une heure**. L'auteur concerné a trouvé cette opération un peu longue.

La création des composants « *Comp-poste n* » a requis 20 minutes (soit 4 minutes par composant). Pour l'auteur, la première réutilisation de ces composants pour construire le composant « *Comp-MES-Démo* » s'est avérée intéressante (au bout de 2 minutes). La simplicité d'assemblage des ressources, ainsi que la construction de composant assemblant des sous-composants, encourage à développer des configurations correspondant à d'autres scénarios pédagogiques, en suivant la même procédure. La génération de la configuration « *Config-MES-Démo 1* » a pris **10 minutes** : l'auteur a signalé que ce temps était long comparé au temps nécessaire pour la création du composant.

L'auteur a apprécié l'application d'assemblage automatique des ressources pédagogiques (documentations et scénario dans la phase de génération de configuration compilée), mais il a constaté également qu'il lui faut du temps pour maîtriser Opale et que cela pourra être un souci pour les futurs utilisateurs du configurateur s'ils ne sont pas déjà formés à cet outil pédagogique. Notons que cette application se déploie progressivement dans les universités, ce qui devrait à l'avenir ne plus être un frein. L'auteur a aussi constaté que le fait de compiler les ressources via des logiciels externes (Opale et Unity Pro) est contraignant. Par contre, il a apprécié la fonctionnalité de recherche. Le gain ressenti par l'auteur se traduit par la disponibilité des ressources archivées et catégorisées accessibles via une application web, la logique d'assembler ses ressources sous un composant à l'aide d'un outil d'assemblage automatique et le lancement automatique des programmes API dans Unity (malgré que cette solution ne représente pas un téléchargement automatique réel des programmes API vers les automates). En dehors de l'insertion des ressources au sein du configurateur et l'utilisation des logiciels externes pour éditer et compiler les ressources d'une configuration, l'auteur a

constaté un gain du temps pour la génération de configuration et a estimé que le configurateur pourra considérablement réduire le temps de configurer un TP avec un contenu plus riche de ses bibliothèques de ressources pédagogiques et de configuration.

4.4.3 Réutilisation de configuration du « Transfert libre » dans un autre contexte que le projet MES : TP S.A.P. en GMC

Dans cette section, nous présentons une expérimentation sur le système du transfert libre de l'atelier AIP pour un TP dans le département GMC (Génie Mécanique Construction) à l'INSA de Lyon. Cette expérimentation a pour but d'illustrer la possibilité de reconfigurer une partie d'un SAP, via notre configurateur, tout en favorisant la réutilisation des ressources enregistrées dans les bibliothèques de notre système. L'objectif du travail proposé est de mettre en application les concepts vus dans le cours *Systemes Automatisés de Production* en 3^{ème} année GMC et lors du projet effectué en TD. Ce TP se déroule à travers trois séances, pendant lesquelles les apprenants réalisent une partie du travail demandé dans le cahier des charges. La description complète de ce TP est présentée dans l'**annexe D**.

4.4.3.1 Scénario pédagogique : gestion d'un poste de travail

Dans ce scénario pédagogique, le travail à réaliser consiste à mettre en place, à partir d'un cahier des charges, la commande d'un automate pour la gestion d'un poste de travail du transfert libre. L'étude est divisée en 6 étapes :

- stockage amont ;
- entrée du poste ;
- travail effectué sur le poste ;
- sortie du poste ;
- stockage aval ;
- gestion de la sécurité.

La partie concernant la gestion de la zone d'entrée est partiellement donnée, afin de faciliter une prise en main rapide du logiciel Unity Pro.

Dans le cadre de cette expérimentation, l'auteur a choisi de configurer un poste de travail (le Poste 1) via notre configurateur afin que les étudiants mettent en œuvre la commande de la zone d'entrée sur l'automate. Notre objectif est de mettre en valeur la réutilisabilité des ressources enregistrées dans notre système, dans le cadre de scénario d'apprentissage différent de celui pour lequel ces ressources ont été développées.

4.4.3.2 Utilisation du configurateur dans la phase de préparation

Traçons l'activité de l'auteur lors de la phase de préparation :

- il est allé directement à la page de recherche en tapant les mots clés : « *poste du transfert libre* ». Cette phrase correspond à une catégorie créée lors de l'expérimentation précédente ;
- le moteur de recherche a proposé en réponse toutes les ressources (précédemment créées) correspondant à ce poste ;
- dans la liste, l'auteur a sélectionné le composant « *Comp-poste 1* » qui comporte un programme pour le *poste 1* (développé dans le cadre du projet MES), et les documents connexes. Il a laissé, logiquement, de côté le scénario pédagogique associé à ce composant ;
- il a ensuite téléchargé le fichier du programme afin de l'adapter au contexte de ce TP. Il a pu en réutiliser une grande partie, notamment les sections bas niveaux qui gèrent l'interface avec la partie opérative qui est la même ;
- il a créé un nouvel élément de programme « *Programme-poste-GMC 1* » dans le configurateur en y téléchargeant le nouveau fichier ;
- le scénario pédagogique pour ce TP était déjà un document PDF développé auparavant par l'auteur, celui-ci n'a pas eu à utiliser Opale pour la documentation. Il a donc créé un élément de scénario « *Scénario-Poste-1-GMC* » en téléchargeant ce document sur le configurateur. Les documentations associées au composant original correspondaient aux besoins de ce TP. Jusqu'ici, l'auteur a considéré que toutes les ressources dont il a besoin étaient disponibles sur le configurateur ;
- il est retourné à la page du composant « *Comp-poste 1* » pour l'éditer ;
- dans la page d'édition de composant, il a changé l'élément de programme associé à ce composant en le remplaçant par le nouvel élément de programme qu'il a créé. Enfin, l'auteur a enregistré ce composant sous une nouvelle version « *Comp-poste 1 (v : 2)* ». De cette manière, le composant original n'a pas été affecté par la modification réalisée, et la bibliothèque de composants dispose désormais de deux composants différents du point de vue de leur programme. À ce stade, ce composant représente une configuration générique à compiler plus tard ;

4.4.3.3 Utilisation du configurateur pour générer la configuration spécifique

Via la page de création de configuration (du configurateur), l'auteur :

- a sélectionné la catégorie « *Poste du transfert libre* » (cf. §4.4.2.4 4.4.2.4), pour filtrer les contenus de la bibliothèque de composants ;
- dans la liste des composants affichés, il a choisi le composant « *Comp-poste 1 (v : 2)* », puis a téléchargé le paquetage de tous les fichiers sources associés à ce composant, avec l'application « *Config_Launcher.exe* ». La compilation des fichiers pour les rendre exploitables est passée à travers les mêmes étapes que dans la section 4.4.2.4 ;
- après avoir compilé toutes les ressources, l'auteur les a remises dans le fichier *ZIP* sur le configurateur, qui a enregistré ce paquetage dans sa bibliothèque de configurations en tant que nouvelle configuration « *Config-poste 1 (v : 2)* ». À ce stade, l'auteur a fini d'intervenir : la configuration de la session d'e-TP est prête à l'utilisation par l'instructeur.

4.4.3.4 Utilisation du configurateur dans la phase d'exploitation

Avant de lancer le TP, l'instructeur (ici c'est la même personne que l'auteur) :

- est allé directement dans la page de bibliothèque de configurations en choisissant la catégorie « *Poste de transfert libre* » ;
- a récupéré, dans la liste des configurations filtrées sur cette catégorie, le paquetage de configuration « *Config-poste 1 (v : 2)* » qui contient toutes les ressources nécessaires, avec l'application (*Config_Launcher.exe*) (les documentations et scénario pédagogique n'ont pas été téléchargés sur un LMS lors de cette expérimentation, mais leur localisation dans un répertoire partagé a été fournie aux étudiants) ;
- a téléchargé manuellement (à l'aide de l'application *Config_Launcher*) puis démarré le programme du poste 1 depuis le logiciel Unity Pro vers l'API du poste 1 ;
- a diffusé l'adresse du répertoire partagé contenant les documentations et le scénario pédagogique aux étudiants via courriel. À ce stade, la session de TP sur le système

du transfert libre était configurée, et les étudiants ont pu commencer à dérouler leurs activités selon le scénario pédagogique proposé.

4.4.3.5 Évaluation

Pour ce TP, l'utilisation du configurateur fut plus rapide que dans les expérimentations précédentes. L'auteur a souligné l'intérêt de la fonctionnalité de recherche, qui lui a permis de prendre connaissance rapidement des contenus du configurateur susceptibles de l'intéresser. L'auteur a créé un élément de programme en éditant un programme déjà existant dans la bibliothèque des ressources pédagogiques du configurateur. Nous n'avons pas compté le temps d'édition de ce programme car son développement n'entre pas dans les tâches du configurateur. La création d'un élément de programme et d'un élément de scénario a requis **5 minutes**. À côté, l'édition du composant (en modifiant le programme associé et en ajoutant un scénario et en générant une nouvelle version) fut rapide (**1 minute**).

La compilation des ressources de ce composant pour générer une configuration a requis **10 minutes**. Cette valeur élevée est due au passage du configurateur à Unity afin de compiler le programme API. L'auteur n'ayant pas d'expérience avec Opale, nous avons dû le guider dans le processus de compilation des documentations pour générer des formats PDF et *HTML*. Il a trouvé que l'obligation d'utiliser Opale pour éditer les documentations téléchargées (éditables uniquement avec Opale) dans le paquetage de configuration était contraignante mais il a également trouvé que la mise à disposition de ces ressources documentaires sur un répertoire en ligne est intéressante.

A l'instar d'Opale, le fait de devoir commuter entre le configurateur et Unity pour compiler les ressources a été perçu comme un inconvénient.

4.4.4 Synthèse

Nous présentons ici un bilan des différents timings sur l'utilisation du configurateur en automation. Le Tableau 4.1 liste toutes les opérations effectuées sur les deux systèmes dans deux sessions, en détaillant le temps requis pour chacune.

	Magasin vertical		Transfert libre	
	Timing			
Opération	Session 1	Session 2	Session 1	Session 2
Insertion des ressources	1 heure		1 heure	
Edition des ressources	×	×	×	1 minute
Création de composant	5 minutes	2 minutes	22 minutes	1 minute
Génération de configurations	8 minutes	10 minutes	10 minutes	10 minutes
Installation de configurations	5 minutes	5 minutes	5 minutes	5 minutes
Timing total	78 minutes	77 minutes	97 minutes	77 minutes

Tableau 4.1 : Bilan des temps d'utilisation du prototype de configurateur.

Le temps total requis pour configurer chaque session sur les deux systèmes en utilisant le configurateur peut apparaître long. Mais nous argumentons ici que ce timing dans les conditions des expérimentations ne représente pas le temps réel pour configurer une session d'e-TP avec une version plus mature du configurateur. Dans les circonstances des expérimentations réalisées, nous comptons cinq opérations essentielles à effectuer par les utilisateurs dont deux sont problématiques en termes de temps de réalisation : l'insertion des ressources et la génération des configurations. En effet, dans une version mature du configurateur, l'utilisateur dispose d'une bibliothèque de ressources pédagogiques assez riche, le dispensant d'insérer de nouvelles ressources. Ici, dans la deuxième session sur le système de transfert libre, le seul élément réutilisé par l'auteur est l'élément « *Programme-poste 1* », créé lors de la première session par un autre auteur. Dans une version plus aboutie du configurateur, la génération de configuration se baserait essentiellement sur la compilation automatique de ressources au lieu d'utiliser des logiciels externes pour les créer. L'édition des ressources est une opération probable dans un contexte de préparation de TP que ce soit d'ailleurs sans ou avec notre configurateur. Logiquement, le temps est donc le même. Cette argumentation nous conduit à réévaluer le timing total sur l'utilisation du configurateur en omettant le temps requis pour chacune de ces deux opérations, comme le montre le Tableau 4.2.

	Magasin vertical		Transfert libre	
	Timing			
Opération	Session 1	Session 2	Session 1	Session 2
Edition des ressources	×	×	×	1 minute
Création de composant	5 minutes	2 minutes	22 minutes	1 minute
Installation de configurations	5 minutes	5 minutes	5 minutes	5 minutes
Timing total	10 minutes	7 minutes	27 minutes	7 minutes

Tableau 4.2 : Estimation de timing sur l'utilisation du configurateur d'une version plus mature.

Le temps total estimé pour configurer une session d'e-TP sur chaque système, avec une bibliothèque de ressources plus riche ainsi qu'une fonctionnalité de génération automatique intégrée au sein du configurateur, semble donc satisfaisant. Notons que pour la deuxième session sur les deux systèmes, la bibliothèque de configuration dispose des composants réutilisés, ce qui réduit le temps total pour configurer la session par rapport à la première session. Il faut également noter que chaque opération listée dans ce tableau est réalisée par un utilisateur non familiarisé avec le prototype du configurateur, ce qui nous encourage à présumer que le temps total pour configurer une session d'e-TP avec notre configurateur pourrait être inférieur à ceux affichés dans le tableau.

4.5 Application sur d'autres environnements de TP : en informatique

Afin de prouver l'applicabilité du configurateur dans d'autres domaines que l'automatique, nous avons mené une expérimentation dans le domaine de l'informatique. Cette expérimentation s'est déroulée à l'aide et en collaboration avec des collègues de la faculté d'informatique, département de Génie logiciel, à l'université de Tishreen en Syrie.

Durant les deux premières années, les apprenants en informatique à l'université de Tishreen apprennent des langages de programmation (Pascal, C++, Java) pour développer de petites applications. En troisième année, les apprenants sont formés au développement d'applications plus conséquentes et apprennent pour cela des méthodologies d'analyses des exigences et d'achèvement des projets. C'est dans ce cadre que s'est déroulée cette expérimentation. Elle a pour objectif de prouver que le configurateur peut être réutilisé dans d'autres disciplines que les automatismes.

4.5.1 Sujet du TP

Dans ce TP, les étudiants ont à développer une application fournissant des outils pour réaliser des dessins en deux dimensions à l'aide de formes basiques (lignes, flèches, cercles,

rectangles, polygones, etc.), et à réaliser des opérations géométriques standards sur les objets dessinés (rotation, translation, agrandissement, etc.).

Dans cet exemple, les étudiants, amenés à développer des opérations géométriques, sont invités à utiliser les techniques de transformation géométrique par calcul matriciel.

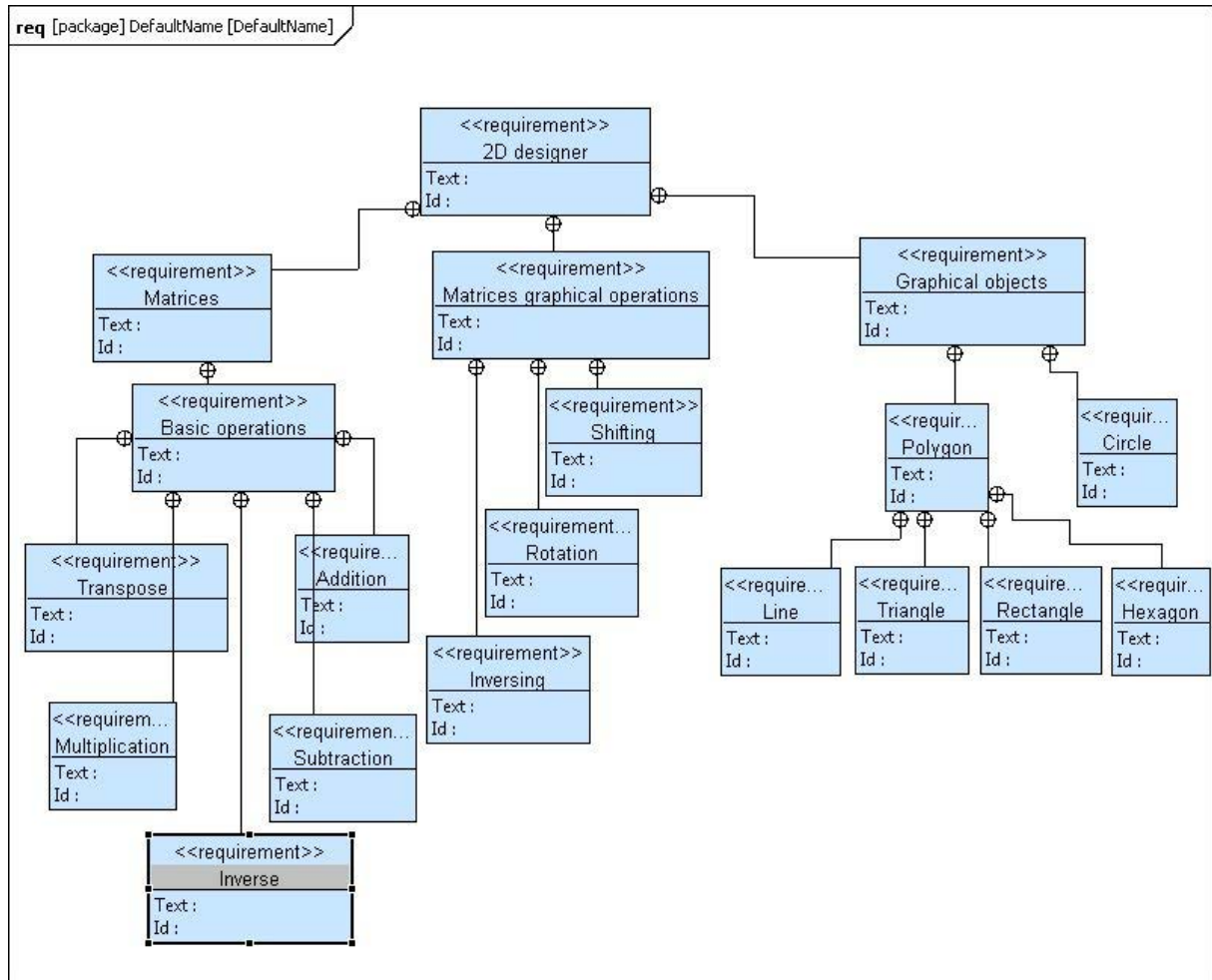


Figure 4.21 : Exigences pour l'application de dessin 2D.

L'auteur du TP a développé des diagrammes de classe, en UML, décrivant les classes à développer par les apprenants. À chaque session, une part de ces classes est transformée en code Java par les apprenants.

Deux sessions ont été configurées :

- 1- Session 1 : développer une classe réalisant des calculs sur des matrices : addition, soustraction, multiplication ;

- 2- Session 2 : développer une classe réalisant des opérations sur des formes géométriques en 2D : rotation, translation, etc.

4.5.2 Préparation des ressources pour la première session

L'auteur a développé un diagramme de classe représentant les opérations sur une matrice (voir Figure 4.22) à l'aide d'un logiciel de conception de diagrammes UML BlueJ⁶⁴, en préparant une documentation sur l'utilisation de ce logiciel. Un scénario pédagogique est développé par Opale et une illustration mathématique (sur les opérations matricielles préparée). Nous avons donc pour cette session les ressources suivantes :

- **diagramme de classe** représentant la classe à développer en Java par les apprenants ;
- **scénario pédagogique** à suivre par les apprenants pendant cette session ;
- **documentation** illustrant des opérations mathématiques sur les matrices ;

Toutes ces ressources sont prêtes et ont été insérées dans le configurateur en suivant la même approche que dans l'expérimentation précédente (cf. §4.3.1). Dans ce cas précis, comme le prototype est développé dès le début pour servir les TP en automation sans prendre en compte son extensibilité dans d'autres domaines scientifiques, et parce qu'un programme API est utilisé pour le fonctionnement de SAP dans un TP en automation, nous avons considéré qu'un diagramme de classe pourra correspondre à l'aspect fonctionnel d'un système dans un TP en informatique. Le fichier de diagramme de classe a été inséré dans le configurateur en tant qu'élément de programme. La nature des autres ressources ne diffère pas de celles insérées pendant l'expérimentation en automatique.

⁶⁴ <http://www.bluej.org/>

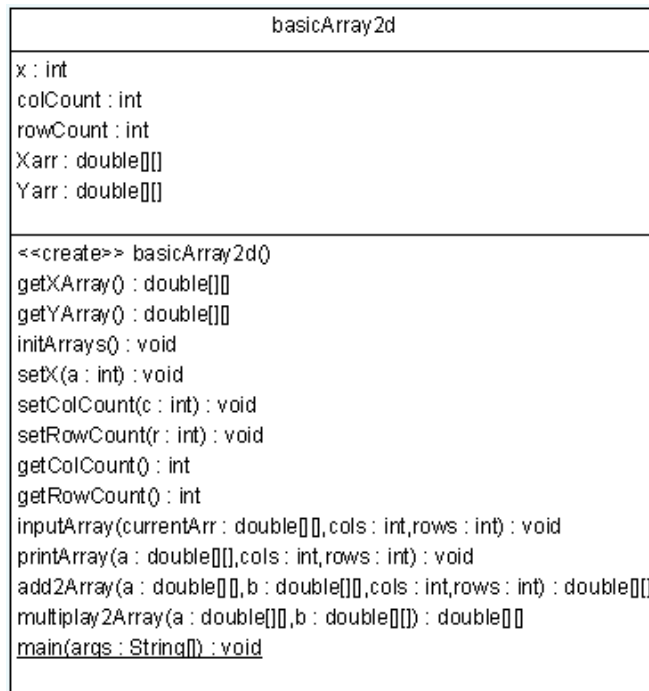


Figure 4.22 : Classe représentant les opérations sur les matrices (pour la 1ère session).

4.5.3 Génération de configuration et son exploitation dans la session de TP

Comme dans les expérimentations précédentes, nous avons d'abord construit un composant rassemblant toutes les ressources exploitées dans la session. Pour compiler les ressources, l'auteur télécharge le paquetage de ressources génériques et compile les pièces de documentations et le scénario pédagogique. Comme l'élément de programme dans ce paquetage est un diagramme UML, le logiciel ArgoUML⁶⁵ a été utilisé pour générer un squelette de code Java. Ces ressources compilées ont été rassemblées dans un fichier ZIP pour l'insérer dans le configurateur et générer un paquetage de configuration. En début de session, l'instructeur a téléchargé le paquetage de configuration et diffuse aux apprenants les fichiers de documentation, scénario pédagogique et le squelette de classe Java à développer.

4.5.4 Deuxième session

Dans cette session, l'auteur était une autre personne que celui de la première session. Nous avons fait ce choix d'une part pour tester l'utilisation du moteur de recherche du configurateur, et, d'autre part, pour permettre au nouvel auteur de réutiliser des ressources qu'il n'a pas développées lui-même.

⁶⁵ <http://argouml.tigris.org>

Cette session de TP est consacrée au développement d'une classe Java réalisant des opérations sur des matrices représentant des transformations géométriques en 2D (rotation, translation, etc.).

Cette classe exigeant des fonctionnalités de calcul matriciel (addition, soustraction, division, etc.), l'auteur a cherché dans le configurateur des ressources idoines. Il a ainsi trouvé un composant qui dispose d'un diagramme de classe (issu de la session précédente) modélisant les opérations basiques sur des matrices. Ce diagramme pouvait être réutilisé pour la deuxième session en l'étendant avec de nouvelles fonctionnalités (voir Figure 4.23). Les ressources documentaires du composant récupéré ont également été réutilisées en ajoutant des documentations sur la représentation mathématique des nouvelles opérations matricielles. Le scénario associé à ce composant a été remplacé par un autre consacré à cette deuxième session.

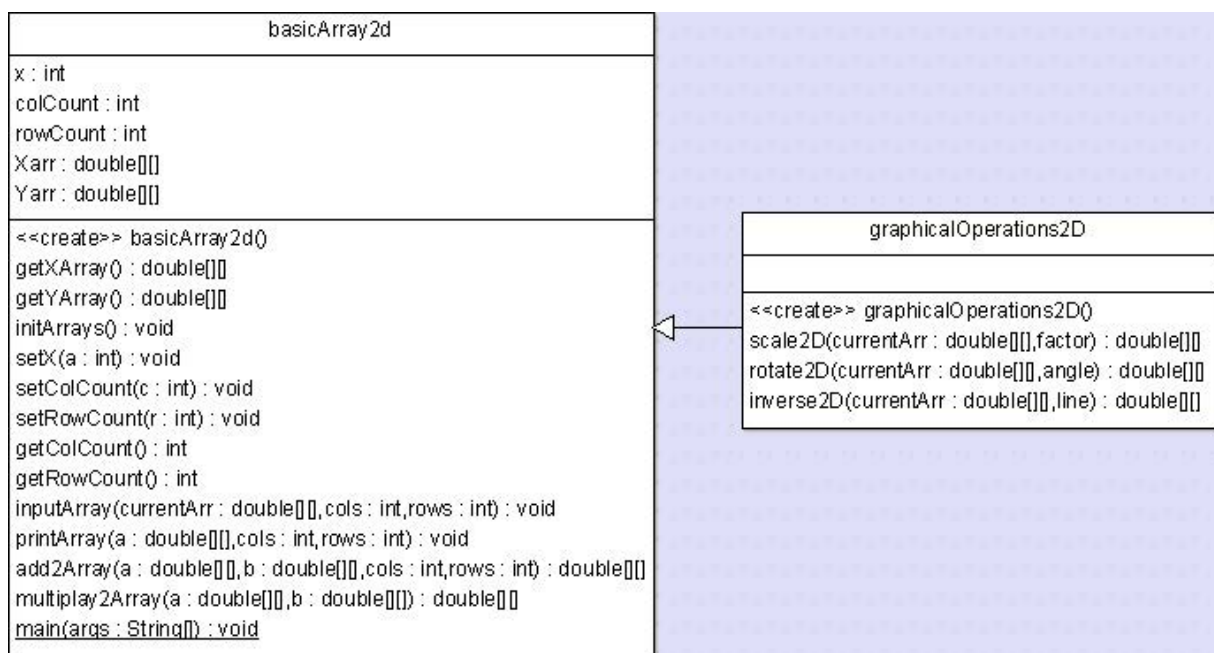


Figure 4.23 : Extension de classe des opérations basiques pour introduire des nouvelles fonctionnalités.

De cette manière, l'auteur a profité d'une configuration générique existante (celle de la première session) dans le configurateur pour construire un composant (configuration générique) pour la deuxième session. La tâche suivante a consisté à compiler les ressources du nouveau composant et générer le paquetage de configuration puis l'enregistrer dans la bibliothèque de configuration du configurateur.

L'instructeur de la deuxième session a récupéré le paquetage de configuration et diffusé le squelette du programme Java à développer (avec la classe complète des opérations, réutilisée

depuis le configurateur), les documentations et le scénario pédagogique à réaliser par les apprenants.

4.5.5 Évaluation des retours des utilisateurs

Nous avons évalué l'utilisation du configurateur selon trois facteurs :

- 1- **l'archivage des ressources** dans une bibliothèque interactive en ligne : comme en automatique, les utilisateurs du configurateur ont considéré (pour la première session) que la procédure d'insertion des ressources dans le système du configurateur était une opération un peu longue. Mais à partir de la deuxième session, la réutilisation des ressources déjà sauvegardées et la simplicité pour générer la configuration de cette session a encouragé ces utilisateurs à suivre cette procédure, en estimant que l'insertion permanente des ressources enrichira cette bibliothèque ;
- 2- **l'indexation des ressources** : les utilisateurs ont été satisfaits par la fonctionnalité de recherche fournie par le configurateur, mais ils estiment que la grande variété des sujets et thèmes pédagogiques, ainsi que des disciplines techniques, en informatique imposent d'améliorer les ontologies utilisées pour catégoriser et indexer les ressources dans le configurateur, surtout l'ontologie de domaine pédagogique. De plus, en automatisme on distingue le concept de « composant automate » (vérin pneumatique, bras robot, ...) qui représente une part d'un système automatisé de manière générale, mais ils n'utilisent pas de concept équivalent en informatique ;
- 3- **la réutilisabilité des ressources** : les utilisateurs ont apprécié l'interactivité du configurateur et la possibilité de consulter des ressources sauvegardées dans le but de les réutiliser. Le seul inconvénient concerne les éléments de programmes : comment se conformer à une forme dite générique d'un programme ?. Les auteurs ont opté pour le diagramme de classe comme modèle générique d'un programme, ce qui permet de générer un code source dans tout langage de programmation orienté objet, mais deux problèmes subsistent : d'une part, le format de fichiers générés par le logiciel « *BlueJ* » n'est pas compatible avec d'autres logiciels de conception UML (par exemple : Rational Rose, Artisan Studio), ce qui impose aux utilisateurs d'utiliser uniquement ce logiciel pour consulter ou éditer les diagrammes de classe archivés. D'autre part, au lieu d'adopter les diagrammes UML dans la phase générique du processus de gestion des programmes informatiques, le code source écrit dans un langage de programmation

standard (C++, Pascal, Java, etc.) peut convenir et éviter cette forte dépendance à un outil d'édition. Le revers de cette approche est que cet élément de ressource générique ne pourra être réutilisé que pour des configurations utilisant le même langage de programmation, ce qui limite sa réutilisabilité.

L'usage du configurateur dans le domaine de l'informatique a été accepté par les utilisateurs de ce système. Les remarques introduites ici ont attiré notre attention sur la partie du modèle du configurateur trop dépendant du domaine de l'automatique : l'élément **central** de toute configuration, le « *Programme* », représente un concept limité pour un TP en informatique, tandis que d'autres termes peuvent être plus représentatifs en fonction de discipline enseignée (classe, diagramme, algorithme, requêtes SQL, tables de base de données, protocole réseau, ...).

Cela ouvre une perspective pour redéfinir l'élément central d'une configuration, qui peut être un ensemble d'éléments centraux, et ses relations avec les autres éléments du modèle pour que ce dernier puisse encore mieux convenir dans d'autres domaines scientifiques.

4.6 Synthèse sur l'utilisation du configurateur

Cette section analyse les gains et déficits sur l'utilisation de notre configurateur, à la suite de ces expérimentations. Nous abordons dans un premier temps les **aspects techniques** liés à l'utilisation du configurateur par les auteurs/instructeurs concernés, puis nous analysons la **valeur ajoutée** par le configurateur ainsi que les limites et les aspects à améliorer.

4.6.1 Analyse technique

Cette analyse est organisée selon les quatre phases d'utilisation du configurateur :

1- la phase d'insertion des ressources : pour toutes les expérimentations menées, et comme la bibliothèque de ressources ainsi que de configurations étaient vierges, nous avons interrogé nos auteurs sur la procédure suivie pour l'insertion des ressources. Ils ont constaté qu'elle représente un travail supplémentaire qu'ils n'ont pas anticipé (définir des catégories représentatives avant insertion, ajouter une description pour chacune des ressources insérées indiquant aux autres auteurs l'intérêt et comment la réutiliser). Cette tâche les oblige, toujours selon les auteurs interrogés, à dédier un temps, variable selon le type et le rôle de la ressource. Ceci est à relativiser du fait que chacun des auteurs s'est déjà constitué ses archives personnelles de ressources (même si elles ne sont pas organisées selon le modèle présenté dans ce document) qu'il utilise

pour préparer ses TP. Nous avons interrogé les auteurs sur le bienfondé de la bibliothèque de ressources centralisées dont l'un des objectifs est d'améliorer la réutilisabilité de ces ressources. Ils ont confirmé qu'une telle bibliothèque pourrait remplir cet objectif, surtout avec la catégorisation des ressources fournissant un accès rapide à des ressources spécifiques dans une bibliothèque bien fournie. L'indexation des ressources (cf. §4.2.3) a porté ses fruits lors de leur consultation via la fonctionnalité de recherche (bien que le contenu de la bibliothèque soit encore pauvre).

- 2- la phase de construction de composants (configurations génériques) :** nous les avons interrogés sur la procédure de création de composants en tant que configurations génériques. Les auteurs ont répondu, dans un premier temps, qu'avec la disponibilité des ressources catégorisées, la construction d'un composant (configuration générique) est une opération plus simple que la création d'un élément de ressources (pas de fichier à télécharger sur le configurateur). Le filtrage des ressources, qui se conforme aux catégories choisies par l'auteur, a réduit le temps requis pour récupérer les ressources constituant le composant visé. Dans un deuxième temps, ils ont trouvé que le principe de générer une configuration générique (composant) dont les ressources sont éditables, et la possibilité d'en générer plusieurs versions (avec la possibilité d'ajouter librement plusieurs versions de pièces de ressources modifiées de l'original, d'enlever et/ou remplacer des ressources par d'autres) remplit l'objectif de réutilisation des ressources, et enrichit le cas pour lequel un composant est généré.
- 3- la phase de génération de configurations spécifiques :** nous les avons interrogés sur la procédure de compilation des ressources d'un composant pour générer une configuration spécifique. Celle-ci s'est révélée un peu perturbante, nous ont-ils répondu. Le recours à des logiciels tiers les oblige à laisser de côté le logiciel de configuration (le navigateur web) pour utiliser les autres logiciels dédiés (Opale et Unity Pro) afin de réaliser la compilation de chaque fichier natif, puis de rassembler les nouvelles ressources dans un paquetage qu'ils doivent télécharger sur le configurateur (via le navigateur web). Si le configurateur pouvait intégrer ces outils pour ne plus avoir à faire qu'à une seule application web, ils ont estimé que cela simplifierait énormément la procédure et le rendrait plus ergonomique. C'est techniquement faisable : cela suppose de proposer ces applications en accès distant depuis une machine virtuelle à travers un accès web.

4- la phase d'utilisation de configuration : nous avons demandé aux auteurs leur niveau de satisfaction quant à l'utilisation du configurateur. Satisfecit pour la fonctionnalité de recherche qui leur permet d'avoir une idée assez claire sur le contenu lié à un sujet ou à un système donné, en fonction des descriptions associées par les auteurs. Ils ont aussi trouvé que la disponibilité de toutes les ressources dans un paquetage de configuration prêt à l'emploi, pour une session de TP, est un avantage. L'outil installateur bénéficiait aussi d'un avis favorable.

4.6.2 Analyse des objectifs remplis

Rappelons ici les objectifs énoncés lors de la conception de notre configurateur (cf. §3.3.1.8), en les analysant par rapport à la réalisation du prototype et le déroulement des expérimentations. Il s'agissait au départ de :

- **OBJ1** : diminuer d'au moins 50% le temps de configuration d'un SAP ;
- **OBJ2** : améliorer la réutilisation des ressources pédagogiques en réduisant le temps de création des configurations de deux TP proches, d'au moins 50% ;
- **OBJ3** : améliorer l'autonomie de l'instructeur, en diminuant le nombre d'appels à un technicien d'au moins 50%.

Il ne nous a pas été possible de quantifier les temps de référence de préparation des sessions (sans configurateur) auxquels nous aurions pu comparer les temps d'usage du configurateur. Nous avons opté pour des expérimentations faites avec de vrais enseignements mais leur programmation dans le temps et le fait que les enseignants avaient préparé leurs ressources bien avant notre intervention ne permettent pas d'estimer ces temps de préparation de manière suffisamment précise, une estimation est donc fournie. De plus, à l'instar de tout autre logiciel, notre configurateur nécessite un **temps de formation** qui n'était pas compatible avec les disponibilités des auteurs expérimentateurs. Ainsi, pendant chaque expérimentation, nous les avons accompagnés, ce qui, au final, aboutit à un temps d'utilisation légèrement supérieur à celui d'un utilisateur habitué. Chaque expérimentation a pris environ deux heures séparées en tutoriel (30 minutes environ) et expérimental (1 heure et 30 minutes environ). Dans ces conditions, l'estimation du gain temporel est à relativiser.

Compte tenu de cette situation, nous tentons ici de quantifier de manière approximative les indicateurs d'atteinte des objectifs déclarés au départ. Pour cela, nous estimons le timing sur

l'utilisation du configurateur présenté et argumenté dans §4.4.4 (voir Tableau 4.2). Prenons la deuxième session configurée pour chaque système parce que lors de cette session, que ce soit pour le système de magasin vertical ou bien pour le système de transfert libre, la bibliothèque de ressources pédagogiques ainsi que la bibliothèque de configurations disposaient de tous les éléments nécessaires. Les utilisateurs n'avaient donc lors de la deuxième session sur les deux systèmes qu'à créer, générer et installer leurs configurations.

La configuration de chacune des deux sessions sur les deux systèmes a pris 7 minutes, selon le Tableau 4.2. En comparant ce temps avec les résultats du sondage (voir Figure 2.12) diffusé au départ de cette étude, plus de la moitié des participants a déclaré avoir besoin de 15 minutes environ pour configurer une session de TP (les autres participants ont déclaré avoir besoin de plus de temps, cf. §2.7.5). Notre premier objectif est donc atteint.

Pour le deuxième objectif, et en profitant des timings listés dans le Tableau 4.2, nous remarquons que la création de composant (configuration générique) pour la deuxième session a pris, pour l'expérimentation sur chaque système utilisé, moins de la moitié du temps requis dans la première session, grâce à la réutilisation des ressources fournies par le configurateur. Le deuxième objectif est donc rempli.

L'atteinte du troisième objectif n'était pas estimée, vu que les participants n'avaient effectivement pas besoin de l'aide d'un technicien lors des expérimentations réalisées. Cet objectif est donc à réévaluer après l'utilisation du configurateur par plusieurs autres participants.

Nous avons mené en parallèle une mesure qualitative de la satisfaction des utilisateurs ayant utilisé le configurateur. Nous avons évalué le niveau de satisfaction des auteurs pour chaque fonctionnalité utilisée pendant les expérimentations. À travers l'intérêt porté par les auteurs sur l'outil, après utilisation, nous évaluons qualitativement l'adéquation entre la demande et la réponse offerte via ce prototype. Nous avons diffusé un questionnaire aux participants (voir l'**Annexe E**). Pour chaque question, l'utilisateur peut choisir l'une des quatre options:

1. très satisfaisante ;
2. assez satisfaisante ;
3. peu satisfaisante ;

4. pas du tout satisfaisante.

Le Tableau 4.3 résume les résultats de cette évaluation d'intérêt du configurateur selon les fonctionnalités principales (cinq participants aux expérimentations ont répondu aux questions).

Fonctionnalité	Niveau de satisfaction	Commentaires des utilisateurs
F 1 : insertion des ressources	Peu satisfaisante	<ul style="list-style-type: none"> - Il ne s'agit pas d'un téléchargement simple sur le configurateur, c'est une procédure longue. - Il faut réfléchir à une description représentative pour chaque pièce de ressources indiquant aux autres utilisateurs leur contenu. - Il faut aussi bien choisir des noms de catégories à créer pour indiquer les ressources y associées.
F 2 : modification des ressources	Assez satisfaisante	<ul style="list-style-type: none"> - La fonctionnalité d'édition du contenu d'un élément de ressource (fichier) nécessite l'utilisation d'un logiciel tiers. - L'édition d'un composant est aussi simple que sa création. - L'édition d'une configuration est plus simple que sa création. Sauf s'il s'agit de recompiler de(s) ressource(s), cela revient une opération perturbatrice. - La gestion des droits d'accès aux ressources les protège d'une intervention non désirée.
F 3 : sauvegarde des ressources	Très satisfaisant	<ul style="list-style-type: none"> - L'indexation des ressources selon des catégories simplifie leur traçabilité. - La gestion des droits d'accès protège l'utilisateur d'utiliser des ressources non prêtes.
F 4 : recherche et réutilisation des ressources	Très satisfaisante	<ul style="list-style-type: none"> - L'utilisation des mots clés identiques aux noms des catégories indique au moteur de recherche de filtrer les ressources listées selon ces catégories. - Le moteur de recherche récupère des ressources qui contiennent dans leurs titres et/ou description les mots identiques à ceux dans la phrase de recherche (récupère des ressources précises par nom et/ou selon objectif pédagogique décrits dans le texte de recherche). - La disponibilité des ressources aux formats éditables simplifie leur édition et réutilisation.
F 5 : l'assemblage automatique de ressources	Très satisfaisant	<ul style="list-style-type: none"> - Le filtrage des ressources selon des catégories réduit le temps d'en repérer pour les assembler sous un composant - La génération automatique d'un paquetage de ressources éditables améliore leur réutilisabilité par le même ou différents utilisateurs.
F 6 : la génération automatique des configurations	Peu satisfaisante	<ul style="list-style-type: none"> - L'obligation de laisser de côté le configurateur pour compiler les ressources d'une configuration à l'aide des logiciels tiers est perturbante.
F 7 : exploitation des configurations d'e-TP	Assez satisfaisante	<ul style="list-style-type: none"> - L'installateur a amélioré l'autonomie de l'instructeur.

Tableau 4.3 : Résultats de l'évaluation de performance du configurateur par les utilisateurs.

En transformant ces quatre options en chiffres, de 1 à 4, nous avons construit une courbe représentant le niveau de satisfaction atteint actuellement (voir Figure 4.24, trait en pointillé). La moyenne (points-trait) entre le niveau de satisfaction atteint par la version actuelle du configurateur et le niveau idéal (trait plein), on obtient le degré d'insatisfaction des

utilisateurs par le configurateur. En effet, ce moyen se trouve au-dessus du niveau 3. Les utilisateurs sont donc satisfaits en moyenne à 78.5% par le configurateur. L'écart sur la satisfaction de certaines fonctionnalités est produit par des limites techniques (faciles à corriger) dans la version actuelle du configurateur, et les conditions d'expérimentations.

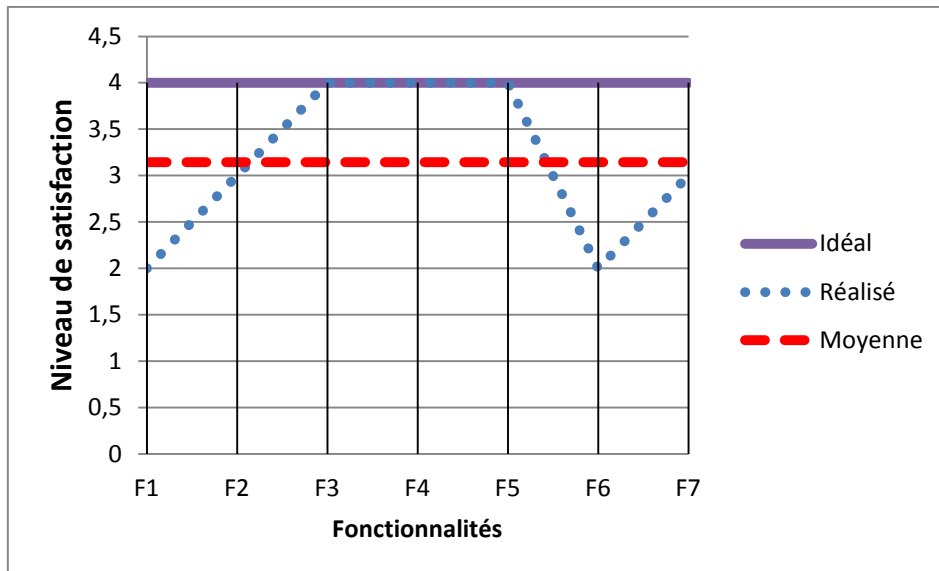


Figure 4.24 : Analyse de niveau de satisfaction des utilisateur par le configurateur.

Nous avons choisi deux systèmes automatisés. Le magasin vertical, correspond à un système de petite taille et l'objectif était de vérifier le fonctionnement du configurateur. Comme l'utilisation de ce système se limite à quelques cas, le principe de réutilisation des ressources se limite à aux ressources développées sur les mêmes configurations : on redéroule les mêmes sessions de TP sans profiter des ressources et configurations pour en générer de nouvelles correspondant aux nouveaux scénarios pédagogiques.

Concernant l'expérimentation sur le transfert libre, et malgré le fait que les ressources exploitées soient spécifiques et donc plus difficilement réutilisables (cf. §4.4.2.3.1), les expérimentations ont démontré que le configurateur remplit l'objectif de réutilisabilité des ressources. L'automatisation de l'installation des configurations a été appréciée par l'instructeur. Dans la version actuelle du configurateur, nous n'avons pas intégré la méthodologie de génération automatique de code source compatible avec la norme IEC 61131-3. L'intérêt de cette méthodologie est d'automatiser la compilation et donc l'installation de programmes sur un SAP.

L'expérimentation dans le domaine de l'informatique a confirmé l'applicabilité du configurateur dans d'autres domaines. Les remarques des utilisateurs ont mis le doigt sur les

parties trop spécifiques (de l'automatique) de notre modèle, tout en confirmant que le prototype de configurateur a le potentiel pour répondre aux besoins pédagogiques et techniques dans d'autres disciplines que l'automatique.

4.6.3 Limites et points à améliorer

Lors des trois expérimentations, tous les utilisateurs se sont plaint du **temps d'insertion** des ressources dans le configurateur. La **réutilisabilité des ressources insérées** est aussi un problème : nous avons utilisé le format standard de Unity Pro pour enregistrer les programmes API (au format *.STA), ce qui limite drastiquement leur réutilisabilité avec d'autres logiciels de développement de programmes API. Techniquement parlant, il est toutefois possible avec Unity Pro d'exporter les fichiers au format *XEF* (FEF pour PL7Pro) au format OpenPLC, ce qui, dans l'avenir devrait améliorer cet aspect. Il en va de même pour les ressources documentaires et les scénarios pédagogiques. Pour réutiliser un fichier de documentation ou de scénario, l'utilisateur doit aujourd'hui disposer du logiciel Opale. Cependant, le contenu enregistré n'est réutilisable par un autre logiciel qu'après conversion car encodé de manière lisible en XML.

Le double format des ressources est un aspect « complexifiant » : à différents moments du processus de création, un élément peut être « générique » (fichier source Opale, FEF, ...) puis « spécifique » (fichier PDF, paquetage SCORM, fichier STA, ...). Il n'existe pas aujourd'hui de format générique pour un type de données (document multimédia, programme API, ...) ; à l'avenir, de tels standards aideraient notre configurateur à gagner en intérêt. D'autre part, une intégration de l'ensemble des logiciels nécessaires à cette préparation sous forme d'application web capables de coopérer sans la nécessité de transférer manuellement des données augmentera également grandement l'ergonomie de notre configurateur.

Finalement, les utilisateurs ont estimé que le configurateur est un outil intéressant et nécessite d'être utilisé d'une façon permanente pour enrichir le contenu de la bibliothèque de ressources ainsi que celle de configurations. Sur le problème technique lié à la réutilisabilité des ressources mentionné ci-dessus, les utilisateurs impliqués pensent qu'un contenu plus riche des bibliothèques peut indiquer les types de fichiers les plus fréquents pour chaque type de ressources, et donc les logiciels auteurs fréquemment utilisés. Cela nous aidera à développer des fonctionnalités de création/édition des ressources sans avoir besoin de logiciels extérieurs.

Les ontologies de catégorisation des ressources doivent être étendues pour que le configurateur soit exploitable dans d'autres domaines scientifiques.

4.7 Conclusion

Dans ce chapitre, nous avons réalisé un **prototype de configurateur** selon les modèles présentés dans le chapitre 3. Ce prototype dispose du minimum de fonctionnalités nécessaires pour dérouler des expérimentations sur des TP dans le domaine d'automatique. Le choix de la plate-forme Drupal pour réaliser le configurateur nous a libérés de la programmation des fonctions de base (création, édition, suppression de contenus, gestion des droits d'accès, ...) afin de pouvoir nous concentrer sur les fonctions spécifiques au configurateur.

Les conditions de déroulement des expérimentations ainsi que le temps consacré ne nous auront permis d'obtenir que des estimations des gains temporels apportés par le configurateur, mais l'enquête de satisfaction menée auprès des utilisateurs impliqués a convergé vers une conclusion positive et encourageante. L'analyse de l'usage du configurateur a montré le potentiel de ce concept non seulement dans le domaine des automatismes mais aussi dans d'autres, comme l'informatique.

L'utilisation continue de ce système pourra enrichir les contenus pédagogiques et en réaction attirer l'attention et l'intérêt d'autres utilisateurs dans ces domaines. La valeur ajoutée de ce système pour d'autres domaines scientifiques reste à être confirmée en élargissant la panoplie des champs disciplinaires.

L'extension du configurateur à d'autres domaines doit conduire à harmoniser les services fournis et à augmenter la genericité du configurateur.

Cette recherche ouvre donc plusieurs perspectives. Dans un premier temps, l'amélioration des fonctionnalités du configurateur, vues les limites techniques rencontrées pendant les expérimentations et son applicabilité sur différentes plates-formes. Dans un deuxième temps, la prise en compte des remarques des utilisateurs en ce qui concerne l'ergonomie du logiciel. Dans un troisième temps, l'augmentation du niveau de genericité de configurateur dans le but d'étendre la préparation des e-TP dans d'autres domaines scientifiques que l'automatique.

Conclusion générale

Contexte

Depuis les années « 90 », les « Sciences et Technologies de l'Information et de la Communication (STIC) » ont donné naissance aux « Technologies de l'Information et de la Communication dans les Enseignements (TICE) », dont une des thématiques concerne les Environnements Informatiques pour l'Apprentissage Humain « EIAH », couvrant l'ensemble des modes pédagogiques. La nécessité de confronter l'apprenant à la réalité, surtout dans les disciplines scientifiques et techniques, rend le « TP » irremplaçable par des environnements exclusivement virtuels. Ce qui ne veut pas dire que les EIAH ne peuvent pas être utilisés pour la réutilisabilité de ces ressources. Les environnements que l'on est alors amené à développer dépendent du contexte des activités conduites par l'apprenant (simulation, télé-opération, téléTP, ...), du public apprenant (adaptation), du formateur (personnalisation) et la discipline enseignée (médecine, chimie, production, ...).

Dans la recherche bibliographique effectuée environ 40 citations portent sur l'informatisation des expérimentations dans le domaine de l'automatisation. Les activités proposées au sein des plates-formes d'e-TP se limitent à la manipulation de matériels spécifiques, sans possibilité de réutiliser les ressources pédagogiques exploitées ou d'échanger des pratiques sur des plates-formes similaires. La genericité de plate-forme d'e-TP est donc un sujet majeur qui doit apporter des solutions pour gérer la création/édition des contenus pédagogiques ainsi que la préparation des sujets de TP, dans le but d'améliorer leur réutilisabilité. Une telle plate-forme ne peut se généraliser et s'intégrer dans une plate-forme de e-formation que si elle réutilise les outils généraux (communication, évaluation, ...) de manière homogène et s'ouvre à l'échange de ressources pédagogiques entre plates-formes à l'instar de documents pédagogiques plus classiques (scénarios de cours, d'exercices, ...).

Forts de ces constats, nous avons entrepris d'élaborer un système gérant le processus de configuration d'un Système Automatisé de Production capable de fournir, aux auteurs et instructeurs de TP, les moyens de créer et manipuler des ressources pédagogiques, tout en fournissant le moyen de (ré)configurer le SAP sujet du TP. L'objectif est de réduire le temps de configuration d'un SAP, en réduisant la dépendance de l'instructeur à un technicien et en améliorant la réutilisabilité des ressources pédagogiques.

Choix et démarche

Nous avons suivi l'approche **MDA** pour modéliser notre système à l'aide du langage SysML. En nous appuyant sur les modèles **CIM** et **PIM** de cette approche, nous avons pu reporter le choix de technologie cible pour implémenter notre solution de configurateur jusqu'à la phase finale. Après avoir réalisé une étude bibliographique du domaine, nous avons étudié les besoins des utilisateurs (auteur et instructeur de TP en automatique). En réponse à ces besoins, nous avons spécifié et réalisé un système visé, à partir de modèles structurels et des fonctionnalités adaptées à chaque phase d'utilisation. Notre choix technologique de développer le configurateur sous forme **d'application web** répond à des besoins de simplicité d'utilisation ainsi que de partage et de réutilisabilité des ressources pédagogiques. Enfin, le choix des expérimentations présentées valident le configurateur à la fois dans son bon fonctionnement et dans son utilité.

Résultats

Les retours d'usage sont organisés via une interrogation directe des participants et à travers des questionnaires. L'analyse technique (cf. §4.6.1) a pour but de mesurer la satisfaction des utilisateurs lors de l'utilisation du configurateur et de l'enchaînement des opérations. Cette mesure nous a aidés à construire des indicateurs « qualité » des fonctionnalités développées, et de l'aptitude de ce logiciel à être couramment utilisé et à continuer de se développer. Les participants ont témoigné de leur satisfaction sur les fonctionnalités fournies (niveau de satisfaction : 87.5 %). Seul bémol : le temps requis pour créer (insérer) des ressources.

Le constat de l'atteinte des objectifs annoncés (cf. §4.6.2) a pour but de s'assurer de la valeur ajoutée de notre système au sein d'e-TP. Les résultats des deux expérimentations en automation, à travers deux sessions pour chacune, montrent que les objectifs de diminuer d'au moins 50% le temps de préparation et de reconfiguration d'un SAP sont atteints. Malheureusement, le troisième objectif sur l'amélioration de l'autonomie d'instructeur n'a pas pu être estimé parce que dans les deux expérimentations, les participants n'ont effectivement pas eu besoin de l'aide d'un technicien.

Apports et contributions

Nos apports et contributions scientifiques à la configuration d'e-TP en automatique peuvent se résumer à :

- un **état de l'art**, avec une étude analytique et comparative, sur les différents types d'e-TP afin de situer l'état actuel de notre recherche ;
- un approche et une **méthodologie globale** efficaces ayant conduit à un modèle générique de Configurateur ;
- le recours à des **techniques idoines pour résoudre des problèmes** particuliers de configuration : l'adoption du langage de modélisation SysML pour modéliser des aspects matériels du système, l'adoption du CMS Drupal pour la gestion des utilisateurs ainsi que la création et l'édition de contenus ;
- l'introduction d'une **application web** libérant les utilisateurs de certaines installations logicielles, et accessible depuis n'importe où sans se limiter à un système d'exploitation précis.

Limites

Nos travaux présentent bien sur des limites parmi lesquelles:

- le faible contenu des bibliothèques des ressources pédagogiques et de configuration ;
- la réutilisabilité des ressources, limitée aux utilisateurs possédant des logiciels capables de manipuler les formats des différents fichiers de ressources disponibles dans le configurateur ;
- les fonctionnalités de l'installateur non testées sur une autre plate-forme que celle de l'atelier « AIP-Priméca RAO », ce qui conduit à l'incertitude de la généricité prétendue.

Perspectives

La démarche MDA adoptée pour construire le modèle générique de Configurateur ouvre la porte à plusieurs évolutions, à court et à long terme, à la fois pour notre modèle et pour l'environnement informatique qui le supporte. Les principaux développements envisageables :

- étudier l'applicabilité du Configurateur dans d'autres domaines scientifiques que l'automatique ;
- améliorer les fonctionnalités de création/édition des ressources en intégrant des outils d'édition au sein du configurateur afin de libérer l'utilisateur d'utiliser des logiciels externe (tels que Unity et Opale) ;

- intégrer une fonctionnalité de recherche par sémantique pour améliorer la réutilisation des ressources.

Annexe A : L'approche objet

L'approche objet est « une méthode de modélisation basée sur une représentation abstraite des entités du monde réel, une méthode qui regroupe les données et les traitements sur ces données au sein d'une entité unique : l'objet »⁶⁶. Cette méthode exprime les fonctionnalités sous la forme de collaboration entre les objets.

D'après [Chiron-2007], la réutilisation de solutions techniques qui ont déjà fait leurs preuves lors de développements précédents représente une création de valeur indirecte mais réelle pour l'entreprise. Avec l'augmentation régulière de la complexité des systèmes modélisés, résultant d'une approche globale, et la difficulté pour les concepteurs de visualiser le développement dans son intégralité, il est devenu incontournable de décomposer les problèmes à résoudre et d'apporter les solutions à chaque élément ainsi identifié. Le concept d'objet en informatique est né de ces problématiques en proposant de stocker un ensemble de données et d'instructions dans une entité unique (*classe*) réutilisable ses (*objets*) dans différents contextes. On trouve des exigences similaires dans le domaine de l'automatisation, bien que les objets du monde de l'automatisation doivent représenter à la fois des mécanismes virtuels (comme en informatique) et physiques (flux de matière, actionnements, ...).

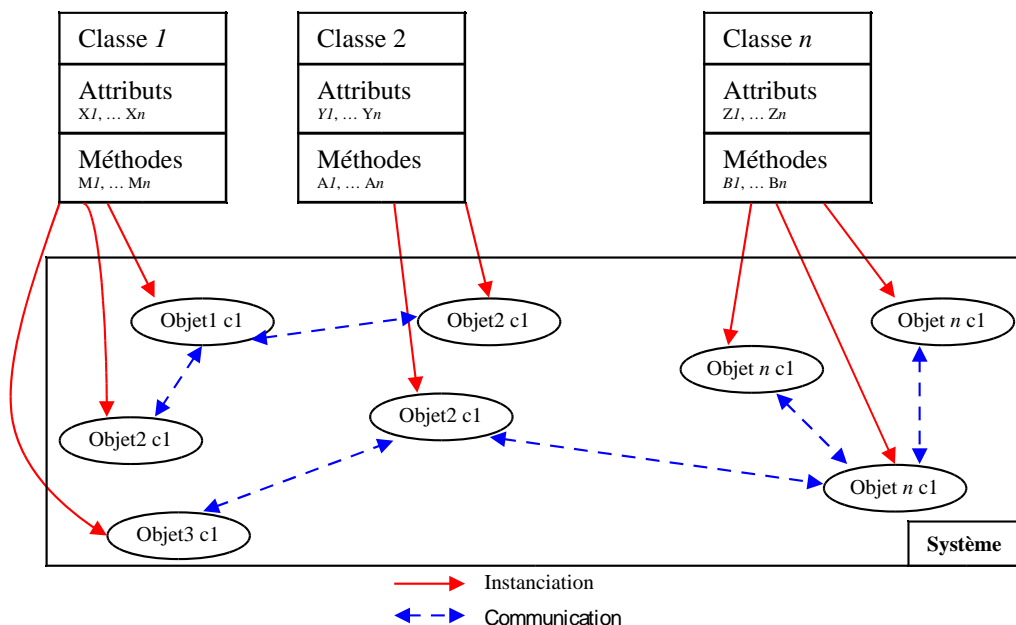


Figure 1: La collaboration entre les objets pour élaborer une tâche du système.

L'approche orienté objet propose le concept de « *Classe* » qui contient des attributs (données) et des méthodes (fonctions). L'organisation de ces attributs et méthodes dans la

⁶⁶ Définition de Pierre-Alain Muller, Professeur des universités en informatique - Université de Haute-Alsace.

classe repose sur trois principes essentiels : l'encapsulation, l'héritage et le polymorphisme [Supavita-2009]. L'objet représente donc une instanciation d'une classe, c'est-à-dire : l'objet comporte une copie des attributs et des méthodes de la classe. Un système peut donc être vu comme un ensemble d'objets qui collaborent et communiquent entre eux pour élaborer une tâche donnée (voir Figure 1).

Le principe de la programmation orientée objet (POO⁶⁷) est aujourd'hui largement répandu dans tous les langages de programmation (C++, Java, Python, ...). Les avantages de cette approche ne se limitent pas à la réutilisabilité des solutions produites, mais également à l'extensibilité de ces solutions et à leur flexibilité de s'adapter à des cas similaires.

1. Approche orientée objet en informatique

L'approche orientée objet (traduction de l'anglais : « Object Oriented Approach ») est reconnue comme principe fondamental dans plusieurs méthodes modernes de conception de logiciels. Cette approche, selon plusieurs recherches dont [Supavita-2009, Kaur-2011], n'est pas seulement associée à plusieurs langages de programmation d'utilisation commune, mais aussi à des techniques de modélisation et de conception graphiques, parmi lesquelles UML (Unified Modeling Language) [UML2-2005] représente le langage de modélisation le plus connu dans le domaine de la conception logicielle. Les avantages de cette approche sont :

- la réutilisabilité du code : c'est l'aptitude à être repris pour de nouvelles applications ;
- la modularité des programmes : c'est l'aptitude à construire une application selon une combinaison de plusieurs modules indépendants et réutilisables ;
- et la conception par extension : c'est l'aptitude à définir de nouveaux composants logiciels basés sur ceux déjà développés, soit par l'agrégation des blocs basiques et hétérogènes soit en ajoutant de nouvelles fonctionnalités à un composant plus générique.

Les techniques de développement de logiciels ont longtemps reposé sur des langages de programmation structurels. La notion d'objet a été introduite dans les années 1960, ce qui a provoqué dans les années suivantes le développement de nouveaux langages de programmation supportant ce concept. Suite à l'apparition de nouveaux systèmes d'exploitation graphiques (Microsoft Windows 95, par exemple) et des réseaux informatiques

⁶⁷ Définition de POO selon [Madsen-1988] : dans la programmation orientée objet, l'une exécution d'un programme est considérée comme un modèle physique simulant le comportement d'une partie soit réel ou imaginaire du monde.

(Internet), les techniques de langages orientés objet ont évolué en ouvrant la voie au développement d'applications graphiques, et ensuite d'applications web (voir Figure 2).

Les langages orientés objet ont rapidement été adoptés par les développeurs, conscients de l'important tournant que l'informatique venait de prendre et des nouvelles possibilités qui s'ouvraient à eux dans les techniques de programmation, ainsi que du temps qu'ils allaient gagner lors de leurs développements grâce à la réutilisation de fonctions préprogrammées. Ainsi, dans les années 1990, pas moins d'une cinquantaine de méthodes orientées objet ont vu le jour, se basant sur les différents concepts définissant une approche orientée objet pour les programmes informatiques [Chiron-2007]. Une illustration de la chronologie de l'évolution des langages de programmation les plus connus est proposée en Figure 2.

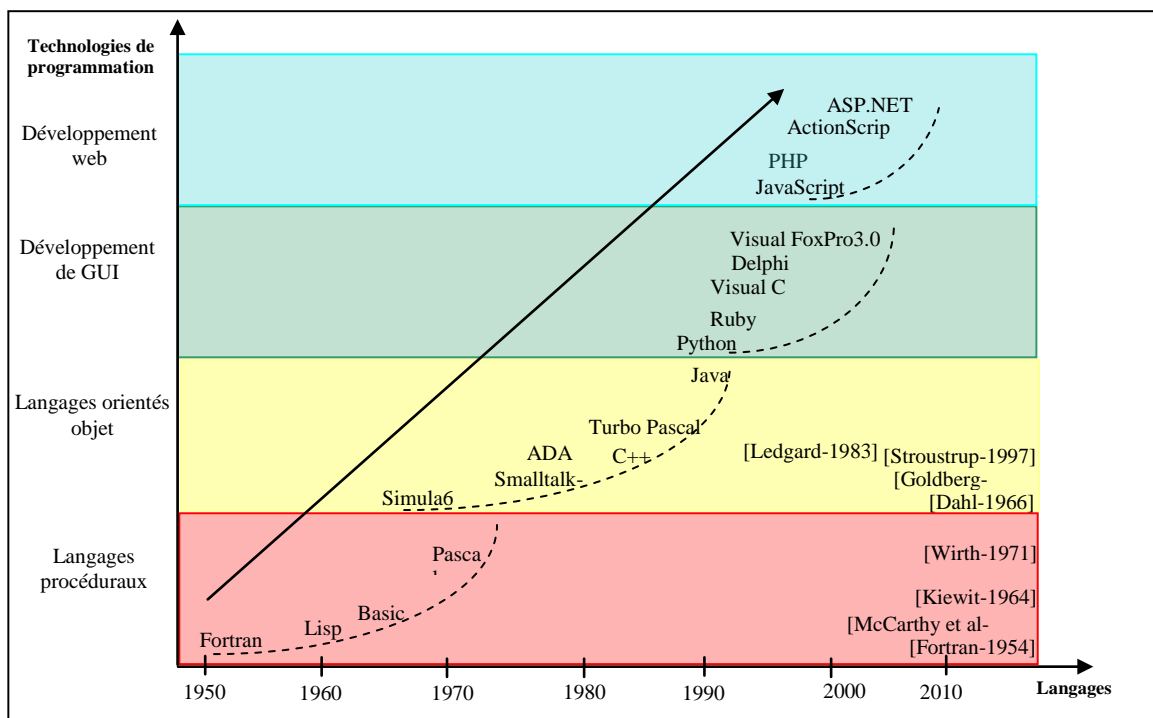


Figure 2 : Évolution des langages et techniques de programmation orienté objet.

2. Application dans le domaine de l'automatisation

Les concepts de programmation, dans le domaine de l'automatisme, n'ont pas évolué aussi vite qu'en informatique du fait des contraintes de sûreté de fonctionnement et de la gestion de la mémoire limitée des automates. Le concept de Bloc de Fonction « FB » (*Fonction Block*) a été introduit dans la norme IEC 61131-3 [John-2001], mais les avantages de l'approche objet dans ce domaine sont encore loin d'avoir été valorisés comme en informatique. Ce fait, a

même conduit à proposer un nouveau langage de programmation d'automates, comme constaté dans l'étude bibliographique des travaux de thèse de F. Chiron [Chiron-2007]. Actuellement [Bonfe-2006] la notion d'héritage (et donc toutes les notions liées, tel que le polymorphisme) est absente de la spécification IEC 61131-3 . Les solutions logicielles dans le domaine de l'automatisme présentent donc encore aujourd'hui des limites en ce qui concerne leur réutilisabilité et leur adaptabilité.

2.1. POO dans les environnements de programmation en automatisme

Reconnaissons une progression récente dans l'intégration des « techniques » de programmation orientée objet dans les programmes pour automatismes :

Certains fournisseurs d'environnement de développement tels que Schneider Electric ou Dassault Systèmes intègrent dans leurs outils de développement des mécanismes de POO. Par exemple, Schneider Electric⁶⁸ a introduit le concept de DFB (*Derived Function Block*) dans son logiciel Unity Pro [Unity_Pro-2011]. Grâce aux DFB le développeur peut définir un modèle de bloc de fonction possédant des variables d'entrées, de sorties, des variables internes privées ou publiques, et dont le comportement est codé dans un des langages de la norme IEC61131-3 (LD, ST, IL et/ou FBD). Ce modèle peut être ensuite instancié autant de fois que nécessaire dans un autre DFB (encapsulation) ou dans une section de programme automate. Toute modification du modèle engendre la modification de chaque instance. Par contre, aucun mécanisme d'héritage n'est proposé.

Les variables déclarées dans un DFB n'ont pas d'allocation mémoire fixe, ce qui rend le DFB indépendant du logiciel dans lequel il sera instancié et a fortiori du matériel. Un DFB bien conçu peut donc être réutilisé dans différentes applications. Les avantages des DFB sont [Unity_Pro-2011] :

- simplification de la conception et de la programmation, en plus d'une augmentation de la lisibilité du code ;
- facilitation du débogage de l'application (toutes les variables manipulées par le bloc de fonction sont identifiées dans son interface) ;
- réduction du volume du code généré : le code correspondant à un DFB n'est chargé qu'une seule fois. Seul le volume de données est proportionnel au nombre d'instances.

⁶⁸ Voir : <http://www.schneider-electric.com/site/home/index.cfm/ww/>

Ce concept de bloc de fonction instanciable est aussi proposé par Siemens [Siemens-2009] (en tant que UDFB « *User Defined Function Block* »). Un programme peut contenir une ou plusieurs définitions d'UDFB et autant d'instances que nécessaire. La modification du modèle est répercutée automatiquement sur toutes les instances du programme, Mais, contrairement aux DFB de Schneider, chaque UDFB ne peut contenir que des sous-programmes utilisant des fonctions standards (issus des bibliothèques de fonctions du constructeur) : impossible d'instancier un UDFB à l'intérieur d'un autre et donc ainsi de créer des structures imbriquées [Siemens-2001]. Les UDFB sont préconisés pour des fonctions bas-niveau présentes en nombre dans un programme d'automatisme (gestion bas niveau de pompe, de vérins, ...).

Un autre atelier de développement proposant des mécanismes de POO en automatisme est ControlBuild de Dassault Systèmes [Evrot-2008]. Il s'agit, entre autres fonctions, d'un atelier logiciel pour la conception et la validation du contrôle-commande de systèmes automatisés. Le logiciel propose de concevoir et réaliser un programme pour API en utilisant les langages de la norme IEC 61131-3 et des schémas électriques, en se basant sur des modèles réutilisables. Il génère ensuite le code automate (au format PLCopen ou propriétaire : Schneider Unity, Siemens STEP7, Rockwell Rslogix). Dans ControlBuild le développeur peut, à travers un éditeur de composants complexes (appelés MAC), représenter des associations de composants simples ainsi que les échanges d'informations (connexions) entre ces composants (voir Figure 3).

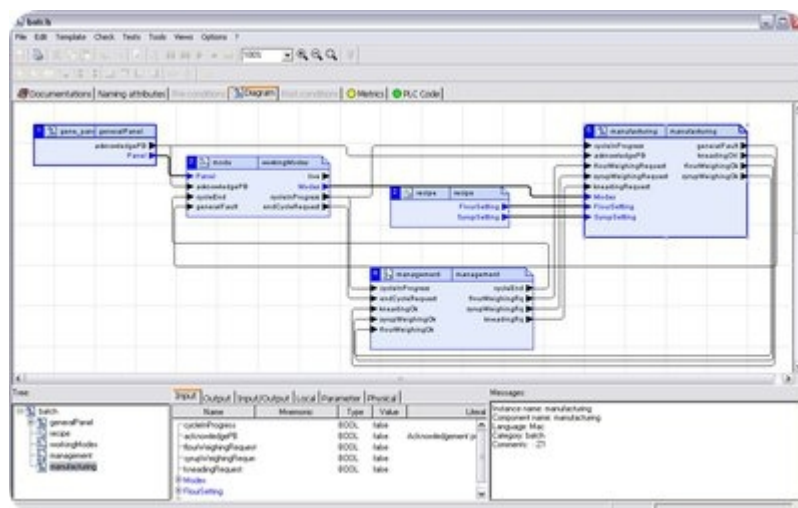


Figure 3 : ControlBuild : l'éditeur de composants complexes (MAC)⁶⁹.

⁶⁹ Voir le site : http://www.geensoft.com/fr/article/controlbuild_design/

Un composant MAC est un élément de programme qui a pour vocation d'être utilisé plusieurs fois dans la même application et sur d'autres projets. Ces MAC peuvent représenter, selon l'objectif recherché (voir Figure 4) :

- des modèles de comportement :
 1. à des fins de simulation fonctionnelle d'une partie opérative :
 - des éléments de la Partie Opérative (équipements électromécaniques, machines...)
 - des modèles de déplacement et de détection d'objets (convoyeurs, butées, capteurs...)
 2. à des fins de contrôle commande :
 - des asservissements de position, régulation, filtres ;
 3. des flux d'énergie et d'information (température, pression, débit...)
 4. des événements aléatoires ou actions à temps variables (pannes mécaniques, défaillances de capteurs, comportement des opérateurs, etc.) ;
- des éléments de pupitre de conduite et animation graphique des modèles de comportement.

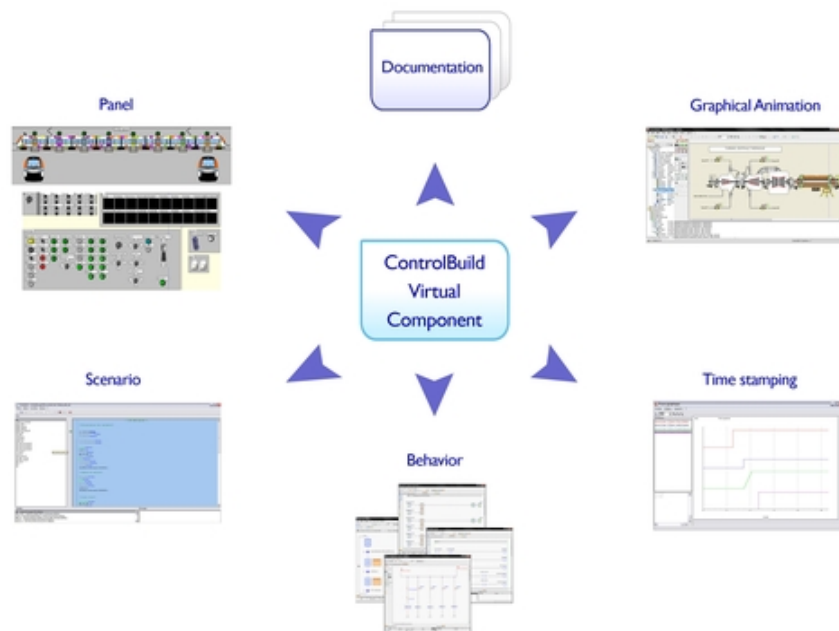


Figure 4 : Réutilisation du savoir faire et bibliothèques que ControlBuild-Design⁷⁰ dispose à ses utilisateurs.

⁷⁰ Figure extraite du site : http://www.geensoft.com/fr/article/controlbuild_design/

Des MAC sont proposés par défaut dans des bibliothèques de composants (ouverts : il est possible de visualiser comment ils sont construits) mais peuvent aussi être développés par le programmeur qui peut ensuite les classer et les archiver afin de capitaliser le savoir faire de son entreprise.

Dassault Système va plus loin que Schneider Electric et Siemens dans l'approche POO : les composants intègrent en plus du comportement qu'ils doivent reproduire, des éléments graphiques illustrant leur état (pour les tester et animer dans une interface homme machine et/ou dans une interface de simulation). Par ailleurs, des notices et guides d'utilisation textuelles (fonctionnel, interfaces, spécifications de test) sont également associées à chaque composant. Ainsi il est possible avec ce logiciel de générer automatiquement la documentation du programme notamment par agrégation des documentations de chaque composant instancié.

2.2. Langages POO en informatique au service de l'automatisme

Une autre approche consiste à faire appel aux langages objet traditionnellement utilisés en informatique. (Java, C++, par exemple) plutôt qu'aux langages traditionnels de la norme IEC61131-3. L'usage de langages plus classiques comme le C est usuellement réservé aux systèmes embarqués, aux systèmes programmés *ad vitam aeternam* ou aux systèmes temps-réel nécessitant de fortes puissances de calcul et non aux automates programmables industriels classiques dans le domaine manufacturier. Notons tout de même que de plus en plus de fournisseurs proposent des API pouvant être programmés en C (comme par exemple Advantech). Le passage à un langage objet est redouté du fait des mécanismes cachés qui rendent la programmation plus aisée mais l'exécution moins maîtrisée dans des applications temps réel.

Maja Arvehammar a proposé dans son mémoire de master recherche un petit système d'automation orienté objet [Arvehammar-2007]. Il s'agit d'un système de magasin d'alimentation de feuilles (*a sheet feeding magazine*) contrôlé via une machine JAVA temps réel. L'auteur a fait appel à UML pour le développement d'un modèle orienté objet en utilisant les diagrammes de classe et des machines d'état. Le programme Java accède au matériel via des fonctions C accessibles dans des bibliothèques de fonctions fournies par le fournisseur du matériel (voir 5). La contribution principale de ce travail a consisté à tester cette architecture de contrôle novatrice sur un banc d'essai et à examiner si l'environnement Java fonctionne correctement avec l'équipement d'automatisme et répond aux exigences critiques du temps

réel. L'idée novatrice de ce projet est d'exploiter les techniques de programmations orientée objet (instanciées ici en Java) dans un environnement d'automatique. Une perspective de ce travail est de développer une fonctionnalité générique de base pour le contrôle de mouvement, puis d'utiliser l'outil UML ou SysML pour le développement de l'application désirée.

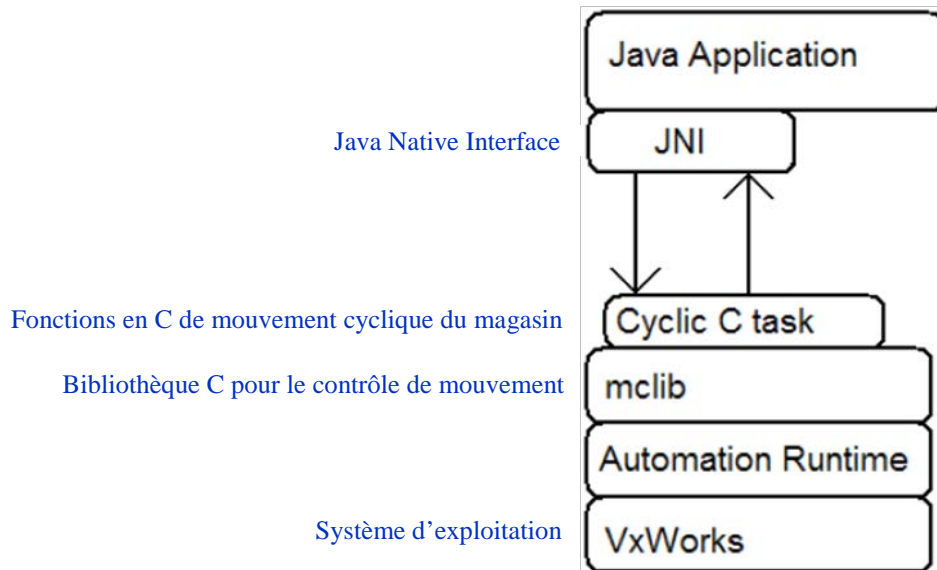


Figure 0A.1 : L'architecture de solution orientée objet⁷¹.

Le logiciel LabVIEW (National Instruments) intègre un langage objet pour la communication avec des instruments d'acquisition de données, des caméras de vision industrielle, des dispositifs de commande de mouvement, ... à travers des interfaces telles que GPIB (*General Purpose Interface Bus*), PXI (*PCI eXtensions for Instrumentation*), VXI (*Virtualization Experience Infrastructure*), RS232⁷² ou encore RS485⁷³ [LabVIEW-2007]. La programmation orientée objet est devenue disponible dans LabVIEW (LVOOP : *LabVIEW Object Oriented Programming*) à partir de sa version 8.2 [LabVIEW_OOP-2009], en intégrant deux concepts principaux de la POO : l'encapsulation et l'héritage. Une étude de conception a été menée dans [Beck-2007] pour tester l'applicabilité de la LVOOP pour la conception des systèmes de contrôle, et une illustration des concepts de POO et de design patterns dans LVOOP est aussi présentée. L'auteur conclut que la gestion de mémoire de LVOOP est bien faite et l'accès aux données via membres de classe peut être accompli rapidement (même pour une taille de données dépasse 10 MB). Il s'avère aussi que

⁷¹ Extrait de [Arvehammar-2007] p.34.

⁷² RS-232 (parfois appelée EIA RS-232, EIA 232 ou TIA 232) est une norme standardisant un bus de communication de type série sur trois fils minimum (électrique, mécanique et protocole).

⁷³ EIA-485 (souvent appelée RS-485) est une norme qui définit les caractéristiques électriques de la couche physique d'une interface numérique sérielle.

l'utilisation des méthodes abstraites des classes de base simplifie le codage et améliore la réutilisabilité du code.

Le choix d'utiliser un langage informatique dans le domaine d'automatisme semble une solution acceptable dans certains cas, comme dans les systèmes temps-réel ou les systèmes embarqués qui nécessitent de fortes puissances de calcul, mais cette utilisation reste redoutée du fait des mécanismes cachés qui rendent la programmation plus aisée mais l'exécution moins maîtrisée dans des applications temps réel.

Annexe B : Langages et méthodologies de conception orientée objet

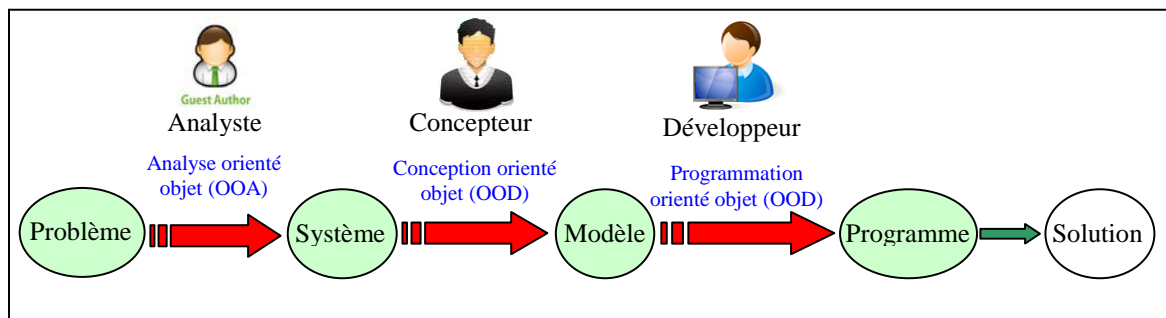


Figure 0.1 : Les trois étapes générales de la modélisation orientée objet.

La modélisation orientée objet consiste à créer un modèle informatique du système qui peut rassembler aussi bien des éléments du monde réel que des concepts ou des idées propres au métier ou au domaine duquel fera partie le système. Elle passe à travers trois étapes générales (voir Figure 1) : dans un premier temps, cette modélisation commence par l'analyse orientée objet OOA (*Object-Oriented Analysis*) qui consiste à définir, à qualifier ces éléments sous forme de types, donc indépendamment de la mise en œuvre. Dans un deuxième temps, la modélisation implique de faire une conception orientée objet OOD (*Object-Oriented Design*) qui sollicite de proposer une ou des solutions techniques pour représenter les éléments définis dans le système informatique. Les développeurs s'occupent, dans la dernière étape, de l'implémentation des modèles développés, à l'aide d'un langage de programmation orientée objet OOP (*Object-Oriented Programming*), en concrétisant la solution proposée par cette modélisation.

Selon [Turki-2008], le concepteur analyse le système, son environnement et son domaine de connaissances puis représente par des objets ces éléments du monde réel ou des concepts abstraits. Les objets décrits sont les types des données qui seront utilisés lors des activités de conception orientée objet. Durant cette analyse le concepteur ne prend en compte aucun aspect d'implémentation, seul le champ du problème est important. C'est principalement cette particularité de l'analyse orientée objet OOA qui en fait une méthode extensible à tous les autres domaines non logiciels.

Le choix d'un langage de modélisation orienté objet doit donc prendre en compte la satisfaction de différents domaines ainsi que l'hétérogénéité du système à modéliser. Nous

présentons dans la suite le langage UML, puis, nous motivons le choix du langage de modélisation des systèmes SysML.

1. L'arrivée d'un langage de modélisation unifié (UML)

Dans le secteur du développement logiciel, suite à l'adoption massive de l'approche objet pour la réalisation des applications, et face au besoin pour les développeurs d'avoir à leur disposition de nouvelles méthodologies d'aide à la conception objet, une multitude de méthodes a été lancée dont une cinquantaine entre 1990 et 1995, chacune essayant de s'imposer sur le vaste marché du développement logiciel [Chiron-2007]. Face à cette multitude, il a été décidé de définir un langage commun à toutes les méthodologies, c'est le langage UML adopté et développé par l'OMG⁷⁴.

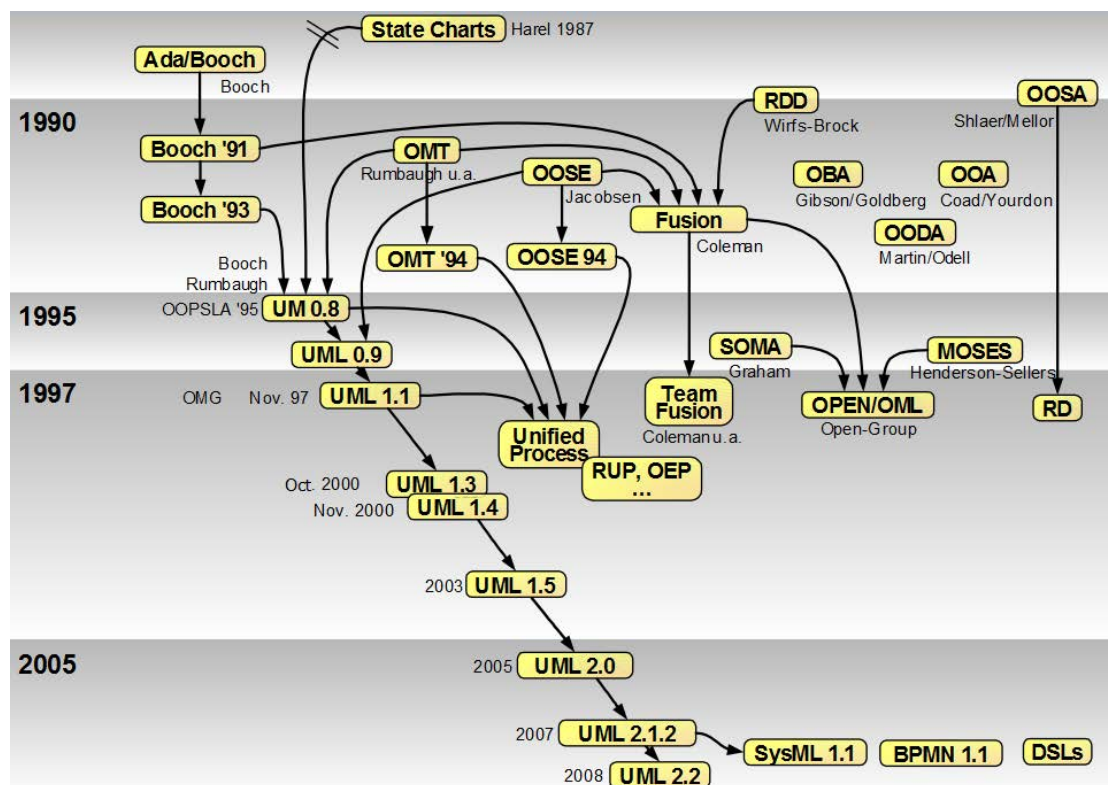


Figure 0.2 : Historique de méthodes orientées objet (AxelScheithauer)⁷⁵.

Ce langage a rapidement fait l'unanimité au sein de la communauté des développeurs en informatique. Son extensibilité avec l'utilisation de profils et de stéréotypes permet de spécialiser un modèle pour un langage donné et de générer automatiquement un squelette de

⁷⁴ (OMG) Object Management Group : est un consortium à but non lucratif initié en 1989 par des industriels désireux de promouvoir l'approche objet dans le monde de la conception, ceci en spécifiant des standards d'échange ou de représentation des données et des systèmes. Voir : <http://www.omg.org/>

⁷⁵ Schéma disponible sur (OO-histoire.svg : <http://commons.wikimedia.org/wiki/File:OO-histoire.svg>), développé par (AxelScheithauer : <http://www.oose.de/>).

code exploitable. La modélisation prend donc du recul par rapport à une technologie de programmation spécifique et se rapproche davantage d'une vision objet puriste, permettant une analyse plus propre pour aborder la conception. La chronologie d'apparition d'UML et les principes globaux tenus en compte lors de sa conception sont décrits dans (Figure 2).

F. Chiron [Chiron-2007] ainsi que S. TURKI [Turki-2008], dans leurs mémoires de thèse, ont présenté une étude bibliographique du langage UML et ses extensions. Ce dernier a résumé les principes globaux tenus en compte lors de sa conception, définis dans [UML_Infra-2004] :

1. La modularité et la réutilisation : Les éléments du langage sont regroupés dans des packages dont la cohésion est maximisée et dont l'interdépendance (ou couplage) est minimisée. Cette modularité améliore la réutilisation des packages de l'infrastructure ainsi que de la superstructure.
2. Une architecture en couches : Elle est réalisée selon deux aspects. Le premier est l'hierarchie des packages qui sépare les éléments de base des éléments de haut niveau qui les utilisent. Le deuxième aspect est visible dans l'architecture en quatre couches d'abstractions.
3. Le partitionnement : Les packages de l'infrastructure d'UML sont partitionnés de manière à faciliter la création de nouveaux langages à partir du méta-langage d'UML. Les packages de la superstructure sont partitionnés de manière modulaire.
4. L'extensibilité : Le partitionnement et la modularité favorisent l'extension par création de nouveaux langages basés sur l'infrastructure d'UML. De plus, des mécanismes d'extension permettent l'adaptation d'UML à des domaines particuliers.

2. Le langage de modélisation des systèmes SysML

UML a été construit pour le génie logiciel, et les vues et diagrammes qu'il offre sont donc celles utiles à la spécification de logiciels [Turki-2008]. Pour les systèmes hétérogènes composés de sous-systèmes non logiciels (matériels), la tâche de conception devient de plus en plus compliquée et exige une approche plus sensible à l'interdisciplinarité de tel système. L'ingénierie système (cf. section 4 de cette annexe) a pour objectif de contrôler la conception de systèmes dont la complexité ne permet pas le pilotage simple. Certaines faiblesses d'UML

devaient donc être comblées pour en faire un langage efficace à l'ingénieur système, citons [Turki-2008] :

- besoin de décrire les exigences dans le modèle UML, et d'en assurer la traçabilité vers la conception ;
- besoin de représenter des éléments non-logiciels et d'en spécifier le type (mécanique, circuit, hydraulique, câblage, capteur...) ;
- besoin de représenter des attributs de performance, des attributs physiques et non comportementaux ;
- types de données pour des éléments physiques ;
- éléments de modélisation explicites pour représenter des entrées/sorties physiques ;
- représentation des attributs continus (fluides, énergie..) ;
- sémantiques limitées pour spécifier des événements et coupler l'événement avec l'action.

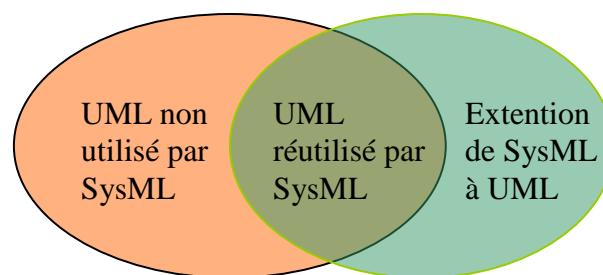


Figure 0.3 : SysML en tant que profil UML⁷⁶.

L'OMG L'OMG a émis une demande de proposition (*Request For Proposal* ou RFP) pour un profil UML destiné à couvrir les tâches de l'ingénierie système [OMG_UML-RFP-2003], et SysML⁷⁷ a été ensuite proposé par l'INCOSE⁷⁸. SysML a été conçu comme une extension à UML pour minimiser les difficultés de son implémentation dans les outils UML existants. Sa spécification a été écrite de manière à être abordable contrairement à celle d'UML pour ne pas constituer un frein à son adoption dans le monde de l'ingénierie système [Turki-2008].

⁷⁶ Extrait de [Roques-2009] p.15.

⁷⁷ Toutes les informations officielles sur le langage SysML se trouvent sur : <http://www.omg.sysml.org/>

⁷⁸ INCOSE : International Council on Systems Engineering. Voir : <http://www.incose.org/>

L'OMG a soutenu SysML comme un nouveau langage différent d'UML, tout en profitant de sa filiation directe (voir Figure 3).

SysML s'articule autour de neuf types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système (voir Figure 4). Ces types de diagrammes sont répartis par l'OMG en trois grands groupes :

- quatre diagrammes comportementaux :
 1. diagramme d'activité (montre l'enchaînement des actions et décisions au sein d'une activité complexe) ;
 2. diagramme de séquence (montre la séquence verticale des messages passés entre blocs au sein d'une interaction) ;
 3. diagramme d'états (montre les différents états et transitions possibles des blocs dynamiques) ;
 4. diagramme de cas d'utilisation (montre les interactions fonctionnelles entre les acteurs et le système à l'étude) ;
- un diagramme transverse : le diagramme d'exigences (montre les exigences du système et leurs relations) ;
- quatre diagrammes structurels :
 1. diagramme de définition de blocs (montre les briques de base statiques : blocs, compositions, associations, attributs, opérations, généralisations, etc.) ;
 2. diagramme de bloc interne (montre l'organisation interne d'un élément statique complexe) ;
 3. diagramme paramétrique (représente les contraintes du système, les équations qui le régissent) ;
 4. diagramme de packages (montre l'organisation logique du modèle et les relations entre packages).

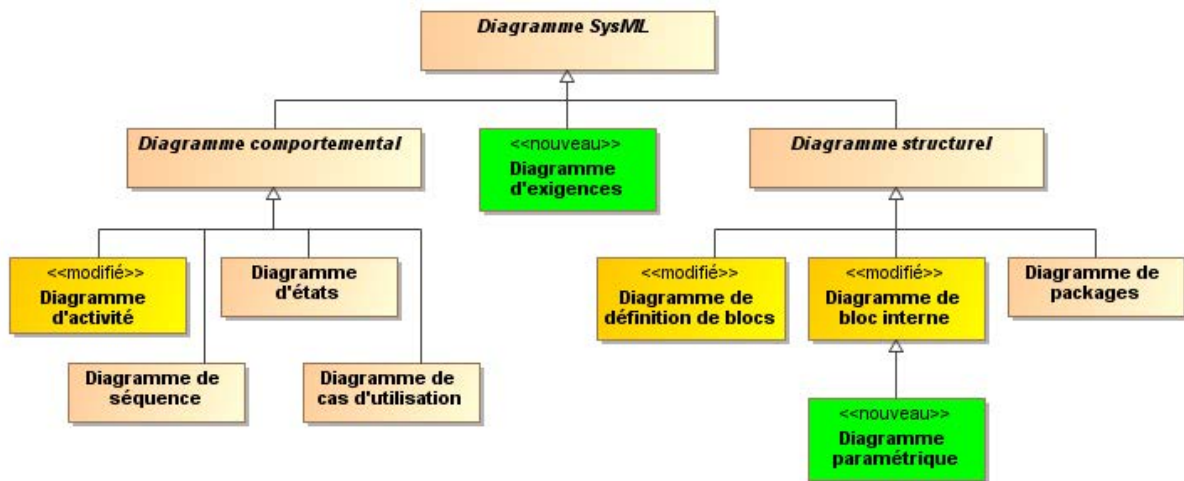


Figure 0.4 : Les 9 types de diagrammes SysML⁷⁹.

Ce langage comble les lacunes d'UML citées au début de cette section.

3. Méthodologies de Conception des systèmes

[Turki-2008] définit un système comme : « un ensemble de composants qui collaborent à la réalisation d'un ensemble de tâches en vue de fournir un ensemble de services, cet ensemble est soumis à un environnement donné et interagit ainsi avec un sous-ensemble des éléments de cet environnement » (voir Figure 5). Par contre, une méthodologie est définie comme « l'ensemble des processus, méthodes et outils pouvant être appliquée à un domaine particulier » [Estefan-2008].

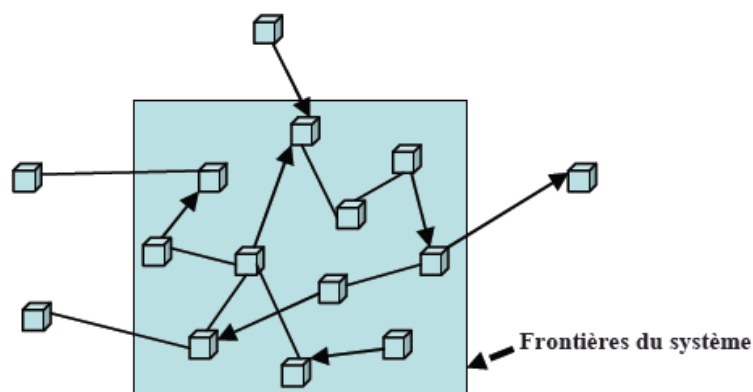


Figure 0.5 : Définition du système⁸⁰.

Avec la croissance continue de la complexité des systèmes (réels ou virtuels) modélisés, les techniques de programmation orientée objet n'étaient plus suffisantes pour construire des solutions stables répondant aux problèmes décrits dans les systèmes complexes. Pour cela,

⁷⁹ Extrait de [Roques-2009] p.12.

⁸⁰ Extrait de [Turki-2008] p.24.

une méthodologie d'analyse et de conception d'un système quelconque devrait être adoptée afin d'organiser la réalisation de ce système. Cette méthodologie devrait également présenter des aspects correspondants aux principes présentés dans les langages de programmation orientée objet.

Cette vision a provoqué la création de différentes méthodologie d'analyse et de conception orientée objet « OOADM » (*Object-Oriented Analysis and Design Methodology*) dont le but est de développer un modèle correspondant à un système. En effet, plusieurs modèles de conception peuvent correspondre à un modèle d'analyse. Pour écrire des différents modèles, différentes méthodes ont été mises au point mais elles ne permettaient de modéliser que certains types d'applications et se trouvaient limitées dans d'autres contextes. Plusieurs méthodologies ont été présentées dans la première partie de la décennie 1990, parmi les quelles nous comptons : OODA⁸¹, OOA/OOD⁸², OOAD⁸³, OMT⁸⁴, OOSA⁸⁵, DOOS⁸⁶. Une comparaison entre ces méthodologies, pour plus de détails, est menée dans [Hong-1993].

Néanmoins, la complexité de système modélisé ne se limite pas à sa grande taille et au nombre de ses sous-systèmes. D'une part, l'interdisciplinarité des sous-systèmes imposent des freins, sur la conception du système global, issus des aspects de types et de natures différents et possiblement interdépendants des ces sous-systèmes. D'une autre part, le choix de la plate-forme où s'appliquera un modèle impose des contraintes sur le processus de conception dès son début, et l'application du modèle développé sous des telles contraintes se limitera donc à certaine(s) plate-forme(s). Ces difficultés ont provoqué le développement des démarches méthodologiques pour les surmonter, il s'agit de : l'ingénierie système (IS : *Systems Engineering* « SE ») et l'architecture guidée par des modèles (MDA : *Model Driven Architecture*).

3.1. Le processus d'ingénierie système

L'ingénierie système « IS » est une démarche méthodologique pour maîtriser la conception des systèmes et produits complexes, il s'agit d'une approche interdisciplinaire rassemblant tous les efforts techniques pour faire évoluer et vérifier un ensemble intégré de systèmes, de

⁸¹ OODA : Object Oriented Design with Applications, [Booch-1991].

⁸² OOA/OOD : Object Oriented Analysis and Object Oriented Design, [Coad-1991a, Coad-1991b].

⁸³ OOAD : Object Oriented Analysis and Design, [Martin-1992].

⁸⁴ OMT : Object Modeling Technique, [Rumbaugh-1991].

⁸⁵ OOSA : Object Oriented Systems Analysis, [Shlaer-1988].

⁸⁶ DOOS : Designing Object Oriented Software, [Wirfs-Brock-1990].

gens, de produits et de solutions de processus de manière équilibrée au fil du cycle de vie pour satisfaire aux besoins client [Roques-2009].

L'ingénierie des systèmes se focalise sur la définition des besoins du client et des exigences fonctionnelles, détectés tôt dans le cycle de vie, en documentant les exigences, puis en poursuivant avec la synthèse de la conception et la validation du système. Une méthodologie supportant une démarche d'ingénierie système (Figure 6) distingue les termes suivants [Estefan-2008] :

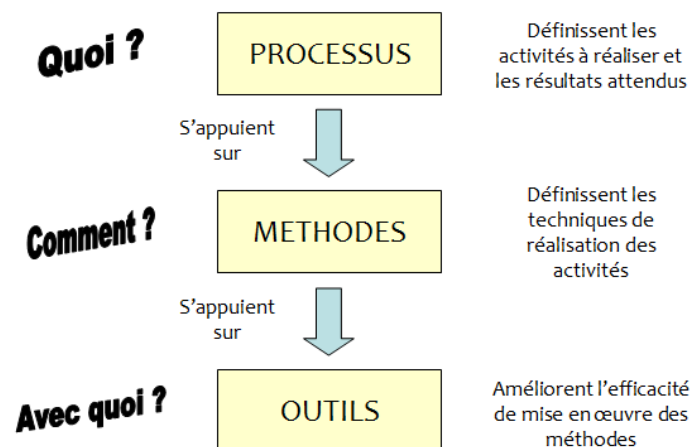


Figure 0.6 : La démarche de l'ingénierie système⁸⁷.

- un processus : est une séquence logique de tâches réalisées pour atteindre un objectif particulier. Un processus définit ce qui est à faire sans préciser le « comment faire ». La structure d'un processus peut se présenter sous différents niveaux d'agrégation pour permettre les analyses et répondre aux différents besoins d'aide à la décision,
- une méthode : est attachée aux techniques de réalisation d'une tâche. Elle définit le « comment faire » de chaque tâche (dans ce contexte méthode, technique, pratique et procédure sont souvent interchangeables). A tous niveaux, les tâches de processus sont réalisées en utilisant des méthodes. Cependant, chaque méthode est aussi, elle-même, un processus avec sa séquence de tâches à réaliser selon des méthodes particulières. En d'autres mots, le « comment » d'un niveau d'abstraction devient le « quoi » du niveau inférieur,
- et l'outil : est attaché à une méthode particulière, pour augmenter l'efficacité de réalisation de la tâche. Le rôle d'un outil est de faciliter le « comment »,
- une méthodologie : est définie comme une collection de processus, de méthodes et

⁸⁷ Extrait de [Roques-2009] p.2.

d'outils reliés. Une méthodologie est essentiellement une « recette » et peut être comprise comme l'application de processus, méthodes et outils relatifs à une classe de problèmes qui ont quelque chose en commun.

Des recommandations (approches abstraites selon [Clarkson-2005]) ont été définies pour donner un cadre global à l'ingénierie système. Par exemple, la norme ANSI/EIA 632 [ANSI/EIA_632-2003] qui a été développée par la EIA (*Electronic Industries Alliance*) et adoptée en 1999 par l'ANSI (*American National Standards Institute*) décrit le processus d'ingénierie système comme illustre la Figure 7 :

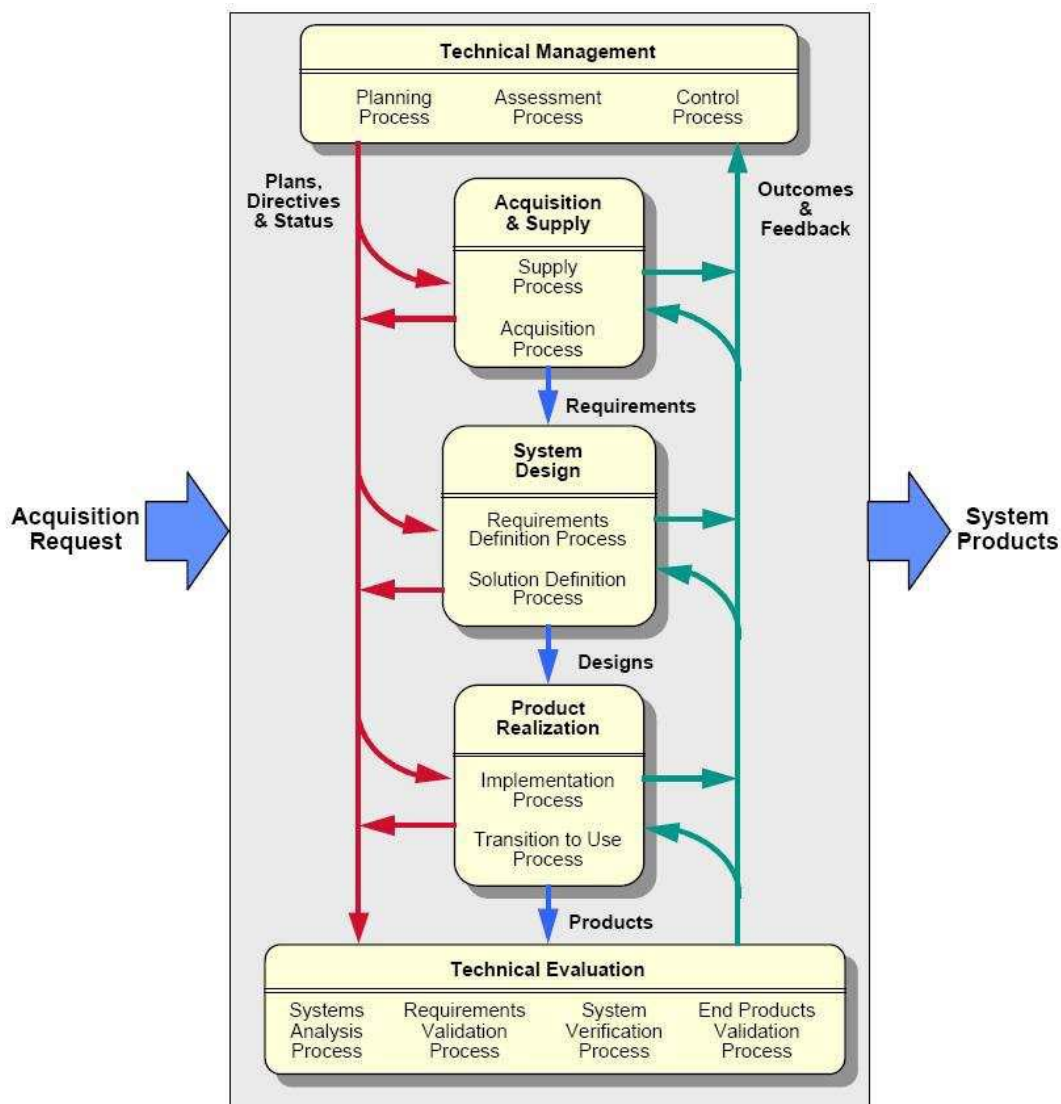


Figure 0.7 : Processus d'ingénierie système ANSI/EIA 632.

D'autres processus ont été identifiés dans plusieurs domaines, tels que SIMILAR [Bahill-1998] (voir section 3.1.1-1 de cette annexe), ESDL [Berger-2002] (voir section 3.1.1-2 de cette annexe), HARMONY (voir section 3.1.1-3 de cette annexe).

L'approche interdisciplinaire de l'ingénierie système est intrinsèquement complexe puisque les interactions avec l'environnement et même les interactions entre toutes les composantes du système ne sont pas toujours bien définies ou bien comprises (du moins a priori). Définir et caractériser de tels systèmes et sous-systèmes ainsi que leurs interactions sont un des objectifs de l'ingénierie système. En faisant cela, le pont entre exigences informelles des utilisateurs et les spécifications techniques, qu'un ingénieur peut implémenter, pourra être franchi. Selon [Rochet-2008], l'IS cherche à gérer cette problématique par une approche holistique et interdisciplinaire :

- **Vue holistique** : L'ingénierie système se fonde sur la définition des besoins du client et l'identification des fonctionnalités du système au plus tôt dans le cycle de développement (traitement et documentation des exigences). Elle réalise alors la conception et la validation du système en considérant le problème sur l'ensemble de son cycle de vie. Le modèle des processus d'ingénierie est au cœur de l'ingénierie système. Ce processus d'ingénierie a pour but d'organiser l'effort technique et organisationnel dans tout le cycle de vie. Il couvre la définition du concept et la production et va, dans certains cas, jusqu'à la mise à disposition et au retrait de service du système.
- **Couverture interdisciplinaire** : Le développement de systèmes demande souvent les contributions de diverses disciplines techniques. Pour obtenir une telle expertise, un ingénieur système est souvent un ingénieur traditionnel possédant une expertise dans un domaine et une connaissance des autres domaines incluant la gestion et les processus d'affaire, ce qui aide à l'intégration des sous-systèmes et à la validation des exigences. En donnant une vue système (holistique) à l'effort de développement, l'IS aide à intégrer toutes les contributions techniques dans un effort commun coordonné.

3.1.1. Exemples de processus d'ingénierie système

- SIMILAR [Bahill-1998]

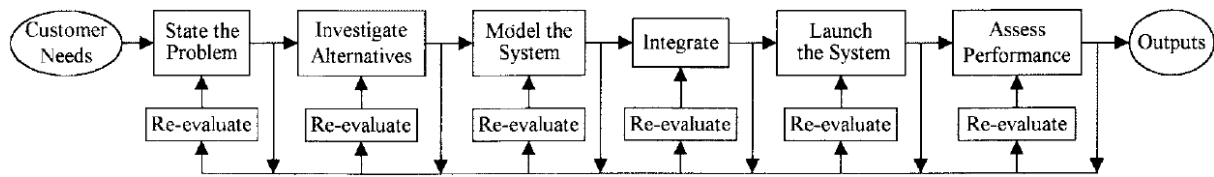


Figure 0.8 : Le processus SIMILAR⁸⁸.

SIMILAR (voir Figure 8) est un processus qui définit sept activités principales que sont la définition du problème, la découverte de solution alternative, la modélisation du système, l'intégration, la mise en service, la vérification des performances et la réévaluation de décisions prises au préalable suivant le résultat de cette vérification. Il est important de noter que l'exécution de ces tâches n'est pas séquentielle et qu'aucun ordre n'est établi a priori. L'idée générale de ce processus est de réduire les risques pris en systématisant les mécanismes de spécification-validation.

- ESDL [Berger-2002]

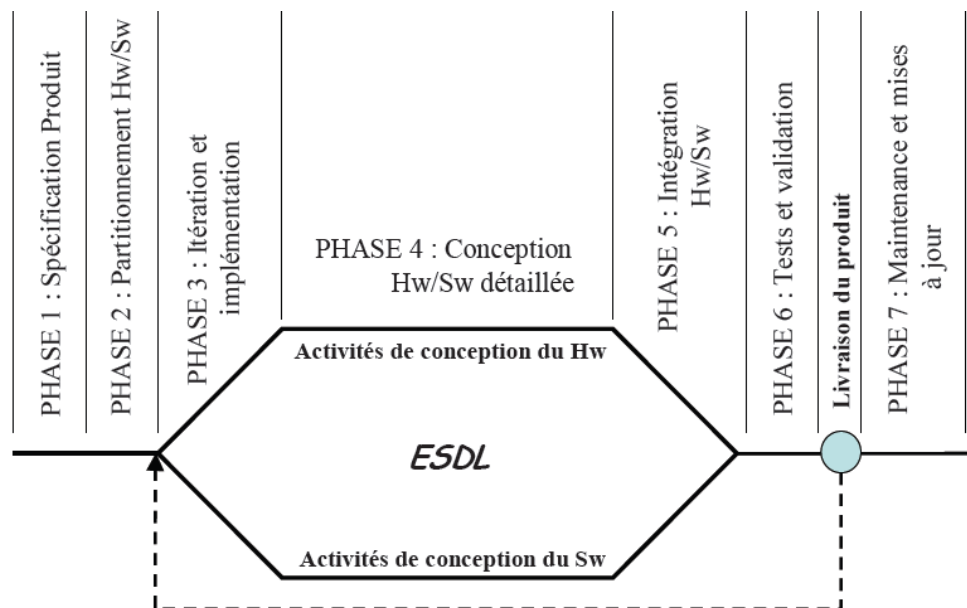


Figure 0.9 : Le processus ESDL⁸⁹.

Le processus ESDL (Embedded Systems Design Lifecycle) se base sur le cycle de conception illustré dans la Figure 9. La première phase consiste en la compréhension et la définition des besoins du client. Dans la deuxième phase on choisit quels sont les problèmes qui seront résolus par le matériel et ceux qui seront résolus par la partie logicielle. Ensuite

⁸⁸ Extrait de [Bahill-1998].

⁸⁹ Extrait de [Berger-2002] – chapitre 1, p.1.

deux activités concurrentes se déroulent pour la conception du matériel et pour la conception de la partie logicielle. Ensuite, il y a une phase d'intégration, une phase de tests et de validation qui peut nécessiter une réitération du processus et finalement le produit est livré puis maintenu.

- **HARMONY [Estefan-2008]**

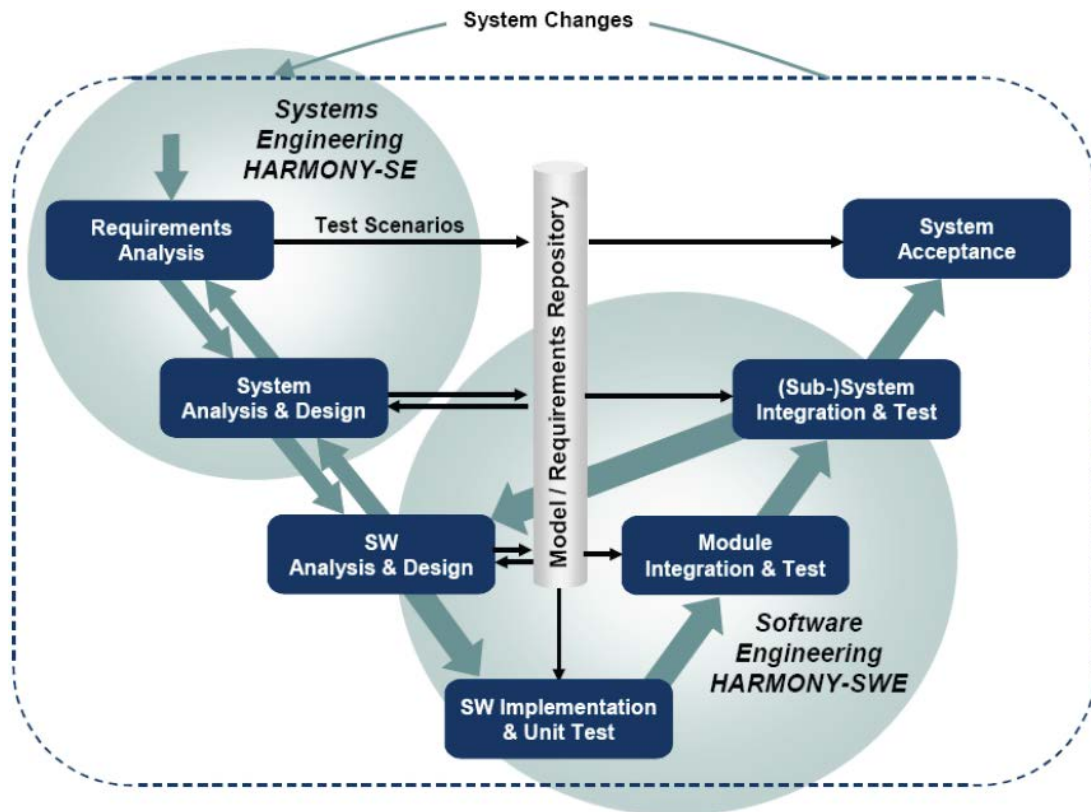


Figure 0.10 : Le processus HARMONY⁹⁰.

Harmony (voir Figure 10) est un processus d'ingénierie système destiné à des systèmes à forte composante logicielle. Harmony s'organise autour des exigences, il commence par identifier ces exigences et en déduire les fonctionnalités requises du système, ensuite par concevoir le système, définir ses modes de fonctionnement et ses états, ensuite concevoir et implémenter la partie logicielle puis l'intégrer en allouant le logiciel à l'architecture physique et finalement le valider.

3.2. L'approche de modélisation MDA

L'architecture guidée par les modèles MDA est une démarche de réalisation de systèmes logiciels, proposée et soutenue par l'OMG [OMG_MDA-2003]. La MDA prend en compte les contraintes qui empêchent les méthodologies de conception classiques d'obtenir un degré de

⁹⁰ Extrait de [Estefan-2008], p.23.

généricité important, pour proposer des plates-formes et des langages de programmation à adopter par le concepteur. Une plate-forme est en général le conteneur qui va accueillir les éléments développés dans un langage donné. Les éléments déployés peuvent donc vivre et communiquer entre eux dans cette plate-forme qui est constituée d'un ensemble de services. Cet ensemble de services est connu comme « *Middleware* ». Un niveau d'abstraction supplémentaire peut être introduit en spécifiant des éléments collaboratifs et la manière dont ils communiquent indépendamment d'une technologie de plate-forme donnée.

Dans une démarche d'ingénierie système, l'intérêt du MDA est de transformer un modèle abstrait en un modèle spécifique (à une technologie ou à un domaine donné), grâce à un ensemble de correspondances entre les 2 formalismes de représentation. Le concepteur peut soit travailler directement sur le modèle obtenu, soit appliquer une nouvelle transformation pour obtenir un modèle encore plus spécifique. La MDA fournit une succession de modèles :

- le CIM (*Computational Independent Model*) : il se concentre sur les besoins auxquels doit répondre le système plutôt que sur les moyens qui seront mis en œuvre pour y parvenir, il s'agit donc de modéliser les spécifications. Le CIM centralise les informations relatives aux services que devra implémenter le système et implique la participation des futurs utilisateurs du système pour la spécification des besoins et des cas d'utilisation. Ce modèle est également un moyen de communication entre les personnes qui vont concevoir le système, et les personnes exprimant des besoins sans la moindre idée de la façon dont il faut les implémenter ;
- le PIM (*Platform Independent Model*) : il a pour but de décrire des objets et leurs interactions dans un formalisme indépendant d'une technologie donnée, il s'agit donc de décrire les scénarios de fonctionnement conduisant à répondre aux exigences du CIM. Les comportements internes des objets avec la description des méthodes qui leur sont associées sont également représentés dans ce modèle ;
- le PSM (*Platform Specific Model*) : il représente les modèles spécifiques à obtenir en phase de réalisation concrète du système. L'obtention du PSM suppose qu'une technologie cible ait été choisie et que la correspondance entre les concepts généraux du PIM et ceux utilisés dans le PSM aient été établis dans un profil spécifique à la plate-forme cible ;
- le PM (*Platform Model*) : Il est fourni par le distributeur de la plate-forme afin de représenter les mécanismes de fonctionnement ainsi que les services de la plate-forme

de façon précise, il s'agit donc d'un ensemble de modèles qui caractérisent une plate-forme. Ces spécifications serviront par la suite durant les transformations en fournissant un ensemble de concepts techniques pour les correspondances avec les concepts de niveau plus abstrait.

Le passage du *PIM* au *PSM* s'effectue durant une étape appelée *mapping*. Les éléments compris dans le *PIM* sont alors analysés de façon automatique et les concepts précédemment définis, à l'aide de types génériques et indépendants d'une plate-forme, sont remplacés par des éléments spécifiques à la plate-forme choisie (Voir Figure 11).

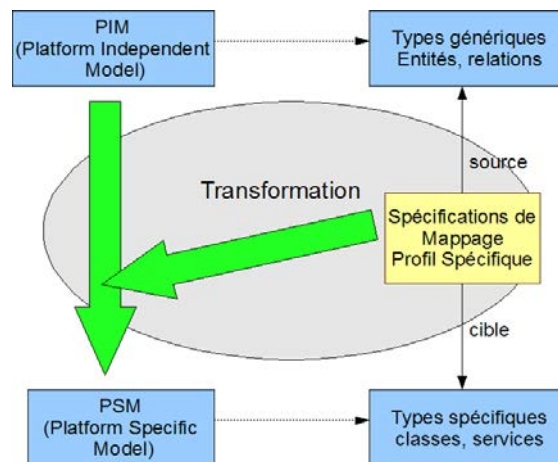


Figure 0.11 : Transformation de modèle selon le paradigme MDA⁹¹.

Le langage UML est la pierre angulaire de l'approche MDA de l'OMG, utilisé pour la description graphique de tous les aspects d'un système et cela sur différents niveaux de détail. Le stockage des spécifications au sein de plusieurs modèles avec une spécificité croissante garantit un maximum d'indépendance vis-à-vis d'une technologie particulière. Ainsi lorsqu'une technologie en remplace une autre, une grande partie du modèle reste inchangé.

⁹¹ Extrait de [Chiron-2007], p.84.

Annexe C : Sujet sur le magasin vertical

1. Présentation du TP

Objectif

L'objectif du TP est de programmer un système séquentiel. La programmation se fera à l'aide d'un logiciel industriel : PL7Pro de la société Schneider. A travers ce TP l'élève apprendra à utiliser le Grafset et le langage à contact pour programmer une application utilisant différents types de variables : internes, externes (câblées), binaires, mots, tableaux, adresses mémoires. L'élève découvrira également l'automate industriel TSX 37 (ou TSX Micro).

Organisation du sujet

Le sujet comprend trois parties :

- une partie « *présentation du TP* » (celle-ci) qui vous présente l'objectif à atteindre, le sujet, le travail à faire et les documents à rendre ;
- une partie « *le système* » qui vous présente la partie opérative et la partie commande du système ;
- une partie « *travail à faire* » qui détaille le cahier des charges et qui pose les questions auxquelles vous devrez répondre. Cette partie comprend tout d'abord deux exercices introductifs puis l'étude du système qui est présentée de façon progressive.

Organisation du travail

Les deux exercices introductifs **sont absolument à faire avant la séance de TP**. Dès votre arrivée en salle de TP, vous les utiliserez pour prendre en main le logiciel PL7Pro. Préparez également en détail le reste du TP avant la séance. Si le travail n'est pas sérieusement préparé, vous ne pourrez pas finir le TP...

Documents à rendre

Une semaine après la séance de TP, vous devez rendre :

- le sujet du TP (n'hésitez pas à corriger les fautes ou à mettre des remarques sur les manques ou incohérences du sujet) ;
- le compte-rendu de TP avec l'ensemble des réponses aux questions. Les questions doivent être traitées dans l'ordre.

Tous ces documents vous seront rendus une fois corrigés.

2. Le système

La partie opérative

La partie opérative du système que vous allez utiliser est un magasin permettant de stocker des pièces qui sont rentrées et sorties manuellement par une porte située à la base du magasin. Le principe de fonctionnement de ce magasin est simple : un moteur est situé à la base de la face arrière de ce magasin, par l'intermédiaire d'un réducteur il entraîne en rotation une poulie crantée. Les nacelles qui portent les pièces sont fixées sur cette poulie, et se déplacent donc dans le magasin. Pour rentrer (respectivement sortir) une pièce dans le magasin, il faut donc positionner une nacelle vide (respectivement pleine) en face de la porte, ouvrir la porte, et placer (respectivement enlever) la pièce sur la nacelle.

La partie commande

La partie commande est basée sur un automate programmable industriel. L'automate utilisé sur ce système est un TSX 37. Il est équipé de racks optionnels qui lui permettent de disposer d'entrées et de sorties TOUT OU RIEN, d'entrées et de sorties ANALOGIQUES et d'un module de COMPTAGE. L'automate peut être programmé par différents langages qui peuvent tous être utilisés au sein d'une même application. Le langage permettant de construire l'ossature séquentielle du programme est basé sur le **Grafcet**. Les transitions et les traitements sont programmés en **listes d'instructions (IL)**, en **langage à contacts (LD)** ou en **langage littéral structuré (ST)**. Vous utiliserez le langage à contacts.

Le logiciel PL7Pro permet de construire complètement l'application, c'est-à-dire de configurer l'automate, d'écrire les programmes de les tester, les transférer et les exécuter et même de modifier les programmes en cours d'exécution dans l'automate.

L'automate MICRO utilisé est un automate modulaire. Les modules sont des éléments optionnels de l'automate : ils sont montés et organisés en fonction de l'application commandée. Ceci permet d'adapter précisément la commande au besoin. L'automate installé pour cette application comprend six modules :

- le module 1 est composé d'entrées binaires,
- le module 2 (indissociable du module 1) est composé de sorties binaires,
- le module 3 est aussi un module d'entrées binaires,
- le module 4 est un module de comptage (doit être configuré avec la fonction comptage/décomptage, en tâche MAST, sur l'événement 1),
- le module 5 est composé d'entrées analogiques,
- le module 6 est composé de sorties analogiques.

Tableau 1 :

Adresse module	Famille	Référence
00	Processeur	TSX 3721
01-02	Tout ou rien	TSX DMZ 28DR
03	Tout ou rien	TSX DEZ 12D2
04	Comptage	TSX CTZ 1A
05	Analogique	TSX AEZ 801
06	Analogique	TSX ASZ 401

Tableau 0.1 : Références des modules.

Le module 0 n'est pas optionnel : il contient le processeur.

Les tableaux suivants indiquent module par module la fonction des entrées et sorties. Lors de la programmation vous utiliserez les repères de ces entrées et sorties.

Repère	Symbole	Fonction
%I1.0	Clavier_a	Ces entrées correspondent au codage en binaire (BCD) des touches du clavier. %I1.0 est le bit de poids faible, %I1.4 le bit de poids fort.
%I1.1	Clavier_b	
%I1.2	Clavier_c	
%I1.3	Clavier_d	
%I1.4	Strobe_clavier	une touche du clavier activée
%I1.5	Bp_plus	bouton plus devant magasin
%I1.6	Bp_moins	bouton moins devant magasin
%I1.7	Bp_acquit	bouton « OK » sur l'armoire de commande

%I1.8	Manu	sélecteur « manu-auto » sur l'armoire de com
%I1.9	Balancement_av	balancement avant des nacelles
%I1.10	Balancement_ar	balancement arrière des nacelles
%I1.11	Nacelle_pleine	présence pièce dans nacelle devant porte
%I1.13	Parois_ouverte	une des cinq parois (cotés et haut) est ouverte
%I1.14	Porte_haute	porte en position haute
%I1.15	Porte_basse	porte ne position basse

Tableau 0.2 : Module TSX DMZ 28DR (position 1) : 16 entrées binaires 24VCC.

Repère	Symbole	Fonction
%Q2.0	Pwr_var	auto maintien de l'alimentation du variateur
%Q2.1	Pwr_mot	alimentation du moteur (nécessite %QW6.0)
%Q2.2	Frein	
%Q2.3	Voyant_def	voyant du bouton « OK » de l'armoire
%Q2.4	Voyant_rouge	voyant rouge au-dessus du clavier
%Q2.5	Voyant_orange	voyant orange au-dessus du clavier
%Q2.6	Voyant_vert	voyant vert au-dessus du clavier
%Q2.7	Eclairage_porte	
%Q2.9	Aimant	aimant de maintien de la porte en haut

Tableau 0.3 : Module TSX DMZ 28DR (position 2) : 16 sorties binaires à relais.

Repère	Symbole	Fonction
%I3.0	Prog_2	prog1 (faux) ou prog2 (vrai) devant magasin
%I3.1	Km1_pwr_var	validation marche
%I3.2	Km2_pwr_mot	nacelles en rotation
%I3.3	Km3_frein	coupure du freinage
%I3.4	Thermique	déclenchement thermique
%I3.5	Pom	nacelle 1 en position devant la porte
%I3.6	Nacelle_devant_porte	une nacelle devant la porte
%I3.7	Mode_cpe	mode consigne de position
%I3.8	Mode_int	mode consigne de vitesse
%I3.11	Pres_24v	présence du 24 V d'alimentation des entrées

Tableau 0.4 : Module TSX DEZ 12D2 (position 3) : 12 entrées binaires 24VCC.

Repère	Symbole	Fonction
%IW5.0	Cve	entrée analogique voie 0
%IW5.1	Cpe	entrée analogique voie 1
%IW5.7	Wm	Vitesse mesurée

Tableau 0.5 : Module TSX AEZ 801 position 5) : 8 entrées analogiques +/- 10V.

Repère	Symbole	Fonction
%QW6.0	C_var	Consigne de vitesse
%QW6.1	Ev	Écart de vitesse = %QW6.0 - %IW5.7
%QW6.2	Mp	Mesure de position en format réel (à calculer)
%QW6.3	Ep	Écart de position (à calculer)

Tableau 0.6 : Module TSX ASZ 401 (position 6) : 4 sorties analogiques +/- 10V.

Utilisation de PL7Pro

En arrivant en séance de TP et en vous aidant de l'annexe, utilisez le logiciel PL7Pro pour programmer et tester les deux exercices précédents. En cas de problème d'utilisation du logiciel, n'hésitez pas à demander à votre enseignant. Ces exercices doivent être testés très rapidement.

Étude du système

Vous devrez maintenant écrire un programme qui permettra la mise en œuvre complète du magasin automatisé. Cette mise en œuvre comprend trois phases de fonctionnement. Ces phases sont successivement l'initialisation du système, la mise à jour des données et le chargement ou déchargement de pièces du magasin.

L'application finale devra donc comporter quatre graficets :

- un graficet appelé « Maître » sera utilisé pour autoriser le passage d'une phase à une autre ;
- un graficet appelé « P.O.M. » décrira la commande durant la phase d'initialisation du système ;
- un graficet appelé « Mise à jour » décrira la commande durant la phase de mise à jour des données ;
- un graficet appelé « Normal » décrira la commande durant la phase de chargement ou déchargement du magasin.

Chacune de ces phases est décrite dans les chapitres suivant qui vous permettront de prendre en main progressivement le logiciel et le système. Faites attention au temps passé sur chacune des parties : elles sont de plus en plus longues, évitez donc de perdre du temps...

Initialisation du système

Sécurités câblées

Les sécurités du système sont des sécurités câblées. Ces sécurités interdisent par défaut le démarrage du moteur et du variateur. Ainsi si un programme active la rotation du moteur, celui-ci ne pourra tourner si ces sécurités n'ont pas été au préalable désactivées par l'opérateur.

Fonctionnement :

L'alimentation électrique du variateur est conditionnée par la fermeture d'un certain nombre d'interrupteurs. Ces interrupteurs sont les arrêts d'urgence et le bouton de puissance (pour plus de détails, voir les schémas électriques et les schémas d'implantation). La mise en route du moteur est conditionnée par l'alimentation du variateur, l'alimentation du moteur et l'envoi d'une consigne de vitesse au moteur. Pour démarrer l'alimentation du variateur, il faut appuyer sur le bouton de puissance. Pour maintenir l'alimentation du variateur il faut mettre à 1 la sortie %Q2.0 de l'automate.

Prise d'origine machine

Le détecteur de nacelle ne permet de détecter que la présence d'une nacelle mais pas son numéro. Par contre un capteur spécifique permet de détecter le plot métallique de la nacelle 1. Au démarrage, l'automate n'a donc aucune information lui permettant de déterminer quel est le numéro de la nacelle qui est en face de la porte. Il est donc nécessaire de mettre en route le moteur pour que les nacelles se déplacent, et ainsi amener la nacelle 1 en face de ce capteur. A partir de ce moment-là, l'automate pourra à chaque instant déterminer le numéro de la palette en face de la porte, en fonction du sens de rotation et du nombre de nacelles passées devant le détecteur.

Programmation

Cette application est déjà implantée dans le fichier nommé « base ».

3. Scénario pédagogique de la première session

L'objectif de ce scénario est de guider les étudiants pendant la programmation du système de magasin vertical en état de fonctionnement normal. Le chargement/déchargement des pièces métalliques se fait manuellement.

Exercice introductif

L'objectif de cet exercice est de vous apprendre la programmation sous PL7Pro, pas de vous poser des problèmes de logique. Ces exercices ne vous feront utiliser que le clavier numérique, les boutons + et -, et les trois voyants vert, orange et rouge de la partie opérative.

Cahier des charges

Il s'agit maintenant d'utiliser le clavier pour programmer l'allumage d'un des trois voyants pendant une durée qui est aussi indiquée par l'intermédiaire du clavier :

- au départ les trois voyants sont éteints,
- j'appuie sur une première touche pour indiquer quel voyant devra s'allumer : 1 pour le vert, 2 pour l'orange, 3 pour le rouge,
- j'appuie ensuite sur une touche pour indiquer combien de secondes devra s'allumer ce voyant : entre 1 et 9 secondes suivant la touche,
- dès que j'ai appuyé sur la seconde touche, le voyant s'allume pendant le temps voulu,
- je peux alors recommencer le cycle.

Les sorties du clavier sont quatre sorties binaires `clavier_a`, `clavier_b`, `clavier_c` et `clavier_d` reliées aux quatre entrées %I1.0, %I1.1, %I1.2 et %I1.3 de l'automate. Pour connaître la valeur numérique de la touche sélectionnée, il est donc nécessaire de traduire ces informations binaires en une information numérique.

Touche	0	1	2	3	4	5	6	7	8	9	A	B
Clavier_a (%I1.0)	0	1	0	1	0	1	0	1	0	1	0	1
Clavier_b (%I1.1)	0	0	1	1	0	0	1	1	0	0	1	1
Clavier_c (%I1.2)	0	0	0	0	1	1	1	1	0	0	0	0
Clavier_d (%I1.3)	0	0	0	0	0	0	0	0	1	1	1	1

Tableau 0.7 : Codage du clavier.

Question 1 : Écrivez le programme PL7Pro décrivant le cahier des charges. Vous utiliserez le grafset pour décrire la partie séquentielle du cahier des charges et le langage à contacts pour la partie combinatoire.

La mise à jour des données

La mise à jour des données devra se faire par l'intermédiaire d'un grafset nommé « Mise à jour ». Ce grafset aura à gérer les tâches suivantes :

- rotation des nacelles jusqu'à une nacelle pleine,

- allumer le voyant rouge,
- mémoriser la première touche saisie au clavier dans une variable interne,
- allumer le voyant orange (éteindre le rouge),
- mémoriser la seconde touche saisie au clavier dans une variable interne,
- recalculer le numéro de la pièce à partir de ces deux chiffres (éteindre le voyant orange),
- affecter cette valeur à la variable correspondant à cette nacelle,
- recommencer jusqu'à ce que toutes les nacelles pleines soient passées.

A) Utilisation du clavier

Une fois la prise d'origine effectuée, il est nécessaire de mettre à jour la base de données du système. En effet, il est possible que le système démarre avec des nacelles contenant déjà des pièces. Les pièces sont identifiées par un numéro. Ce numéro doit être fourni par le magasinier à l'aide du clavier numérique. Ce numéro comporte deux chiffres. Le premier chiffre est saisi lorsque le voyant vert est allumé, il représente le chiffre des dizaines. Le second est saisi lorsque le voyant orange est allumé, il représente le chiffre des unités.

Question 2 : Comment reconstituer le numéro à deux chiffres à partir des deux appuis successifs sur les touches ?

B) Mémorisation des informations

Le numéro de pièce doit être stocké en mémoire dans un emplacement correspondant à la nacelle présente devant la porte. Une solution simple est d'utiliser autant de mots mémoire que de nacelles. Il est indispensable d'utiliser des mots d'adresses consécutives afin quelles constituent un tableau de mots. Ceci permettra par la suite d'utiliser des fonctions sur tableaux de mots.

Question 3 : Comment changer l'adresse mémoire dans laquelle sera mémorisée le numéro de la pièce en fonction du numéro de nacelle présente devant la pièce ?

C) Le grafctet

Le grafctet doit donc permettre l'exécution de toutes les tâches présentées sur la page précédente. L'évolution de ce grafctet peut commencer dès que la phase d'initialisation est terminée, sans intervention du magasinier. Lorsqu'un numéro de pièce a été mémorisé, l'opérateur doit appuyer sur la touche « + » pour passer à la recherche de la nacelle pleine suivante. A la fin de la phase de mise à jour, la seule étape active de ce grafctet doit être son étape initiale et la nacelle présente en face de la porte doit être la nacelle n°1.

Question 4 : Écrire le grafctet (sur une nouvelle page), les transitions et les actions correspondant à cette phase de mise à jour. Avant de tester ce programme, répondez aux deux questions suivantes.

Question 5 : Modifiez le grafcet maître pour autoriser le passage de la phase « P.O.M. » à la phase « Mise à jour ».

4. Scénario pédagogique de la deuxième session

L'objectif de ce scénario est de guider les étudiants pendant la programmation du système de magasin vertical avec le translateur pour permettre le fonctionnement normal du système où le chargement/déchargement des pièces métalliques se fait automatiquement à l'aide du translateur.

Vous devrez maintenant écrire un programme qui permettra la mise en œuvre complète du magasin automatisé en le synchronisant avec le translateur. Cette mise en œuvre comprend trois phases de fonctionnement. Ces phases sont successivement l'initialisation du système, la mise à jour des données et le chargement ou déchargement de pièces du magasin.

L'application finale devra donc comporter quatre grafcets :

- un grafcet appelé « Maître » sera utilisé pour autoriser le passage d'une phase à une autre ;
- un grafcet appelé « P.O.M. » décrira la commande durant la phase d'initialisation du système ;
- un grafcet appelé « Mise à jour » décrira la commande durant la phase de mise à jour des données ;
- un grafcet appelé « Normal » décrira la commande durant la phase de chargement ou déchargement du magasin à l'aide du translateur.

Annexe D : TP sur le Transfert libre en GMC

Objectifs : L'objectif du travail proposé est de mettre en application les concepts vus dans le cours Systèmes Automatisés de Production en 3ème année GMC et lors du projet effectué en TD. Le travail à réaliser consiste à mettre en place, à partir d'un cahier des charges spécifié, la commande d'un automate pour la gestion d'un poste de travail. Chaque poste de travail comporte trois zones : Une zone d'entrée, une zone de travail, et une zone de sortie. L'étude est divisée en 6 étapes : une étape pour chaque zone, et les trois dernières étapes pour les capacités de stockage et gestion de la sécurité. La partie concernant la gestion de la zone d'entrée est partiellement donnée, afin de permettre une prise en main du logiciel Unity Pro.

Compte-rendu : Un seul compte-rendu pour chaque groupe est demandé après la dernière séance. Ce rapport doit contenir l'ensemble du travail réalisé sur les 3 séances. Expliquer l'adéquation entre toute solution proposée et le cahier des charges. Si le cahier des charges a été modifiée, ou si d'autres contraintes ont été ajoutées, le spécifier dans le rapport. Mettre en évidence les limites de chacune de vos solutions.

1 Architecture du système étudié

1.1 Le transfert libre

Le système de production à l'Atelier Inter-établissement de Productique (AIP) représente une installation industrielle de type « transfert libre » qui permet la circulation de palettes grâce à un convoyeur central à bande et d'effectuer au niveau de six postes de travail des opérations différentes (voir Figure 1). Les pièces sont distribuées aux postes de travail par le convoyeur principal par orientation des palettes sur des boucles de dérivation associées à chacune des six zones de travail. Les palettes transportent des pièces destinées à subir diverses opérations selon un cycle de production (gamme de fabrication). Les informations, contenues dans une étiquette électromagnétique située sur chaque palette, permettent de renseigner les postes sur la nature du traitement à exercer sur leur chargement. Lorsque les opérations à réaliser sur un poste sont achevées, la palette est réorientée sur le convoyeur principal a n de continuer le cycle de production.

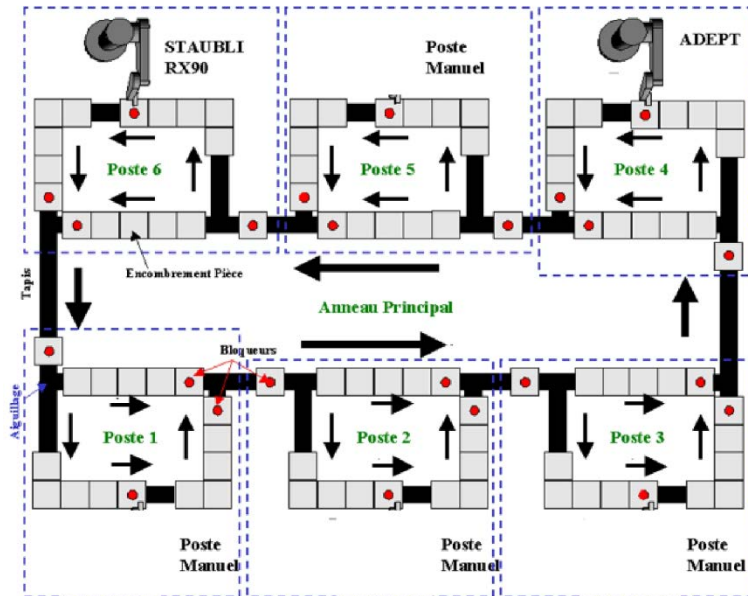


Figure 0.1 : Représentation schématique de l'AIP, avec les 6 automates et postes de travail respectifs, ainsi que le robot Staubli RX90.

Le poste 1 joue le rôle particulier de la gestion du chargement des pièces brutes sur les palettes et du déchargement de celles dont le cycle est terminé. Lorsque toutes les opérations indiquées dans l'étiquette ont été effectuées, la palette entre dans le poste 1 pour y être déchargée, puis rechargée d'une nouvelle pièce pour laquelle est défini un nouveau cycle de production. La conception technologique est identique pour les six postes. La capacité de communication entre et avec les différents équipements du transfert libre est assurée par 2 réseaux de réseaux locaux. Pour la supervision, la gestion de la production, les lancements d'ordre de fabrication et la synchronisation entre postes, le réseau est de type ETHERNET (non déterministe). Au niveau de chaque poste, un réseau UNI-TELWAY (déterministe, architecture maître esclave) permet le dialogue entre un automate (TSX Premium) et des équipements spécifiques tels qu'une interface opérateur (XBT-Magelis) et des plots de lecture/écriture magnétique (XGS). Les stations (ou terminaux de programmation) peuvent être connectées aux 2 réseaux : sur ETHERNET, les stations accèdent à tous les automates, alors que sur UNITELWAY, elles ne peuvent communiquer que localement avec l'automate (voir Figure 2).

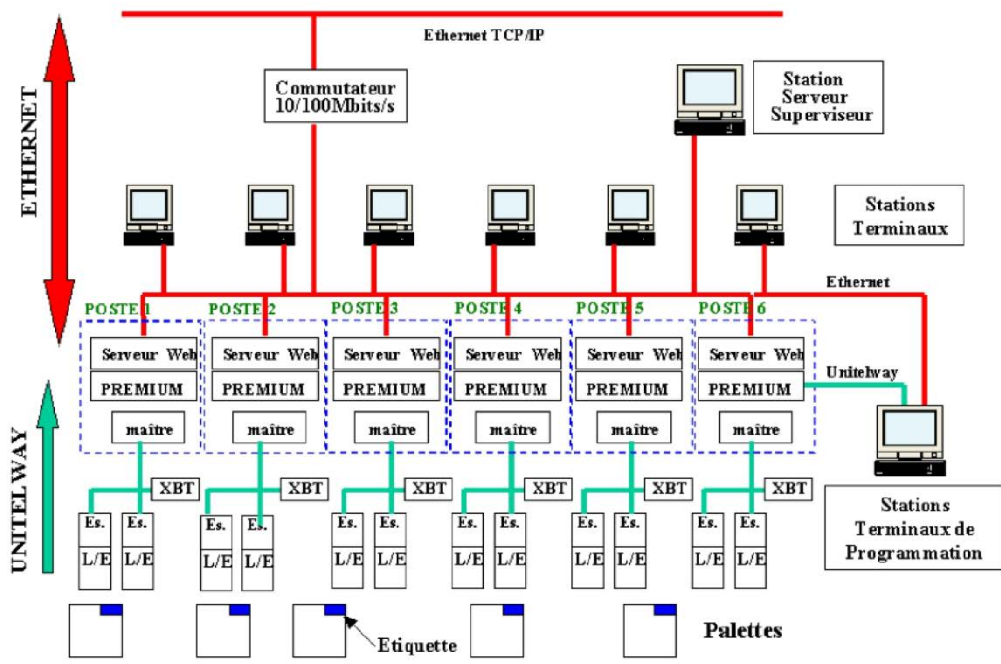


Figure 0.2 : Réseaux de communication à l'AIP.

1.2 Conception technologique des postes

Chaque poste de travail est constitué de trois zones distinctes (voir Figure 3). L'ensemble des informations et des commandes des actionneurs pour chaque poste est câblé sur un unique automate.

1.2.1 La zone d'entrée

Cette zone comprend un capteur inductif pour la détection du passage des palettes, un système de lecture/écriture des étiquettes magnétiques (plot XGS), d'un vérin de blocage des palettes par butée et d'un aiguillage pilotable permettant de dévier les palettes du convoyeur central vers la zone de traitement au besoin. Le vérin d'aiguillage permet d'actionner simultanément la montée de l'aiguillage côté dérivation et la descente côté ligne directe (et réciproquement). La détection de la position basse du vérin de blocage et de la position haute ou basse de l'aiguillage côté dérivation ou côté ligne directe est également possible.

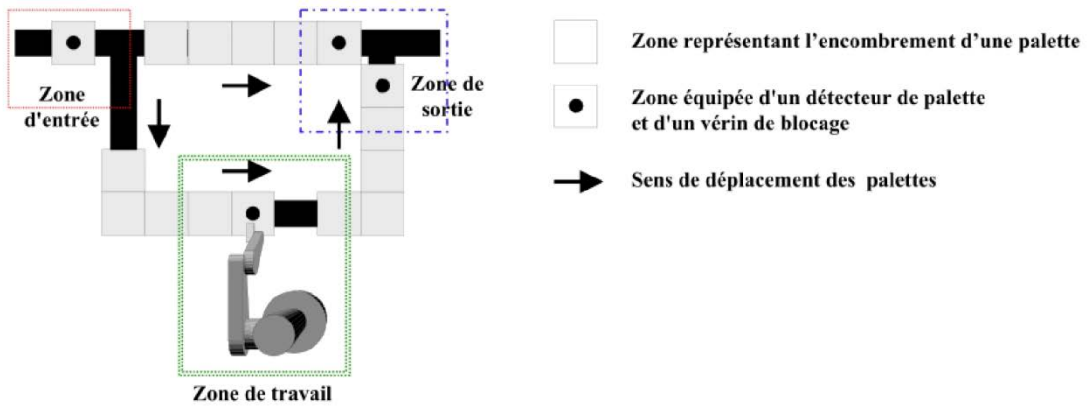


Figure 0.3 : Schéma d'un poste de travail.

1.2.2 La zone de travail

Cette zone dispose d'un système d'indexage des palettes au niveau de la platine d'assemblage. Cette platine d'assemblage est munie d'un mécanisme de blocage et un système de lecture/écriture des informations magnétiques (Plot XGS). Lorsque le traitement est terminé la palette est libérée, puis acheminée vers la zone de sortie. La phase d'indexage est très importante puisqu'elle permet d'assurer une position de maintien fixe et connue pour les opérations à réaliser sur la pièce. Lorsque le capteur inductif en entrée du poste de travail détecte le passage d'une palette, la sortie du vérin de blocage stoppe le mouvement de la pièce au niveau du poste d'indexage, il est en fin possible de relever le système d'indexage constitué de 2 ergots s'insérant dans la palette et la bloquant en appui contre une butée mécanique. La détection de la position haute du système d'indexage permet de valider le positionnement de la palette avant d'entamer toute opération sur la pièce. Une fois l'opération terminée, la descente du mécanisme d'indexage puis du vérin de blocage permet de libérer la palette.

1.2.3 La zone de sortie

Cette zone comprend un vérin de blocage pour la ligne directe et pour la ligne en dérivation, un capteur inductif pour la détection des palettes au niveau des vérins de blocage, et un mécanisme d'aiguillage permettant de rediriger les palettes venant de la dérivation sur le convoyeur principal. Pour éviter les collisions, il est important ici de gérer correctement le partage de la zone de convergence des deux convoyeurs. Le passage d'une pièce sur la zone de convergence a une durée d'un peu plus d'une seconde.

- Automate TSX57 Premium

Afin de piloter le transfert libre, chaque poste est géré par un automate de type TSX57 Premium. Chaque automate commande tous les actionneurs d'un poste et reçoit l'ensemble des informations des différents capteurs du poste. Il pilote également la lecture/écriture des plots XGS et communique avec l'interface opérateur Magelis. L'automate Premium est composé de plusieurs modules associés sur un rack TSXRK8 comportant un bus d'alimentation et un bus de communication. De ce fait, il peut présenter plusieurs configurations. Tous les automates de l'AIP ont une configuration identique composée des différents modules suivants (Figure 4) :

- 1 module d'alimentation : TSXPSY2600M
- 1 module processeur : TSXP57204M
- 1 carte PCMCIA pour bus UNITELWAY
- 1 module coupleur Ethernet : TSXETY5103
- 1 module 32 entrées (TOR) : TSXDEY32D2K
- TSXDEY32D2K1 module 32 sorties (TOR) : TSXDSY32T2K

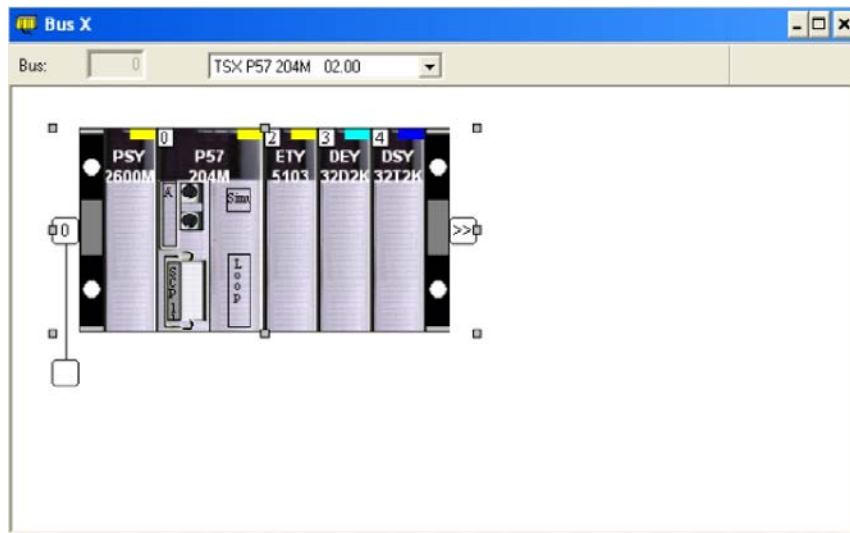


Figure 0.4 : Configuration d'un automate sur Unity Pro.

Le premier module, l'alimentation, permet d'apporter les 24 V nécessaires aux autres modules, c'est à ce niveau que l'automate est relié au réseau électrique. Le second module intègre le processeur de type 57204M et gère le fonctionnement des autres modules de l'automate. Il possède aussi deux ports réservés à la liaison UNITELWAY, le premier (TER) permet de communiquer avec un ordinateur en cas de problème sur le réseau ETHERNET et l'autre est équipé d'une carte PCMCIA Bus UNITELWAY pour le dialogue avec les plots de lecture/écriture et l'interface opérateur. Le module suivant, le coupleur ETHERNET, permet la connexion de l'automate au réseau ETHERNET ; l'adresse IP de l'automate est définie par programmation à l'aide du logiciel UNITY Pro. Il est donc essentiel de ne pas associer la même adresse IP à différents automates pour éviter tout conflit. Ce module peut également accueillir un site WEB. L'adresse X-Way (réseau ETHWAY) est aussi réglable pour différencier les équipements d'un réseau (adressage plus simple que l'adressage IP). Les deux derniers modules gèrent les entrées et les sorties.

Le logiciel UNITY Pro permet de mettre en oeuvre les automates TSX 37 (TSX Micro) et TSX57 (TSX Premium).

2 Initiation : Un premier programme

De façon à explorer les fonctionnalités et l'utilisation du logiciel Unity Pro, on se propose dans un premier temps de programmer un grafcet permettant de piloter le blocage/déblocage d'une palette à l'entrée du poste (zone d'entrée). Le fonctionnement peut se décrire de la façon suivante :

- Blocage de toute palette détectée à l'entrée du poste.
- La palette est bloquée pendant 2s par le vérin de butée d'entrée, puis libérée. Cette temporisation permet de séparer d'éventuelles palettes trop rapprochées.

Récupérer le fichier de configuration de l'automate dans le répertoire U:\Commun\Config. Pour le numéro de poste de l'automate x, copier le fichier postex.stu dans le répertoire U:\tpgmc1 sous le nom postex_INITIALES1_INITIALES2_INITIALES3.stu. Une fois renommé, ouvrir le fichier avec Unity Pro. La fenêtre qui s'ouvre est celle de la Figure 5.

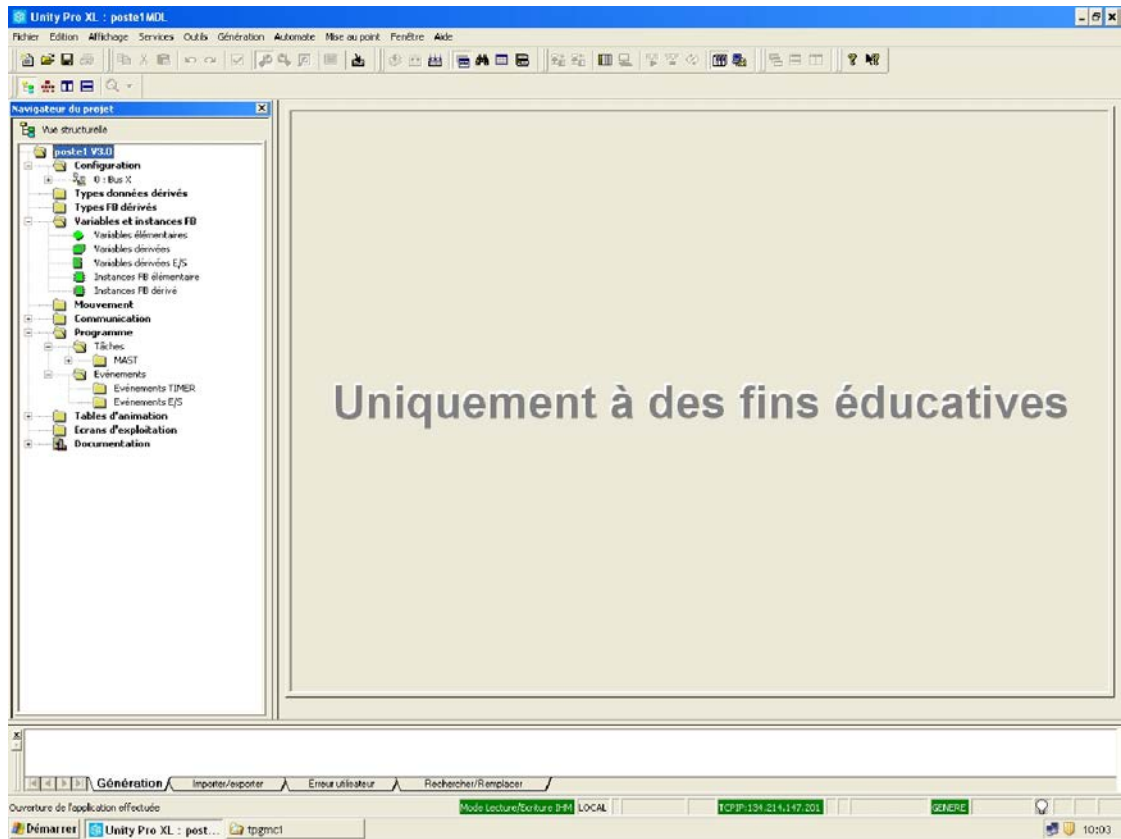


Figure 0.5 : Interface graphique de Unity Pro lors de l'ouverture du fichier de configuration.

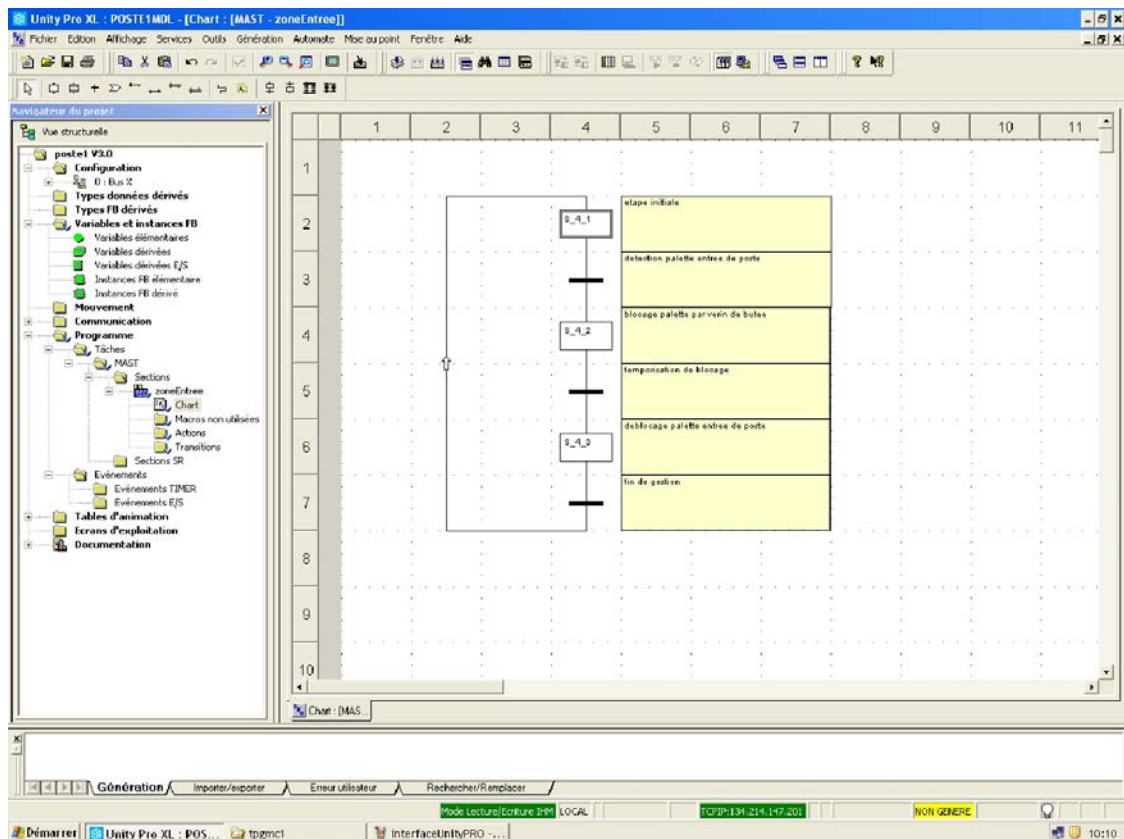


Figure 0.6 : Grafcet de la gestion du verin à l'entrée du poste.

Dans le navigateur de projet, créer une nouvelle section (clic droit) dans l'onglet **MAST** de la rubrique **Programme**. Taper le nom de cette section (**ZoneEntree**) et choisir le langage de syntaxe (SFC). Ouvrir l'onglet **Chart** de la section créée, puis construire le grafcet de gestion d'entrée de la Figure 6. Il faut ensuite renseigner les actions associées à chaque étape ainsi que les transitions. Pour l'étape S_4_1, en double-cliquant sur son icône, sélectionner **étape initiale**. Pour la 1ère transition, choisir variable en sélectionnant le langage LADDER, et insérer, avec la table des entrées/sorties le capteur d'entrée de poste E_capt_pre.

Pour l'étape S_4_2, sélectionner dans l'onglet Action la variable S associée au vérin de butée d'entrée E_verin_b. Pour la 2ème transition, insérer un bloc comparateur pour une temporisation de 2 secondes (Figure 7).

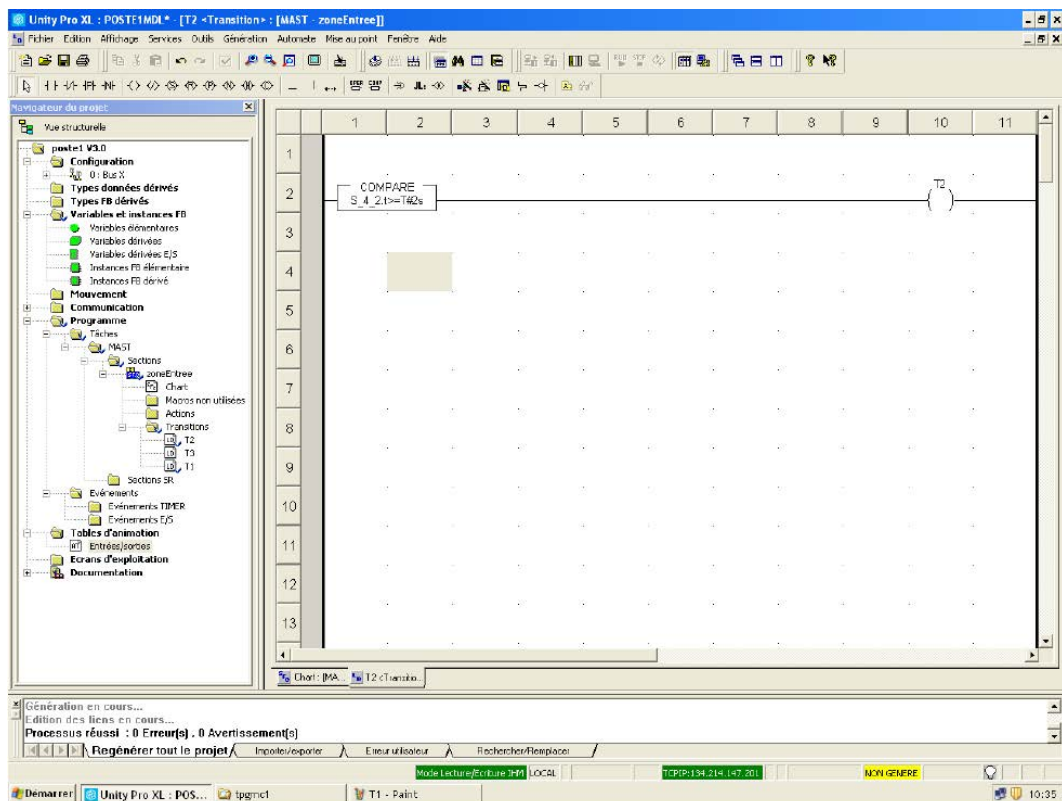


Figure 0.7 : Transition entre S_4_2 et S_4_3.

Pour l'étape S_4_3, sélectionner dans l'onglet Action la variable R associée au vérin de butée d'entrée E_verin_b. Pour la 3ème transition, insérer un bloc comparateur avec une temporisation de 1 seconde.

Analyser l'ensemble de votre projet (vérification syntaxique), puis générer votre projet. Votre projet étant compilé, se connecter à l'automate, effectuer le transfert de votre programme sur l'automate et mise en mode run de l'automate. Avant d'effectuer ces deux dernières opérations, appeler votre responsable de TP.

3 Etude complète : Gestion d'un poste de travail

L'approche proposée ici suit un ordre de complexité croissant. L'objectif final est de gérer globalement un poste de travail.

Question 1 : Donner un descriptif de la technologie de chaque actionneur et capteur utilise sur le poste de travail.

3.1 Zone d'entrée

Pour la gestion de la zone d'entrée, on utilise le blocage/déblocage de palette proposée dans le § 2. Afin de contourner le problème de la lecture des étiquettes, une contrainte de production simple est utilisée : On dévie 1 palette sur 3 (ou plus) dans la zone de travail. Le fonctionnement peut se décrire de la façon suivante :

- Des qu'une palette se présente en entrée du poste, le vérin de blocage est sorti. Le vérin bloque la pièce pendant 2s puis la libère.
- Le vérin de blocage est abaissé et l'aiguillage est orienté en fonction de la contrainte de production (ligne directe ou ligne déviée).

Question 2 : Mettre en œuvre la commande de la zone d'entrée sur l'automate.

3.2 Zone de travail

Pour la zone de travail, le cahier des charges est le suivant :

- Lors de la détection du passage d'une palette à l'entrée de la ligne déviée, le voyant vert clignote.
- Lorsqu'une palette est détectée à l'indexage, le vérin bloque la palette, puis on procède à l'indexage. Lorsque la palette est prête pour l'opération, le voyant orange clignote. Cela signifie que l'indexage s'est bien déroulé, et que la palette est prête pour toute opération.
- Le début de l'opération sur la palette est simulé par un appui sur le bouton poussoir bleu. Le voyant rouge est alors allumé, pour indiquer que la zone de travail est occupée.
- La fin de l'opération sur la palette et son évacuation sont simulés par un 2ème appui sur le bouton poussoir bleu.

Question 3 : Mettre en œuvre la commande de la zone de travail sur l'automate.

3.3 Zone de sortie

Pour la gestion de la zone de sortie, afin d'éviter toute collision dans la zone d'aiguillage de sortie, on autorise la circulation d'une palette si la zone d'aiguillage est libre, avec une priorité pour la ligne directe.

Question 4 : Mettre en œuvre la commande de la zone de sortie sur l'automate.

3.4 Capacité de stockage amont

A partir de la description donnée au §1, on souhaite gérer la capacité de stockage de la ligne amont à la zone de travail. On choisit d'interdire l'entrée de nouvelles palettes dans la zone de travail tant que la zone de stockage est saturée. Le voyant rouge clignotant indiquera la saturation du stock amont au poste de travail.

Question 5 : Mettre en œuvre la commande de la capacité de stockage amont sur l'automate.

3.5 Capacité de stockage aval

De la même manière, on décide de gérer la capacité de stockage de la ligne aval à la zone de travail. La priorité donnée à la ligne directe est annulée tant que le stock aval au poste de travail n'a pas été complètement vide. Le voyant orange fixe indiquera la saturation du stock aval au poste de travail.

Question 6 : Mettre en œuvre la commande de la capacité de stockage aval sur l'automate.

3.6 Sécurité du poste

En vue d'introduire une sécurité locale sur le poste, on utilise le commutateur (automatique / manu) pour basculer d'un mode automatique à un mode semi-automatique. Le mode automatique correspond à la description du fonctionnement qui a été faite jusqu'à maintenant. En mode semi-automatique, le lancement des tâches d'entrée, d'indexage, d'évacuation ou de sortie d'une palette du poste se fait manuellement par appui sur le bouton noir. Chaque appui sur le bouton noir lance la tâche suivante, si la tâche précédente est terminée ou est toujours en attente de débiter.

Question 7 : Mettre en œuvre les modes automatique et semi-automatique sur l'automate.

3.7 Conclusion

Question 8 : Conclure votre travail, en donnant quelques perspectives d'évolution.

Table des entrées/sorties

entrées :

nom symbolique	Commentaire
E_capt_pre	capteur de présence à l'entrée du poste
E_capt_verin_aig1	capteur du vérin d'aiguillage ligne déviée
E_capt_verin_aig2	capteur du vérin d'aiguillage ligne directe
E_capt_verin_b	capteur de vérin à l'entrée du poste (position basse à 1)
G_BP_bleu	bouton poussoir bleu
G_BP_Noir	bouton poussoir noir
G_capt_sat	capteur de saturation du poste
G_mode_automatique	commutateur en position automatique
G_mode_manu	commutateur en position manuelle
G_pre_pal_s	présence palette au poste suivant
P_capt_pre1	capteur de présence au poste de travail (position)
P_capt_pre2	capteur de présence au poste de travail (indexation)
P_capt_verin_i_b	capteur de vérin d'indexage position basse

P_capt_verin_i_h	capteur de vérin d'indexage position haute
P_verin_capt_b	capteur de vérin de butée au poste de travail (position basse à 1)
S_capt_pre_d	capteur de présence à la sortie du poste (ligne directe)
S_capt_pre_s	capteur de présence à la sortie du poste (ligne de sortie de poste)
S_capt_verin_b_d	capteur de vérin à la sortie du poste (ligne directe) (position basse à 1)
S_capt_verin_b_s	capteur de vérin à la sortie du poste (ligne déviée) (position basse à 1)

sorties :

nom symbolique	Commentaire
E_verin_aig	vérin d'aiguillage
E_verin_b	vérin de la butée d'entrée
G_s_voyant_r	voyant de signalisation rouge
G_s_voyant_v	voyant de signalisation vert
G_voyant_o	voyant orange
G_voyant_r	voyant rouge
G_voyant_v	voyant vert
P_verin_b	vérin de butée du poste de travail
P_verin_i_r	vérin d'indexage rentre
P_verin_i_s	vérin d'indexage sorti
S_verin_b_d	vérin de la butée de sortie du poste (ligne directe)
S_verin_b_s	vérin de la butée de sortie de poste (ligne déviée) (position basse à 1)

Annexe E : Questionnaire sur les retours d'utilisation du configurateur

I. Questionnaire destiné aux tuteurs :

Vous avez participé à la création des ressources et configurations de vos TP via le système de configurateur que nous vous avons proposé. Les questions suivantes ont pour but d'évaluer le degré de satisfaction des objectifs que nous avons énoncé pour motiver l'utilisation du configurateur...

1. Vous trouvez la procédure d'insertion des ressources de TP dans le configurateur :
 - Très satisfaisante
 - Assez satisfaisante
 - Peu satisfaisante
 - Pas du tout satisfaisante

2. Vous trouvez la façon de catégoriser les ressources insérées :
 - Très satisfaisante
 - Assez satisfaisante
 - Peu satisfaisante
 - Pas du tout satisfaisante

3. Vous trouvez la procédure d'édition des ressources sauvegardées :
 - Très satisfaisante
 - Assez satisfaisante
 - Peu satisfaisante
 - Pas du tout satisfaisante

4. Vous trouvez la fonctionnalité de recherche des ressources :
 - Très satisfaisante
 - Assez satisfaisante
 - Peu satisfaisante
 - Pas du tout satisfaisante

5. Vous trouvez l'assemblage des ressources sous un composant (configuration générique) :
 - Très satisfaisant
 - Assez satisfaisant
 - Peu satisfaisant
 - Pas du tout satisfaisant

6. Vous trouvez la réutilisabilité des ressources :
 - Très satisfaisante
 - Assez satisfaisante
 - Peu satisfaisante
 - Pas du tout satisfaisante

7. Vous trouvez la réutilisabilité d'un composant pour en générer un autre :
 - Très satisfaisante
 - Assez satisfaisante

- Peu satisfaisante
- Pas du tout satisfaisante

8. Vous trouvez la réutilisabilité d'une configuration pour en générer une autre :
 - Très satisfaisante
 - Assez satisfaisante
 - Peu satisfaisante
 - Pas du tout satisfaisante

9. Vous trouvez la procédure de compilation des ressources de composant pour générer une configuration :
 - Très satisfaisante
 - Assez satisfaisante
 - Peu satisfaisante
 - Pas du tout satisfaisante

II. Questionnaire destiné aux instructeurs :

Vous avez participé à l'exploitation des configurations pour vos TP via le système de configurateur que nous vous avons proposé. Les questions suivantes ont pour but d'évaluer le degré de satisfaction des objectifs que nous avons énoncé pour motiver l'utilisation du configurateur ...

1. Vous trouvez la fonctionnalité de recherche pour consulter les contenus du configurateur liés à un sujet et/ou un système donné :
 - Très satisfaisante
 - Assez satisfaisante
 - Peu satisfaisante
 - Pas du tout satisfaisante

2. Vous trouvez le filtrage du contenu de la bibliothèque de configurations selon une (des) catégories(s) pour en profiter :
 - Très satisfaisant
 - Assez satisfaisant
 - Peu satisfaisant
 - Pas du tout satisfaisant

3. Vous trouvez l'installation d'une configuration de session de TP :
 - Très simple
 - Assez simple
 - Peu compliquée
 - Très compliquée

Merci pour votre participation !

Glossaire

ANSI	American National Standards Institute
AMDEC	Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité
AIP	Atelier Interétablissements de Productique
API	Automate Programmable Industriel
APS	Automated Production System
BDD	Bloc Definition Diagram
CIM	Computational Independent Model
CMC	Computer Mediated Communication
CBL	Computer-Based Learning
CBT	Computer-Based Training
CSCL	Computer-Supported Collaborative Learning
CCG	Conseil de Coopération du Golfe
CMS	Content Management System
DFB	Derived Function Block
DOOS	Designing Object Oriented Software
DUT	Diplôme Universitaire de Technologie
EIA	Electronic Industries Alliance
ELab	Electronic Laboratory
ELAMS	Electronic Laboratory Management System
ESDL	Embedded Systems Design Lifecycle
EAO	Enseignement Assisté par Ordinateur
EIAO	Enseignement Intelligemment Assisté par Ordinateur
EIAH	Environnement Informatisé pour l'Apprentissage Humain
ExAO	Expérimentation Assistée par Ordinateur
XML	Extensible Markup Language
FAQ	Foire Aux Questions
FB	Fonction Block
FAD	Formation À Distance
FOAD	Formation Ouverte et À Distance
GPIB	General Purpose Interface Bus
GI	Génie industriel
GMC	Génie Mécanique Construction
HLab	Hybrid Laboratory
HTML	HyperText Markup Language
IPR	Indice de Priorité de Risque
IS	Ingénierie Système
INSA	Institut National des Sciences Appliquées
IL	Instructions list
IA	Intelligence Artificielle
IHM	Interaction Homme-Machine
IBD	Internal Bloc Diagram
INCOSE	International Council on Systems Engineering

IEC	International Electrotechnical Commission
IP	Internet Protocole
LVOOP	LabVIEW Object Oriented Programming
LD	Ladder Diagram
LCMS	Learning Content Management System
LG	Learning Games
LMS	Learning Management System
LOD	Learning Object Download
MES	Manufacturing Execution System
MOOC	Massive Open Online Course
5M	Matière, Milieu, Méthodes, Matériel et Main d'œuvre
MDA	Model Driven Architecture
Moodle	Modular Object-Oriented Dynamic Learning Environment
OMG	Object Management Group
OMT	Object Modeling Technique
OOA	Object Oriented Analysis
OOAD	Object Oriented Analysis and Design
OOD	Object Oriented Design
OODA	Object Oriented Design with Applications
OOP	Object Oriented Programming
OOSA	Object Oriented Systems Analysis
OOADM	Object-Oriented Analysis and Design Methodology
PXI	PCI eXtensions for Instrumentation
PCI	Peripheral Component Interconnect
PC	Personal Computer
PDCA	Plan-Do-Check-Act
POM	Planification, Optimisation et Modélisation
PIM	Platform Independent Model
PM	Platform Model
PSM	Platform Specific Model
Priméca	Pôle de Ressources Informatiques pour la Mécanique
PDF	Portable Document Format
PLM	Product Lifecycle Management
PLC	Programmable Logic Controller
POO	Programmation Orientée Objet
RFID	Radio-Frequency Identification
RA	Réalité Augmenté
RM	Réalité Mixte
RLab	Remote Laboratory
RAO	Rhône Alpes Ouest
STIC	Sciences des Technologies de l'Information et de la Communication
STIC	Sciences des Technologies de l'Information et de la Communication
SG	Serious Games
SCORM	Sharable Content Object Reference Model

SIMILAR	State, Investigate, Model, Integrate, Launch, Assess, and Reevaluate
SWOT	Strengths, Weaknesses, Opportunities, and Threats
SQL	Structured Query Language
ST	Structured Text
SAP	Système Automatisé de Production
STI	Systemes Tutoriels Intelligents
SE	Systems Engineering
SysML	Systems Modeling Language
TRS	Taux de Rendement Synthétique
TIC	Technologies de l'Information et de la Communication
TICE	Technologies de l'Information et de la Communication pour l'Enseignement
TEL	Technology-Enhanced Learning
TéléTP	télé Travaux Pratiques
TrAVis	Tracking Data Analysis and Visualization
TD	Travaux Dirigés
TP	Travaux Pratiques
e-TP	Travaux Pratiques Electroniques
UML	Unified Modeling Language
UNESCO	United Nations Educational, Scientific and Cultural Organization
UDFB	User Defined Function Block
VLab	Virtual laboratory
VA	Virtualité Augmentée
VXI	Virtualization Experience Infrastructure
WBT	Web-Based Training

Références

[Abdulwahed et al.-2011]

Abdulwahed, M. & Nagy, Z. K. "The TriLab, a novel ICT based triple access mode laboratory education model", *Comput. Educ.*, Elsevier Science Ltd., volume 56, pages 262-274, 2011.

[Alvarez-2008]

Julian ALVAREZ, Laurent MICHAUD "Serious games: advergaming, edugaming, training", de l'audiovisuel et des télécommunications en Europe (France), I. IDATE, 2008.

[Amat-2003]

Christian AMAT "La pédagogie par projet", Institut Universitaire de Formation des Maîtres de l'Académie de Montpellier - Génie électrotechnique, Terminale S SI, Lycée Mermoz, Montpellier, 2003.

[ANSI/EIA_632-2003]

"Processes for Engineering a System", Government Electronics and Information Technology Association, ANSI/GEIA EIA-632, 01 Septembre 2003.

[Ardito et al.-2004]

C. Ardito, M. De Marsico, R. Lanzilotti, S. Levialdi, T. Roselli, V. Rossano, M. Tersigni "Usability of E-learning tools", Proceedings of the working conference on Advanced visual interfaces, ACM, pages 80-84, 2004.

[Arnous et al.-2009]

Saher Arnous, Arnaud Leleve, Khalid Kouiss, Patrick Prevot "Towards semi-automatic generation of training scenarios in industrial automated systems", Proceedings of the International Conference on Management of Emergent Digital EcoSystems, ACM, Article No 74, pages 460-465, 2009.

[Arnous et al-2012]

Saher ARNOUS, Arnaud LELEVE, Patrick PREVOT, Khalid KOUISS, "Configuration and Management Tool for Laboratory Training", 6th IEEE International Conference on ELearning in Industrial Electronics (ICELIE 2012), Montreal-Canada, 25-28 October 2012.

[Arvehammar-2007]

Arvehammar, M. "Object-Oriented Automation System", Department of Automatic Control, Lund Institute of Technology, Sweden, 2007.

[Ashish-2006]

Mani et C. PATVARDHAN, "A study of ICT enabled laboratories", India Conference, Annual IEEE , page 1, ISBN 1-4244-0369-3, doi: 10.1109/INDCON.2006.302765, New Delhi, India, 15-17 sept. 2006.

[Baccigalupi et al.-2006]

A. Baccigalupi, C. De Capua, and A. Liccardo. "Overview on development of remote teaching laboratories: from labview to web services". In *Instrumentation and Measurement Technology Conference, 2006. IMTC 2006. Proceedings of the IEEE*, pages 992–997, 2006.

[Backström-2000]

Backström, T. & Döös, M. "Problems with machine safeguards in automated installations", *International Journal of Industrial Ergonomics*, volume 25, pages 573-585, 2000.

[Bagnasco et al.-2008]

Bagnasco, A.; Buschiazzo, P.; Ponta, D. & Scapolla, M. "A learning resources centre for simulation and remote experimentation in electronics" Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments, ACM, pages 63:1-63:7, 2008.

[Bahill-1998]

Bahill, A. T. & Gissing, B. "Re-evaluating systems engineering concepts using systems thinking", *Trans. Sys. Man Cyber Part C*, IEEE Press, volume 28, pages 516-527, 1998.

[Balasubramanian et al.-2009]

K. Balasubramanian, Willie Clarke-Okah, John Daniel, Frances Ferreira, Asha Kanwar, Angela Kwan, John Lesperance, Joshua Mallet, Abdurrahman Umar, Paul West "ICTs for Higher Education", Commonwealth of Learning, UNESCO World Conference on Higher Education, Paris, 5-8 juillet 2009.

[Barron-2002]

Barron, T. "Evolving Business Models in e-Learning", *Learning on Demand (LoD)*, volume 2, pages 1-14, 2002.

[Balter et al.-2005]

Bälter, O.; Engwall, O.; Öster, A.-M. & Kjellström, H. "Wizard-of-Oz test of ARTUR: a computer-based speech training system with articulation correction", *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*, ACM, pages 36-43, 2005.

[Beck-2007]

D. Beck, H. B. "Control System Design using LabVIEW Object-Oriented Programming", *Proceedings of ICALEPCS07*, Knoxville, Tennessee, USA, 2007.

[Bellmunt et al.-2006]

O.G. Bellmunt, D.M. Miracle, S.G. Arellano, A. Sumper, and A.S. Andreu. "A distance plc programming course employing a remote laboratory based on a flexible manufacturing cell". *Education, IEEE Transactions on*, 49(2):278-284, 2006.

[Ben Youssef-2010]

BEN YOUSSEF, Adel and DAHMANI, Mounir (2010) "The impact of ICT's on students' performance in Higher Education: Direct effects, Indirect effects and Organizational change", *The First International Conference on e-Learning For All (LEAFA)*, Hammamet-Tunisia, 3-5 Juin 2010.

[Benadi-2003]

BENADII, M. D. "Construction d'environnements de télé-expérimentation", *DEA Informatique et Systèmes coopératifs pour l'entreprise*. Lyon: INSA de Lyon, page 30, 2003.

[Benmohamed-2007]

Hcene Benmohamed "ICTT@Lab: un environnement informatique pour la génération et l'exécution de scénarios de téléTP", thèse de doctorat en informatique, L'Institut National des Sciences Appliquées de Lyon, Lyon, 2007.

[Benmohamed et al.-2009]

Hcene Benmohamed, Arnaud Leleve & Prévot, P. "Reusing Electronic Laboratory Learning Scenarios" *IEEE Transactions on Learning Technologies*, 2009.

[Blanch-2010]

Angel Blanch, Anton Aluja "Job involvement in a career transition from university to employment", *Learning and Individual Differences*, volume 20, pages 237-241, 2010.

[Berger-2002]

A. S. Berger. *Embedded Systems Design : An Introduction to Processes, Tools and Techniques*. CMP Books, ISBN 1 578 20073 3, 2002.

[Bonfe-2006]

Marcello Bonfe, Cesare Fantuzzi, C. S. "Behavioral inheritance in object-oriented models for mechatronic system", *International Journal of Manufacturing Research*, volume 1, No 4, 421-441, 2006.

[Booch-1991]

Booch, G., "Object-Oriented Design with Applications", The Benjamin/Cummings Publishing Company Inc., Redwood City, CA, 1991.

[Bouquet et al.-1997]

Gabrielle BOUQUET, Yannick BRUXELLE, Jean BURGER, Louis ESPINASSOUS, Roland GERARD, Yvan LE GOFF, Laurent MARSEAULT, Jean Paul SALASSE "Alterner pour apprendre: entre pédagogie de projet et pédagogie de l'écoformation", Réseau Ecole et Nature, 1997.

[Boychev et al.-2012]

Boychev, P.; Kanev, K. & Nikolov, R. "Technology enhanced learning with subject field multiplicity support", Proceedings of the 2012 Joint International Conference on Human-Centered Computer Environments, ACM, pages 39-44, 2012.

[Brennan et al.-2010]

Brennan, J.; Arthur, L.; Little, B.; Cochrane, A.; Williams, R.; Locke, W.; Singh, M.; David, M.; Kim, T. & King, R. "Higher Education and Society: A research report CHERI", 2010.

[Buchanan-2011]

Buchanan, L, W. F. & Zinghini, F. "Blending Bloom's Taxonomy and Serious Game Design", In Proceedings of the 2011 International Conference on Security and Management, Las Vegas-Nevada USA, volume 2, 2011.

[Buhler et al.-2000]

Buhler, D., Kuchlin, W., Grubler, G. & Nusser, G. "The Virtual Automation Lab-Web based teaching of automation engineering concepts Engineering of Computer Based Systems", ECBS 2000 Proceedings. Seventh IEEE International Conference and Workshop, pp. 156-164, 2000.

[Candelas et al.-2006]

F. A. Candelas Herías, C. A. Jara Bravo, F. Torres Medina. "Flexible virtual and remote laboratory for teaching robotics". pages 1959–1963, 2006.

[Chartrand-2005]

Chartrand-Beauregard, Julie et Sylvie Gingras. "L'Économie du savoir au Québec", Ministère du Développement économique, de l'innovation et de l'exportation. Gouvernement du Québec, 28 pages, p. 7, mai 2005.

[Chen et al.-2012]

Jian Chen, Hongyuan Wang, K. W. Y. Y. J. T. "A Configurable and Extensible Portal Framework for E-Learning Environments", 3rd International Conference on e-Education, e-Business, e-Management and e-Learning (IC4E2012), IPEDR volume 27, IACSIT Press, Singapore, pages 21-26, 2012.

[Chiron-2007]

Fabien CHIRON, "Contribution à la flexibilité et à la rapidité de conception des systèmes automatisés avec l'utilisation d'UML", Thèse de doctorat en informatique, université Blaise Pascal de Clermont-Ferrand, 2007.

[Christou et al.-2007]

I. T. Christou, S. Efremidis, T. Tiropanis, and A. Kalis, "Grid-based virtual laboratory experiments for a graduate course on sensor networks, " IEEE Transactions on Education, vol. 50, no. 1, pp. 17–26, 2007.

[Cisel-2012]

Matthieu Cisel, Éric Bruillard "Chronique des MOOC", Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation, volume 19, 2012.

[Clarkson-2005]

- J. Clarkson, C. Eckert, Design process improvement, a review of current practice, ISBN 1-85233-701-X, Springer- Verlag London Limited, 2005.
- [Clavien-2003]
Lionel Clavien, Mireille Bétrancourt "Animations multimédia : quels dispositifs pour réduire la charge cognitive?", Journée EPIQUE 2003, Université de Genève, 2-3 octobre 2003.
- [Coad-1991a]
Coad, P., Yourdon, E. "Object Oriented Analysis (2nd Edition) ", Yourdon Press, Englewood Cliffs, N.J., 1991.
- [Coad-1991b]
Coad, P., Yourdon, E. "Object Oriented Design", Yourdon Press, Englewood Cliffs, N.J., 1991.
- [Consonni-2001]
Consonni, D. & Seabra, A. C. "A modern approach to teaching basic experimental electricity and electronics", IEEE Transactions on Education, 44, 5-15, 2001.
- [Coquard et al.-2008]
Coquard, P.; Guillemot, M.; Lelevé, A. & Noterman, D. and Benmohamed, H. "AIP-Primeca RAO Remote Laboratories in Automation", International Journal of Online Engineering (iJOE), 4 , 12-18, 2008.
- [Corter et al.-2004]
J.E. Corter, J.V. Nickerson, S.K. Esche, and C. Chassapis. "Remote versus hands-on labs: a comparative study", In Frontiers in Education, 2004. FIE 2004. 34th Annual, pages FIG-17-21 Vol. 2, 2004.
- [Corter et al.-2007]
Corter J. E., Nickerson J. V., Esche S. K., Chassapis C., Im S., Ma J., "Constructing reality: A study of remote, hands-on, and simulated laboratories", ACM Trans. Comput.-Hum. Interact., volume 14, n° 2, p. 7, 2007.
- [Davoli et al.-2006]
F. Davoli, G. Spanospano, S. Vignola, and S. Zappatore, "Labnet: Towards remote laboratories with unified access", IEEE Transactions on Instrumentation and Measurement, vol. 55, no. 5, pp. 1551-1558, 2006.
- [Deelstra et al.-2003]
Sybren Deelstra, Marco Sinnema, Jilles Van Gorp, Jan Bosch "Model driven architecture as approach to manage variability in software product families", In Proceedings of the Workshop on Model Driven Architectures: Foundations and Applications, Springer, pages 109-114, 2003.
- [Djamshid et al.-2004]
Djamshid Tavangarian, Markus E Leypold, K. N. M. R. D. V. "Is e-Learning the Solution for Individual Learning ?", Electronic Journal of e-Learning, volume 2, pages 273-280, 2004.
- [Dobre-2006]
Iuliana Dobre "Aspects of Collaboration Tools in E-Learning", Mathematics – Informatics – Physics Series, Bulletin Petroleum-Gas University of Ploiesti, Ed. Petroleum-Gas University of Ploiesti, vol. LVIII, no. 1/2006, pages 51-56, 2006.
- [Domínguez et al.-2011]
M. Domínguez, J. J. Fuertes, M. A. Prada, S. Alonso, and A. Morán, "Remote laboratory of a quadruple tank process for learning in control engineering using different industrial controllers", Computer Applications in Engineering Education, 2011.
- [Doucet et al.-2004]

Patrik DOUCET, François GAUTHIER, Patrick DUBOIS "Démarche d'analyse et de maîtrise du risque (DAMR) intégrée à un Processus de développement de produit (PDP)", Université de Sherbrooke & Université du Québec à Trois-Rivières, 12 Septembre 2004.

[Emami et al.-2008]

M.Reza Emami and M.R. Adrian Martin, A. Emami. "Dynamic load emulation for robotic hardware-in-the-loop simulation platforms", IEEE International Symposium on Industrial Electronics Cambridge, UK, 2008.

[Estefan-2008]

Estefan, Jeff A. "Survey of Model-Based Systems Engineering (MBSE) Methodologies," Rev. B, INCOSE Technical Publication, Document No.: INCOSE-TD-2007-003-01, International Council on Systems Engineering, San Diego, CA, June 10, 2008.

[Evrot-2008]

Dominique EVROT "Contribution à la vérification d'exigences de sécurité : Application au domaine de la machine industrielle", thèse de doctorat en Automatique, L'Université Henri Poincaré, Nancy I, Nancy, 2008.

[Fayolle et al.-2010]

Fayolle, J. and Gravier, C. and Jailly, B. "Collaborative remote laboratory in virtual world", Proceedings of the 10th WSEAS international conference on applied informatics and communications, and 3rd WSEAS international conference on Biomedical electronics and biomedical informatics, World Scientific and Engineering Academy and Society (WSEAS), pages 396-402, 2010.

[Feisel-2005]

Lyle D. Feisel and Albert J. Rosa "The role of the laboratory in undergraduate engineering education", Journal of Engineering Education, volume 94, pages 121-130, 2005.

[Friedenthal et al.-2008]

Sanford Friedenthal, Alan Moore, Rick Steiner "Practical Guide to SysML: Systems Modeling Language", Elsevier Science, 2008.

[Ghasemi-2011]

Ghasemi, B. & Hashemi, M. "ICT: Newwave in English language learning/teaching", Procedia - Social and Behavioral Sciences, volume 15, pages 3098-3102, 2011.

[Gillet et al.-2005]

D. Gillet, A. V. N. Ngoc, and Y. Rekik, "Collaborative web-based experimentation in flexible engineering education", IEEE Transactions on Education, vol. 48, no. 4, pp. 696-704, 2005.

[Gilliot-2013]

Gilliot Jean-Marie "Des formations gratuites et en accès libre sur Internet", Pour la science, no. 431, pp. 34-40, septembre 2013.

[Gravier-2007]

Christophe GRAVIER "Vers la généralisation de manipulations distantes et collaboratives d'instruments de haute technologie", Thèse de doctorat en informatique, université Jean Monnet de Saint-Étienne, 2007.

[Grimaldi-2009]

Grimaldi, D. & Rapuano, S. "Hardware and software to design virtual laboratory for education in instrumentation and measurement Measurement", volume 42, pages 485-493, 2009.

[Guimaraes et al.-2000]

Bruno Guimarães, Alexandre S. Souza, Hugo L. Gosmann, Adolfo Bauchspiess. "Internet based remote laboratory: The level control of three coupled water reservoirs", IFAC, 2000.

[Hakulinen-2011]

Hakulinen, L. "Using serious games in computer science education", Proceedings of the 11th Koli Calling International Conference on Computing Education Research, ACM, pages 83-88, 2011.

[Harper et al.-2004]

Harper, K. C.; Chen, K. & Yen, D. C. "Distance learning, virtual classrooms, and teaching pedagogy in the Internet environment", Technology in Society, volume 26, pages 585-598, 2004.

[Hextlearn-2010]

"Higher Education EXploring ICT use for Lifelong LEARNing", Budapesti Műszaki és Gazdaságtudományi Egyetem, Final report, 2010.

[Honeyman-1993]

Honeyman, M; Miller, G "Agriculture distance education: A valid alternative for higher education?", Proceedings of the 20th Annual National Agricultural Education Research Meeting, pages 67-73, December 1993.

[Hong-1993]

Hong, S. "A formal approach to the comparison of object-oriented analysis and design methodologies", In The Twenty-Sixth Annual Hawaii International Conference on System Sciences, pages 689-698, 1993.

[Houssaye-1993]

Houssaye, J. "Le triangle pédagogique ou comment comprendre la situation pédagogique", La pédagogie : une encyclopédie pour aujourd'hui, Paris : ESF, 1993.

[Hsieh-2001]

Hsieh, S. J. & Hsieh, P. Y. "Intelligent tutoring system authoring tool for manufacturing engineering education", International Journal of Engineering Education, Citeseer, volume 17, pages 569-579, 2001.

[Hubackova-2011]

Hubackova, S. "Possibilities of the use of ICT in interuniversity studies", Procedia - Social and Behavioral Sciences, volume 28, pages 29-33, 2011.

[Hubert-1999]

Hubert, M., "Apprendre en projets : la pédagogie du projet-élèves", Lyon : Chronique sociale., coll. « Pédagogie Formation », ISBN 2-85008-583-9, 1999.

[Ishutkina et al.-2004]

M. Ishutkina, E. Feron, M. Casini, and A. Vicino. "An internet based laboratory for control of a safety critical system", In Systems, Man and Cybernetics, 2004 IEEE International Conference on, volume 3, pages 2707-2712, 2004.

[Jamian et al.-2012]

Jamian, M.; Jalil, H. A. & Krauss, S. E. "Malaysian Public University Learning Environments: Assessing Conduciveness through ICT Affordances", Procedia - Social and Behavioral Sciences, volume 35, pages 154-161, 2012.

[Jesus-2007]

Alamo, Jesus A. Del "Advances on remote laboratories and e-learning experiences", iJOE, volume 3, 2007.

[Jin-2006]

Xin Jin "Applying Model Driven Architecture Approach to Model Role Based Access Control System", Thèse de Master en (Systems Science), University of Ottawa, 2006.

[John-2001]

Karl-Heins John, M. T. "IEC 61131-3: Programming Industrial Automation Systems: Concepts and Programming Languages, Requirements for Programming Systems, Aids to Decision-making Tools", Springer, 2001.

[Jordan-2008]

Jordan, A. G. "Frontiers of research and future directions in information and communication technology", *Technology in Society*, volume 30, pages 388-396, 2008.

[Kaur-2011]

Shubpreet Kaur, Shivani Goel "Testing Anomalies in Multiple and Multilevel Inheritance in Object-Oriented Systems", *Interbational Journal of computers and communications*, Volume 1, No 1, June 2011.

[Kaiser-1978]

Kaiser, F. & Staudenmaier, H. "Computer aided laboratory instruction", *Computer Physics Communications*, pages 335 - 339, 1978.

[Kikuchi et al.-2004]

T. Kikuchi, S. Fukuda, A. Fukuzaki, K. Nagaoka, K. Tanaka, T. Kenjo, and D.A. Harris. "Dvts-based remote laboratory across the pacific over the gigabit network", *Education, IEEE Transactions on*, 47(1):26–32, 2004.

[Kluwer et al.-2003]

Tom Murray, Stephen Blessing, S. A. (Ed.) "Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software", *Kluwer Academic Publishers, Dordrecht*, chapter 11, 2003.

[Knecht et al.-2012]

Knecht, M.; Traxler, C.; Mattausch, O. & Wimmer, M. "Reciprocal shading for mixed reality", *Computers & Graphics*, 2012.

[Kuo-2009]

Yen-Hung Kuo, Yueh-Min Huang "MEAT: An Authoring Tool for Generating Adaptable Learning Resources", *Educational Technology & Society*, volume 12, pages 51-68, 2009.

[Kuester-2007]

F. Kuester, T. C. Hutchinson, "A virtualized laboratory for earthquake engineering education", *Computer Applications in Engineering Education*, vol. 15, no. 1, pp. 15–29, 2007.

[Lallican et al.-2007]

Jean-Louis Lallican, Pascal Berruet, André Rossi, and Jean Luc Philippe. "A component-based approach for conveying systems control design", *kananaskis, alberta, kanada. In ICINCO-SPSMC*, pages 329–338, 2007.

[LabVIEW-2007]

"LabVIEW: Getting Started with LabVIEW", *National Instruments*, 2007.

[LabVIEW_OOP-2009]

"LabVIEW Object-Oriented Programming: The Decisions Behind the Design", *National Instruments*, 26 August 2009.

[LI-2005]

S. Li, A. Khan, "Applying it tools to a laboratory course for measurement, analysis, and design of electric and electronic circuits", *IEEE Transactions on Education*, vol. 48, no. 3, pp. 520–530, 2005.

[Lindsay-2005]

E.D. Lindsay, M.C. Good. "Effects of laboratory access modes upon learning outcomes", *Education, IEEE Transactions on*, 48(4):619–631, 2005.

[Logar et al.-2011]

Logar, V., Karba, R., Papič, M. & Atanasijević-Kunc, M. "Artificial and real laboratory environment in an e-learning competition", *Mathematics and Computers in Simulation*, Vol. 82(3), pp. 517 - 524, 2011.

[Louw et al.-2009]

- Louw, J.; Brown, C.; Muller, J. & Soudien, C. "Instructional technologies in social science instruction in South Africa", *Computers & Education*, volume 53, pages 234-242, 2009.
- [Madsen-1988]
Madsen, O. L. & Moller-Pedersen, B. "What Object-Oriented Programming May Be - and What It Does Not Have To Be", *Proceedings of the European Conference on Object-Oriented Programming*, Springer-Verlag, pages 1-20, 1988.
- [Majchrzak et al.-2005]
Majchrzak Ann, Malhotra Arvind, John Richard "Perceived Individual Collaboration Know-How Development Through Information Technology-Enabled Contextualization: Evidence from Distributed Teams", *Info. Sys. Research, INFORMS*, volume 16, pages 9-27, 2005.
- [Marfisi et al.-2009]
Iza Marfisi-Schottman, Aymen Sghaier, Sébastien George, Franck Tarpin-Bernard, Patrick Prévôt, "Towards Industrialized Conception and Production of Serious Games", *International Conference on Technology and Education ICTE*, Paris, France, pages 1016-1020, 25-27 June 2009.
- [Marfisi et al.-2010]
Iza Marfisi-Schottman, Sébastien George, Franck Tarpin-Bernard, "Tools and Methods for Efficiently Designing Serious Games", *4th European Conference on Games Based Learning ECGBL2010*, Copenhagen, Denmark, pages 226-234, 21-22 October 2010.
- [Marne et al.-2011]
Bertrand Marne, Benjamin Huynh-Kim-Bang, Jean-Marc Labat, "Articuler motivation et apprentissage grâce aux facettes du jeu sérieux", *Environnements informatiques pour l'apprentissage humain, Conférence EIAH*, Belgique, 2011.
- [Martin-1992]
Martin, J., Odell, J. "Object Oriented Analysis and Design", Draft manuscript, 1992.
- [Marton-1994]
Philippe Marton "La conception pédagogique de systèmes d'apprentissage multimédia interactif : Fondements, méthodologie et problématique", *Educatechnologiques*, volume 1, n° 3, pages 91-112, 1994.
- [May et al.-2011]
Madeth May, Sébastien George, Patrick Prévot "TrAVis to Enhance Online Tutoring and Learning Activities: Real Time Visualization of Students Tracking Data", *International Journal of Interactive Technology and Smart Education*, Emerald, volume 8, pages 52-69, 2011.
- [Millar-2009]
Robin Millar, Ian Abrahams "Practical work: making it more effective", *SSR*, volume 91(334), pages 59-64, September 2009.
- [Moum et al.-2009]
Moum A., Koch C., Haugen T. I. "What did you learn from practice today? Exploring experiences from a Danish R&D effort in digital construction", *Advanced Engineering Informatics*, volume 23, pages 229-242, 2009.
- [Munro et al.-1997]
Munro, A.; Johnson, M. C.; Pizzini, Q. A.; Surmon, D. S.; Towne, D. M. & Wogulis, J. L. "Authoring Simulation-centered tutors with RIDES", *International Journal of Artificial Intelligence in Education*, volume 8, pages 284-316, 1997.
- [Muratet-2010]
Muratet Mathieu "Conception, réalisation et évaluation d'un jeu sérieux de stratégie temps réel pour l'apprentissage des fondamentaux de la programmation", *Thèse de doctorat*

- en informatique, Institut de Recherche en Informatique de Toulouse (IRIT), UMR 5505, Université Toulouse III - Paul Sabatier, p. 26, 2010.
- [Ney et al.-2004]
M. Ney, R. Nifle, S. Charles, M. Macedo-Rouet, L. Humblot, C. Batier "Approche méthodologique pour faire évoluer la pédagogie en TD vers un apprentissage actif soutenu par des TICE", in collection, pages 157-165, 2004.
- [Norton et al.-2008]
Norton, C.; Cameron, I.; Crosthwaite, C.; Balliu, N.; Tade, M.; Shallcross, D.; Hoadley, A.; Barton, G. & Kavanagh, J. "Development and deployment of an immersive learning environment for enhancing process systems engineering concepts Education for Chemical Engineers", volume 3, pages e75-e83, 2008.
- [Oblinger-2000]
Diana Oblinger, Jill Kidwell "Distance learning: Are we being realistic?", Educause Review, volume 35, pages 30-39, 2000.
- [OMG_MDA-2003]
"MDA Guide Version 1.0", Object Management Group, 2003.
- [OMG_UML-RFP-2003]
OMG "UML for Systems Engineering Request for Proposal", Needham, MA, USA: Object Management Group. OMG ad/2003-3-41, 2003.
- [Orliac et al.-2011]
Charlotte Orliac, Sébastien George, Christine Michel, Patrick Prévot "Can we use Existing Pedagogical Specifications to Design Mixed Reality Learning Games ?", In 5th European Conference on Games-Based Learning,(ECGBL) Athens, Greece. pages 440-448, 2011.
- [Ott-2011]
Ott M., Pozzi F. "Towards a new era for Cultural Heritage Education: Discussing the role of ICT", Computers in Human Behavior, volume 27, pages 1365-1371, 2011.
- [Ramaswamy et al.-2006]
B. Ramaswamy, Y. Chen, K. Moore, "Omni-directional robotic wheel - a mobile real-time control systems laboratory", in American Control Conference, p. 6, 2006.
- [Randriamalaka-2006]
Randriamalaka, N. "Construction d'une base de patterns pour la réingénierie de scénarios pédagogiques", Rencontres Jeunes Chercheurs EIAH 2006, pages 35-42, 2006.
- [Robertson-2006]
Suzanne Robertson, James Robertson "Mastering the Requirements Process (2nd Edition)", Addison-Wesley Professional, 2006.
- [Rochet-2008]
Samuel ROCHET "Formalisation des processus de l'Ingénierie Système : Proposition d'une méthode d'adaptation des processus génériques à différents contextes d'application", thèse de doctorat en Systèmes Automatiques, L'Université de Toulouse délivré par l'Institut National des Sciences Appliquées de Toulouse, Toulouse, 2008.
- [Rohrig-2003]
C. Rohrig, A. Bischoff. "Web-based environment for collaborative remote experimentation", In Decision and Control, 2003. Proceedings. 42nd IEEE Conference on (volume 3), pages 2514-2518, 2003.
- [Romiszowski-2003]
Romiszowski, A. "The future of E-learning as an educational innovation : Factors influencing project success and failure", Elearning; Tecnologia, T. T. S. & Sistemas, pages 1-14, 2003.
- [Roques-2009]

- Pascal Roques "SysML par l'exemple", Eyrolles, ISBN : 978-2-212-85006-2, 2009.
[Rumbaugh-1991]
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. "Lorensen, W., Object Oriented Modelling and Design", Prentice-Hall, Englewood Cliffs, N.J., 1991.
[Sahfogang-2009]
- Sahfogang Gnetcheyo Guillaume "Conception d'un cours sur les fonctions du tableur excel en classe de seconde", Université de Yaounde I, Département d'informatique et des technologies éducatives, 2009.
[Said et al.-2012]
- Farah Said, Benachenhou Abdelhalim, Neveux Guillaume, Barataud Denis "Design of a Flexible Hardware Interface for Multiple Remote Electronic practical Experiments of Virtual Laboratory", International Journal of Online Engineering (iJOE), Volume 8, Issue S2, pages 7-12, 2012.
[Sánchez et al.-2011]
- Jaime Sánchez, Álvaro Salinas, Jordan Harris "Education with ICT in South Korea and Chile", International Journal of Educational Development, volume 31, pages 126-148, 2011.
[Savaş-2010]
- Savaş, K. & Erdal, H. "Automatic control simulation environment system (ACSES) designed as a virtual tool for control education", Procedia - Social and Behavioral Sciences, volume 2, pages 5233-5237, 2010.
[Schaf-2009]
- Frederico M.Schaf, Carlos Eduardo Pereira "Integrating Mixed-Reality Remote Experiments Into Virtual Learning Environments Using Interchangeable Components", IEEE Transactions on Industrial Electronics, volume 56, Issue: 12, pages 4776-4783, 2009.
[Schmees-2006]
- Schmees, M. "Organizing technology enhanced learning", Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet, ACM, pages 139-150, 2006.
[Schumann et al.-2012]
- Schumann, J. H.; Wunderlich, N. V. & Wangenheim, F. "Technology mediation in service delivery: A new typology and an agenda for managers and academics", Technovation, volume 32, pages 133-143, 2012.
[Settouti-2006]
- Lotfi Sofiane SETTOUTI "Systèmes à Base de Traces Modélisées : Modèles et Langages pour l'exploitation des traces d'Interactions", thèse de doctorat en informatique, Université Claude Bernard Lyon 1, 14 janvier 2006.
[Settouti et al.-2007]
- Lotfi Sofiane Settouti, Yannick Prié, Jean-Charles Marty, Alain Mille "Vers des Systèmes à Base de Traces modélisées pour les EIAH", LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/École Centrale de Lyon, 2007.
[Shlaer-1988]
- Shlaer, S., Mellor, S.J., "Object-Oriented Systems Analysis: Modeling the World in Data", Yourdon Press, Englewood Cliffs, N.J., 1988.
[Siemens-2001]
- "Using the ProcessSuite(TM) 4-mation(TM) Configuration Software-Version 4.30 or Higher", Siemens Energy & Automation, Inc., 2001.
[Siemens-2009]

- "SIMATIC PCS 7 APACS+ ES Library User's Guide for Version 7.1", Siemens, 2009.
- [Simon et al.-2008]
Tobias Simon, Jonathan Streit, and Markus Pizka, "Practically Relevant Quality Criteria for Requirements Documents", Software Engineering Research and Practice, CSREA Press, pages 115-121, 2008.
- [Sivakumar et al.-2005]
S. Sivakumar, W. Robertson, M. Artimy, N. Aslam, "A web-based remote interactive laboratory for internetworking education" IEEE Transactions on Education, vol. 48, no. 4, pp. 586–598, 2005.
- [Spalter-2000]
Spalter A. M., Simpson R. M. "Integrating interactive computer-based learning experiences into established curricula: a case study", Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSEconference on Innovation and technology in computer science education, ACM, pages 116-119, 2000.
- [Stouffer et al.-2011]
Keith A. Stouffer, Joseph A. Falco, K. A. S. "Guide to Industrial Control Systems (ICS) Security: Supervisory Control and Data Acquisition (SCADA) Systems, Distributed Control Systems (DCS), and Other Control System Configurations Such As Programmable Logic Controllers (PLC)", National Institute of Standards & Technology, Gaithersburg-United States, 2011.
- [Stevens-2003]
Stevens P., Weale M. "Education and Economic Growth", National Institute of Economic and Social Research, 2003.
- [Supavita-2009]
Siros Supavita "Object-Oriented Software and UML-Based Testing: A Survey Report", CiteSeerX - Scientific Literature Digital Library and Search Engine, United States, 2009.
- [SysML-2007]
Object Management Group, Inc. "OMG Systems Modeling Language (OMG SysML), v1.0", OMG, 2007.
- [Tabor-2007]
Tabor, Sharon W. "Narrowing the Distance: Implementing a Hybrid Learning Model", Quarterly Review of Distance Education (IAP), volume 8(1), pages 47-57, ISSN 1528-3518, 2007.
- [Tao-2001]
Tao Y.-H., Guo S.-M. "The design of a web-based training system for simulation analysis", Proceedings of the 33rd conference on Winter simulation, IEEE Computer Society, pages 645-652, 2001.
- [Thillou-2002]
Thillou Céline "La réalité augmentée : avec application en vidéoconférence", Stage réalisé dans les bureaux de Suresnes (92), Réseaux et Systèmes audiovisuels-ESIGETEL-3A, Dassault Systèmes, 2002.
- [Tran et al.-2010]
Tran Chi Dung, Sébastien George, Iza Marfisi-Schottman, "EDoS: An authoring environment for serious games design based on three models", 4th European Conference on Games Based Learning ECGBL2010, Copenhagen, Denmark, pages 393-402, 21-22 October 2010.
- [Turki-2008]
Skander TURKI "Ingénierie système guidée par les modèles: Application du standard IEEE 15288, de l'architecture MDA et du langage SysML à la conception des systèmes

- mécatroniques", Thèse de doctorat en Sciences et techniques industrielles, Université du Sud Toulon-Var, 2 Octobre 2008.
- [UML_Infra-2004]
"Unified Modeling Language (UML) Specification: Infrastructure-version 2.0", OMG Adopted Specification ptc/04-10-14. Novembre 2004.
- [UML2-2005]
O.M.G. UML v.2.0, OMG specifications. Document formal, 2005.
- [Unity_Pro-2011]
"Unity Pro: Languages and program structure Reference manual", Schneider Electric, July 2011.
- [Vandyk-2010]
Todd Tomlinson, John K. VanDyk "Pro Drupal 7 Development", Apress, 2010.
- [Wang-2011]
Wang, T.-H. "Implementation of Web-based dynamic assessment in facilitating junior high school students to learn mathematics", Computers & Education, volume 56, pages 1062-1071, 2011.
- [Wagner et al.-2008]
Wagner Nicole, Hassanein Khaled, Head Milena "Who is responsible for E-Learning Success in Higher Education? A Stakeholders' Analysis", Educational Technology & Society, volume 11, No 3, pages 26-36, 2008.
- [Weber-2011]
David M. Weber, Robert J. Kauffman "What drives global ICT adoption? Analysis and research directions", Electronic Commerce Research and Applications, volume 10, pages 683-701, 2011.
- [Wessner-2001]
Martin Wessner, Hans-Rüdiger Pfister "Group formation in computer-supported collaborative learning", Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, ACM, pages 24-31, 2001.
- [Williams et al.-2004]
Williams J. M., Cale J. L., Benavides N. D., Wooldridge J. D., Koenig A. C., Tichenor J. L., Pekarek S. D. "Versatile hardware and software tools for educating students in power electronics", IEEE Transactions on Education, volume 47, pages 436-445, 2004
- [Wirfs-Brock-1990]
Wirfs-Brock, R., Wilkerson, B., Wiener, L. "Designing Object Oriented Software", Prentice-Hall, Englewood Cliffs, N.J., 1990.
- [Wiseman-2012]
Wiseman A. W., Anderson E. "ICT-integrated education and national innovation systems in the Gulf Cooperation Council (GCC) countries", Computers & Education, volume 59, pages 607-618, 2012.