

N° d'ORDRE : 02/2008-E/IN

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

Université des Sciences et de la Technologie
"Houari Boumediène"

FACULTE D'Electronique et d'Informatique



THÈSE

Présentée pour l'obtention du diplôme de DOCTORAT D'ETAT

EN : INFORMATIQUE

Spécialité : Informatique

Par : Latifa MAHDAOUI

Sujet

Proposition d'une Infrastructure Statique et Dynamique pour le Support du Travail Collaboratif : « Application au tutorat dans le e-learning »

Soutenue le 14/12/2008, devant le jury composé de :

Mr- M. AHMED NACER,	Professeur, USTHB,	Président
Mme- Z. ALIMAZIGHI,	Professeur, USTHB,	Directrice de thèse
Mr- M. OCCELLO,	Professeur, Univ. Valence,	Examineur
Mr- Y. AIT-AMEUR,	Professeur, ENSMA Poitiers,	Examineur
Mme- A. AISSANI-MOKHTARI,	Professeur, USTHB,	Examinatrice
Mr- C.B. BENYELLES,	Professeur, IUT Valence,	Examineur
Mr- A. BELKHIR	Maitre de Conférence, USTHB,	Examineur
Mr- A. BALLA,	Maître de Conférence, INI,	Examineur

*Al-Hamdou Lillah,
A mes très chers parents, à qui je dois tout après Allah
et son noble prophète Mohamed "Prière et Salut sur Lui",
A mes sœurs et mes frères.*

- Remerciements -

Je tiens tout d'abord à remercier le Professeur Zaia ALIMAZIGHI pour m'avoir accueillie dans son équipe et pour la confiance qu'elle m'a fait en me laissant donner libre court à mes idées afin de mener ce travail. Je la remercie aussi pour tous les conseils, les critiques et les orientations qu'elle m'a prodigué à chaque étape importante du projet de thèse.

Je remercie également les membres du jury qui m'ont honorés en acceptant de juger ce travail, je cite : le Professeur Mohamed AHMED-NACER, le Professeur Aicha MOKHTARI-AISSANI, le Professeur Michel OCCELLO, le Professeur Yamine AIT-AMEUR, le Docteur Abdelkader BELKHIR et le Docteur Amar BALLA. Mes plus vifs remerciements vont aussi au Professeur Choukri-Bey BEN-YELLES.

Je remercie à titre personnel, le Professeur Yamine AIT-AMEUR pour toute l'aide qu'il m'a procuré et l'intérêt qu'il a porté à ce travail.

Durant toute la période de ce projet de thèse, beaucoup d'étudiants y ont participé, dans le cadre de leur projet de fin d'études. Je tiens à les remercier chaleureusement, je cite (par ordre alphabétique) : ALLAM Nawel, ATALLAH Hamid, BENLEBBAD Dalal, BENTAOUA Kamel, CHENAOUI Zine El-Abidine, EL-KOLLI Kahina, HAMADOUCHE Abdelmalek, HAVYARIMANA Louis, LAOUISSET Brahim, LEFTISSI Samir, LUYEYE Toko, MAGHBOUNE Mohamed, MAICHE Leila, MEGHEZZI Hanane, MEGHLOUI Salah MERTIL Madiha, OULDACHE Oussama, OURAHMOUNE Amel, SI BACHIR Khaled, TOUAZI Assia et ZIDOUNI Ratiba.

Je remercie aussi l'équipe ISI (Ingénierie des Systèmes d'Information) pour leur soutien moral et leurs encouragements, je cite : ABDAT Nadia, BOUKHALFA Kamel, BOUKHEDOUMA Saida, DERBAL Khalissa, DJIROUNE Rahma, HAMDIAH Mohamed et SELMOUNE Nazih. Merci aussi à tout collègue ou toute personne m'ayant aidé ou encouragé.

Enfin, un très grand signe de reconnaissance à tous mes enseignants depuis le primaire jusqu'au dernier en date.

Résumé

Dans ce travail, notre intérêt se porte sur l'usage des technologies du workflow à orientation « métier » au sens large du terme, i-e, concernant toute organisation qui n'est pas forcément, une entreprise. On voudrait pouvoir appliquer les technologies ayant fait leur preuve en milieu d'entreprise dans des organisations dédiées à l'enseignement, en particulier l'enseignement à distance. D'un point de vue système d'information, une institution d'enseignement est une organisation pourvue d'une mission d'éducation et/ou de formation, d'un ensemble d'acteurs ayant des responsabilités, représentant ses « compétences » et « son savoir-faire » donc son « métier », ainsi que d'un ensemble de moyens lui permettant d'accomplir ses tâches. Par analogie, un dispositif de formation à distance utilisant les TIC(s) peut être vu comme une institution virtuelle de formation faisant intervenir un ensemble d'acteurs qui coopèrent entre eux dans le but d'assurer des enseignements.

Un environnement de e-learning est alors formé par trois communautés d'acteurs interagissant entre eux via la plate forme, formant ainsi un Système d'Information Coopératif (SIC). Ces communautés sont les enseignants, les étudiants et les managers (ou administrateurs). Les liens de coopération peuvent être retrouvés à différents niveaux et pour divers objectifs. Le e-learning a tendance à spécifier les rôles et à bien les définir car il n'y a pas cette interaction directe qui n'est d'ailleurs remplaçable par aucun média. De plus, chaque rôle est susceptible d'être joué par une personne différente d'où la difficulté d'assurer un enseignement cohérent. On s'intéresse alors à la relation qu'il y a entre des tuteurs et leurs apprenants dans une plate forme de e-learning. L'idée consiste à modéliser le processus de tutorat distant liant les apprenants et leur tuteur par des modèles de Workflow. Un processus de tutorat distant concerne une matière enseignée dans le cadre d'un cursus de formation. En plus, le tutorat peut concerner un guidage personnalisé pour chaque apprenant ou alors un guidage collectif pour une équipe d'apprenants. L'atout principal derrière l'usage du Workflow serait une meilleure coordination des échanges entre les acteurs impliqués et donc un guidage plus efficace des apprenants.

Le présent travail consiste donc à fournir une infrastructure statique et dynamique permettant de modéliser des processus de tutorat en e-learning par l'usage d'un méta-modèle permettant de générer les modèles et de les exécuter. La partie organisationnelle du méta-modèle est enrichie par la notion d'acteurs automatiques qui sont des agents logiciels dans notre cas et dont la mission principale est de fournir l'assistance nécessaire aux acteurs « Apprenant » et « Tuteur ». Ceci se fait en impliquant les agents durant l'exécution de certaines activités et/ou tâches d'apprentissage ou de tutorat. L'ajout de cette dimension d'assistance automatique permettra de palier à l'inconvénient dû à l'exécution asynchrone des processus de tutorat car les apprenants et les tuteurs ne sont pas tous forcément connectés en même temps.

☞ Table des Matières ☛

Avant Propos	
Introduction Générale	i

Partie I : Etat de l'Art

Chapitre I : Les Systèmes Coopératifs et les Workflows

I.1 - LE SYSTEME D'INFORMATION	1
I.1.1 - Définition	1
I.1.2 - Passage d'une culture « <i>informatique de gestion de données</i> » à une culture « <i>informatique de gestion d'information</i> ».	2
I.1.3 - Passage de la culture verticale à la culture horizontale ou « l'entreprise aplatie »	4
I.2 - LA COOPERATION	6
I.2.1 – Définitions	6
I.2.2 - Les formes de coopération	8
I.2.3 - Mécanismes de la coopération	9
I.3 - MODALITES DE LA COOPERATION	9
I.4 - TRAVAIL COOPERATIF ASSISTE PAR ORDINATEUR (TCAO)	10
I.4.1 - Le travail coopératif	10
I.4.2 - Les outils du TCAO : le Groupware et le Workflow	12
I.4.2.1 - Définitions du groupware	13
I.4.2.2 - Au cœur du Groupware : le Workflow	13
I.5 - LE WORKFLOW	14
I.5.1 - Origines du workflow	14
I.5.2 - Définitions	15
I.5.3 - Workflow et processus d'une organisation	15
I.5.3.1 - Définition d'un processus	15
I.5.3.2 - Modèle de processus	16
I.5.3.3 - Taxonomie des processus	16
I.5.3.4 - Langages de modélisation des processus métier	17
I.5.4 - Systèmes de Gestion de Workflow (SGWF)	18
I.5.4.1 - Définition	18
I.5.4.2 - Environnements d'un SGWF	18
I.5.4.3 - Classification des SGWF	19
I.5.5 - Modèle de référence des SGWF(s)	22
I.5.5.1 - Le modèle de référence de la WfMC pour les SGWF(s)	22
I.5.5.2 - Les composantes de base du modèle	23
I.5.5.3 - Description d'un moteur de Workflow (MWF)	23
I.5.5.4 - Les interfaces du modèle de référence	26
I.5.6 - Architecture client/serveur d'un SGWF	27
I.5.7 - Conceptualisation du Workflow	29
I.5.7.1 - Les concepts de base	29

I.5.7.2 - Les concepts complémentaires -----	30
I.5.8 – Flexibilité et Workflow ou Workflow avancé -----	30
I.5.9 - Interopérabilité des SGWF(s) -----	31
I.5.9.1 - Définition -----	31
I.5.9.2 - Types d'interopérabilité de SGWF -----	31
I.5.10 - Le Web-Service -----	32
I.5.10.1 - Définition -----	33
I.5.10.2 - Le Modèle fonctionnel du Web-Service -----	33
I.5.10.3 - Les technologies des web services -----	34
I.5.10.4 - Composition de Web services -----	37
I.5.11 - Exemples d'outils dédiés au workflow -----	39
I.5.11.1 - W4 Suite (World Wide Web Workflow) -----	39
I.5.11.2 - Oracle Workflow -----	41
I.6 - CONCLUSION -----	45

Chapitre II : Les Systèmes Multi-Agents et leur Application au Domaine de l'enseignement

II.1 - LE PARADIGME AGENT (DANS LE SENS ORIGINE) -----	47
II.1.1 - Domaines de l'intelligence Artificielle (IA) et de l'intelligence Artificielle Distribuée (IAD) -----	47
II.1.2 - Programmation Orientée Agent (POA) -----	48
II.2 - L'AGENT -----	50
II.2.1 - Définitions -----	50
II.2.2 - Structure générale d'un agent -----	51
II.2.3 - Déterminant d'un agent -----	51
II.2.4 - Typologies d'agents -----	52
II.2.4.1 - Les agents réactifs -----	53
II.2.4.2 - Les agents cognitifs ou délibératifs -----	54
II.2.4.3 - Les agents cognitifs par opposition aux agents réactifs -----	57
II.3 - LES SYSTEMES MULTI-AGENTS -----	58
II.3.1 – Définition -----	59
II.3.2 - Environnement & Interaction -----	59
II.3.3 - Société d'agents -----	60
II.3.3.1 - L'organisation sociale -----	60
II.4 - LA COMMUNICATION DES AGENTS -----	61
II.4.1 - Protocole de communication -----	61
II.4.2 - Langage de communication entre agents -----	62
II.4.3 - Architecture de communication -----	62
II.4.4 - Modes de communication -----	63
II.5 - CONTROLE ET PRISE DE DECISION AU NIVEAU DES SMA(S) -----	63
II.6 - LA COOPERATION AU NIVEAU DES SMA(S) -----	63
II.6.1 - Modes de Coopération -----	64

II.7 - ARCHITECTURES D'AGENTS -----	66
II.7.1 - Structures des agents -----	66
II.7.1.1 - La rationalité des agents -----	66
II.7.1.2 - Structure conceptuelle des agents -----	66
II.7.1.3 - Modélisation des agents -----	67
II.7.2 - Type d'architectures des agents intelligents -----	68
II.7.2.1 - Architecture réactive -----	68
II.7.2.2 - Architecture hybride -----	68
II.7.2.3 - Architecture BDI -----	69
II.7.2.4 - Architecture JAM -----	73
II.8 - METHODOLOGIES DE DEVELOPPEMENT ORIENTE AGENTS -----	76
II.8.1 - Ingénierie logicielle orienté agent -----	76
II.8.1.1 - L'approche orientée agent en ingénierie logicielle -----	77
II.8.1.2 - Méthodologies orientées agent -----	78
II.9 - LES SYSTEMES TUTORIELS INTELLIGENTS (STI) -----	85
II.9.1 - Enseignement par ordinateur traditionnel versus systèmes tutoriels intelligents	86
II.9.2 - Structure d'un STI -----	86
II.10 - LES ENVIRONNEMENTS INTERACTIFS POUR L'APPRENTISSAGE HUMAIN (EIAH) -----	88
II.11 - CONCLUSION -----	89

**Partie II : Une Infrastructure Statique et Dynamique pour le Travail Coopératif
appliqué au tutorat en e-learning**

**Chapitre III : Une démarche à base de Méta-Modèle de Workflow
pour le Tutorat en e-learning**

III.1 - DEFINITIONS & CONCEPTS DE BASE -----	94
III.1.1 - Famille, Groupe & Equipe -----	94
III.1.2 - Objectifs Pédagogiques -----	94
III.1.3 - Tâche, Activité Individuelle & Activité Collective -----	95
III.1.4 - Plan d'Apprentissage Individuel (PAI) & Plan d'Apprentissage Collectif (PAC) -----	95
III.1.5 - Tutorat Individuel & Tutorat Collectif -----	96
III.1.6 - Processus e-Learning ou de Tutorat Distant -----	96
III.1.7 - Adaptabilité et adaptativité -----	97
III.2 - LA DEMARCHE DE MODELISATION A BASE DE META-MODELE ----	97
III.2.1 - Analyse des besoins (Elaboration du cahier des charges) -----	99
III.2.2 - Analyse Conceptuelle (Conception des processus de tutorat distant) -----	99
III.2.3 - Le Design (Construction et Implémentation des modèles de processus de tutorat distant) -----	100
III.2.4 - Instanciation des modèles de processus e-learning -----	100

III.3 - MÉTA-MODELE DE WORKFLOW POUR LE TUTORAT EN E-LEARNING -----	100
III.3.1 - Aspect organisationnel -----	101
III.3.2 - Aspect fonctionnel -----	102
III.3.3 - Aspect comportemental -----	106
III.3.3.1 Etats des activités et des tâches -----	108
III.3.3.2 - Les conditions de transition et les blocs d'exécution -----	110
III.3.4 - Aspect informationnel -----	113
III.4 - PRÉSENTATION D'UNE ÉTUDE DE CAS : TUTORAT DISTANT POUR UN TP DU COURS "SYSTÈME D'INFORMATION" -----	114
III.4.1 - Phase 1 : Analyse des besoins -----	114
III.4.2 - Phase 2 : Analyse Conceptuelle -----	115
III.4.3 - Phase 3 : Construction des modèles de "TP-SI-01" -----	131
III.4.4 - Phase 4 : Instanciation des modèles de "TP-SI-01" -----	134
III.5 - FONCTIONNEMENT GLOBAL DU SYSTÈME PROPOSÉ -----	135
III.5.1 - Quels sont les fonctionnalités du système ? -----	135
III.5.2 - Les scénarios du système de tutorat distant -----	136

Chapitre IV : Un Système d'Assistance à Base d'Agents pour le Tutorat Distant

IV.1 - PRESENTATION GENERALE DU SYSTEME D'ASSISTANCE -----	142
IV.2 - EXTENSION DE L'ASPECT ORGANISATIONNEL DU META-MODELE AUX ACTEURS ARTIFICIELS -----	143
IV.2.1 - Description des acteurs du SMA -----	144
IV.2.1.1 - LAA : l'agent assistant de l'apprenant -----	144
IV.2.1.2 - PTA : l'agent tuteur personnel de l'apprenant -----	145
IV.2.1.3 - PA : l'agent pédagogique -----	146
IV.2.1.4 - TAA : l'agent assistant du tuteur -----	147
IV.2.1.5 - GTA : l'agent assistant d'un groupe d'apprenants -----	148
IV.2.1.6 - STA : l'agent super tuteur -----	148
IV.2.1.7 - CMA : l'agent gestionnaire de connexion -----	149
IV.3 - PROPRIETES DES AGENTS DU SMA (SELON LA GRILLE DE J. FERBER) -----	149
IV.4 - LIEN ENTRE LE SYSTÈME WORKFLOW ET LE SMA -----	151
IV.5 - CONCEPTION DU SMA -----	152
IV.5.1 - Démarche de conception du SMA -----	152
IV.5.3.4 - Sollicitation d'assistance par un apprenant -----	152
IV.5.2 - Fonctionnalités du SMA -----	152
IV.5.3 - Description des scénarios par les diagrammes de séquence -----	153
IV.5.3.1 - Connexion d'un apprenant -----	154
IV.5.3.2 - Suivre le déroulement d'un PAI -----	155
IV.5.3.3 - Suivre le déroulement d'activités -----	156
IV.5.3.4 - Sollicitation d'assistance par un apprenant -----	158
IV.5.4 - Diagramme d'état/transition -----	160

IV.5.5 - Diagramme de classes -----	162
IV.5.5.1 - Structure de l'agent LAA -----	164
IV.5.5.2 - Structure de l'agent PTA -----	167
IV.5.5.3 - Structure de l'agent PA -----	169
IV.5.5.4 - Structure de l'agent TAA -----	171
IV.5.5.5 - Structure de l'agent GTA -----	174
IV.5.5.6 - Structure de l'agent CMA -----	176
IV.6 – CONCLUSION -----	177

Chapitre V : Mise en Œuvre d'Outils Dédiés à l'Infrastructure Proposée

V.1 – ARCHITECTURE GENERALE DE L'INFRASTRUCTURE -----	178
V.2 - L'OUTIL AB-LEARNING 1.0 -----	179
V.3 - L'OUTIL E-TUTORING 1.0 -----	182
V.4 – CONCLUSION -----	188

Conclusion Générale -----	
Bibliographie -----	
Index des Figures -----	
Index des Tables -----	
Annexes -----	

Introduction Générale

Il n'est plus à redire aujourd'hui que les Technologies de l'Information et de la Communication (TIC) ont opérés une véritable révolution tant sur le plan des méthodes de travail que sur celui de la réflexion. Grâce à elles, on assiste actuellement à une réelle reconsidération, voir reconstruction de différents secteurs et domaines de la vie. Des concepts émergents comme la e-gouvernance, l'e-administration, la e-médecine ou encore la e-éducation en sont la preuve.

Au tout début, ce fut le secteur du business et du management qui fut le premier à bénéficier des avancées technologiques d'abord en matière d'informatique puis ensuite en matière d'information et de communication. Cela est un peu légitime étant donné que ce secteur a été le principal promoteur du développement de ces technologies afin de répondre à des besoins liés au métier des entreprises et à leurs performances en matière de qualité de service et de management. Les méthodes de management telles que le CPI ou le BPR¹ (Saikali, 2001) ont fait des technologies de Groupware leur principal outil de travail. Ainsi le CPI est utilisé pour améliorer ce qui existe et fonctionne déjà alors que le BPR tend vers la re-conception des processus avec toutes les implications sur les niveaux organisationnels et décisionnels notamment.

D'un autre côté, les NTIC² se sont rapidement trouvées promues au grade de TIC tout simplement, car elles s'inscrivent plus aujourd'hui dans la sphère des technologies « **innovantes** » qui poussent à repenser les processus de travail au sens large du terme. La question donc n'est plus d'introduire les TIC(s) dans les domaines de la vie « coûte que coûte », mais plutôt de le faire de manière judicieuse dans le but d'améliorer l'existant et le cas échéant le reconcevoir.

1. Les systèmes d'information coopératifs :

Les systèmes d'information classiques, à hiérarchie verticale, se sont progressivement fait remplacer dans certains domaines par des hiérarchies plus souples tendant à s'aplanir étant donné qu'une organisation n'est plus décrite par les fonctions qu'elle remplit dans le cadre de sa structure organisationnelle mais bien au travers des prestations qu'elle assure que ce soit vers son environnement extérieur (ses clients) ou en son intérieur. On parle alors de Systèmes d'Information Coopératif (SIC). Même si les définitions de ce qu'est un SIC ne font pas l'unanimité dans la communauté de recherche à cause de la diversité des points de vue et des spécialités impliquées, il n'en reste pas moins qu'un consensus existe sur le fait qu'un tel système implique des acteurs qui coopèrent via des supports technologiques appropriés. Par exemple, (Saadouné & Al, 1996) définissent un SIC comme « *des liens que construisent entre eux des agents en vue de réaliser, volontairement, une œuvre commune* ».

La littérature abonde de classifications visant à définir les outils appropriés à mettre en œuvre ou à utiliser dans un cadre de coopération. Certaines définitions tendent à définir la coopération par les trois fameux « C » du Groupware que sont : « Communiquer », « Collaborer » et « Coordonner » (Khoshafian, 1998), (Saadouné &

¹ CPI: Continuous Process Improvement. BPR: Business Process Reengineering.

² Nouvelles Technologies de l'Information et de la Communication.

Al, 1996). Au cœur du Groupware se trouve la catégorie du TCAO ou CSCW³ auquel on s'intéresse dans le cadre de ce travail de thèse. Ainsi, d'après (Baecker, 1993), «*Le CSCW est une activité, coordonnée assistée par ordinateur, comme la communication ou la résolution de problèmes effectuée par un groupe d'individus qui coopèrent, le Groupware est le logiciel multiutilisateurs qui assiste le CSCW*». L'outil de coordination favori dans le domaine du TCAO est le Workflow que (Saikali, 2001) définit comme «*la partie des Technologies de l'Information qui s'occupe de la coordination des acteurs des processus des entreprises, processus dont ils assurent également l'exécution et la gestion*».

Un processus est donc une procédure de travail visant à atteindre un objectif donné. Cette procédure a un début et une fin et engage un ensemble d'acteurs et de ressources autour d'elle au vu de son accomplissement. D'après (MEDINA-MORA et al, 1992), les processus peuvent être de nature « matériel » visant la production d'objets physiques, « d'information » supporté par exemple par les SGBD ou les systèmes à transaction ou encore « métier » qui reflètent donc la raison d'exister d'une organisation et son savoir faire. Les méthodes de management classent comme « Critique » la catégorie de processus « métier » vu qu'ils représentent un peu l'âme d'une organisation.

Un processus modélisable par les Workflows est dit automatisable. Son automatisation nécessite la prise en compte de deux aspects complémentaires que sont la « statique » et la « dynamique » du processus. La partie statique est représentée par le système d'information décrivant l'environnement et les données du processus alors que la partie dynamique est représentée par les modèles de Workflows. Le système de gestion du Workflows a alors la charge de veiller à la bonne exécution des processus « cas de workflow » en assurant une bonne intégration des aspects statiques et dynamique. L'utilisation des systèmes Workflows a subi plusieurs évolutions en vue de répondre aux besoins des entreprises notamment en matière de capacité de prise en charge des changements. D'après (Saikali, 2001), dans le workflow avancé ou adaptatif, il faut distinguer la flexibilité de l'adaptativité en ce sens que la première est vue comme une gestion d'exception au niveau des instances ou « workflow case » et la seconde comme un changement du modèle de processus workflow

Les Workflows avancés s'inscrivent dans le cercle des Workflows évolutifs visant à offrir le maximum de flexibilité aussi bien au niveau de l'adaptation des modèles de processus que celui de la gestion des exceptions susceptibles d'apparaître durant l'exécution dans processus (instance de workflow). Si la majorité des systèmes Workflows du marché prennent bien en charge l'aspect gestion des exceptions, la partie concernant l'évolution des modèles nécessite des mécanismes plus élaborés. L'approche par méta-modèle constitue un de ces mécanismes car ces derniers captent l'essentiel de tous les concepts reliés à la modélisation des processus en général et en particulier ceux des processus de nature Workflow.

Donc, un méta-modèle de workflow a pour objectif de proposer un cadre conceptuel pour la modélisation des processus Workflows indépendamment des outils et des plateformes qui seront utilisés pour l'implémentation. Ceci est moins contraignant pour les concepteurs et offre la possibilité de modéliser des processus Workflows non forcément destiné à un environnement orienté business.

³ « Travail Coopératif Assisté par Ordinateur » ou « Computer Supported Collaborative Work ».

2. Le contexte du e-learning :

Notre intérêt se porte sur l'usage des technologies du workflow à orientation « métier » au sens large du terme, i-e, concernant toute organisation qui n'est pas forcément, une entreprise. On voudrait pouvoir appliquer les technologies ayant fait leur preuve en milieu d'entreprise dans des organisations dédiées à l'enseignement, en particulier l'enseignement à distance.

D'un point de vue système d'information, une institution d'enseignement est une organisation pourvue d'une mission d'éducation et/ou de formation, d'un ensemble d'acteurs ayant des responsabilités, représentant ses « compétences » et « son savoir-faire » donc son « métier », ainsi que d'un ensemble de moyens lui permettant d'accomplir ses tâches. Par analogie, un dispositif de formation à distance utilisant les TIC(s) peut être vu comme une institution virtuelle de formation faisant intervenir un ensemble d'acteurs qui coopèrent entre eux dans le but d'assurer des enseignements.

L'introduction des TIC(s) dans ce domaine a donné naissance aux TICE(s)⁴ (Peraya & Al., 2004). Notre choix s'est porté sur une des formes de formation à distance qu'est le e-learning. (Kozman, 1991) définit le e-learning comme suit : « *Le e-learning est un mode d'apprentissage basé sur l'utilisation des nouvelles technologies, permettant l'accès à des formations en ligne, interactive et parfois personnalisées, diffusés par l'intermédiaire d'Internet, d'un Intranet ou toute autre média électronique, afin de développer les compétences, tout en rendant le processus d'apprentissage indépendant de l'heure et de l'endroit* ». Le e-learning est supporté par des plates formes logicielles qui lui sont dédiées communément appelées « Plate forme de e-learning » ou LMS⁵.

Un environnement de e-learning est alors formé par trois communautés d'acteurs interagissant entre eux via la plate forme, formant ainsi un SIC. Ces communautés sont les enseignants dont la mission est d'enseigner, les étudiants dont la mission est d'étudier et les managers (ou administrateurs) dont la mission est d'assurer toutes les fonctions de gestion liées aux communautés adhérentes, aux contenus et ressources et au système technologique qui supporte le e-learning. Les liens de coopération peuvent être retrouvés à différents niveaux et pour divers objectifs.

2.1. Processus de e-learning :

Une analyse des processus intrinsèques à un tel environnement d'enseignement permet d'en dégager à notre sens deux principales catégories que sont : les processus liés à l'*activité d'enseignement* en général et ceux liés à l'*activité de management*. Comme exemple de la première catégorie, nous pouvons citer le déroulement d'une session de formation dans une matière ou domaine donné engageant des étudiants et un staff pédagogique ou encore la préparation collective d'un cours par une équipe d'experts. Comme exemple de la deuxième catégorie, les processus de gestion des adhérents à une plate forme ou encore celui de la gestion de contenus.

⁴ TIC pour l'Enseignement ou l'Education.

⁵ Learning Management System.

Bien que dans la majorité des cas, le rôle de la communauté des étudiants est souvent décrit par celui d'« *Apprenant* », divers rôles émergent des deux autres communautés, à titre d'exemple nous avons les rôles de « *Tuteur* », « *Auteur* » ou « *Examineur* » pour les enseignants, et celui de « *manager de contenu* » ou « *d'adhérents* » pour les administrateurs. La question qui se pose alors est : « *Y a-t-il parmi les processus e-learning liés à l'enseignement ceux de nature à être automatisés, donc modélisés par le Workflow ?* ». Les travaux existant dans la littérature comme (Sébastien & Al., 2001) et (Ole & Al., 2003) nous apportent très vite la réponse. Ils proposent un environnement à base de pédagogie de projet pour supporter le travail collectif en milieu d'apprentissage. (Sadiq & Al., 2002) recommandent l'usage du Workflow dans le domaine de l'éducation pour gérer des équipes pédagogiques de rédaction de contenus, (Vantroys & Al., 2002) quant à eux proposent un environnement à base de workflow pour l'exécution et le suivi ou « *tracking* » de travaux d'apprentissage individuels. Enfin (Cesarini & Al., 2004) présentent un moteur de workflow supportant des processus d'apprentissage.

Mais la principale remarque que l'on peut faire est que les automatisations proposées ne sont pas liées à l'acte d'enseigner lui-même qui par nature est interactif. De plus, la majorité des normes et outils existants se focalisent plus sur le contenu de l'enseignement que sur la relation d'enseigner elle-même. En effet, dans un enseignement classique, l'enseignant joue différents rôles en même temps devant ses élèves, usant de ses connaissances, de son expérience et de son intuition pour redresser et adapter le parcours d'apprentissage. Un des rôles principaux de l'enseignant est celui de « *tuteur* ». Cette fonction fait la distinction, dans le e-learning, entre le concepteur de cours et celui qui suit l'apprenant, le guide, le conseille, le soutient, répond à ses questions et ses attentes tout au long de son parcours d'acquisition de connaissances. Un enseignement en e-learning ne peut se concevoir sans tuteur.

Le e-learning a donc tendance à spécifier les rôles et à bien les définir car il n'y a pas cette interaction directe qui n'est d'ailleurs remplaçable par aucun média. De plus, chaque rôle est susceptible d'être joué par une personne différente d'où la difficulté d'assurer un enseignement cohérent. On s'intéresse alors à la relation qu'il y a entre des tuteurs et leurs apprenants dans une plate forme de e-learning. L'idée consiste à modéliser le processus de tutorat distant liant les apprenants et leur tuteur par des modèles de Workflow. Un processus de tutorat distant concerne une matière enseignée dans le cadre d'un cursus de formation. En plus, le tutorat peut concerner un guidage personnalisé pour chaque apprenant ou alors un guidage collectif pour une équipe d'apprenants. Cette deuxième approche s'inspire largement de ce qui se fait dans une classe réelle, mais elle est aussi confortée par différents travaux de recherches en la matière.

A cet effet, beaucoup d'ouvrages et de travaux de recherche se sont focalisés sur les aspects collaboratifs en milieu professionnels. (Saadoun & Al., 1996) soulève les concepts de groupe et d'équipe soutenus par les TIC(s). (Lewis, 1998) parle de « *Joint Learning* » (Apprentissage collaboratif) basé sur la théorie de l'activité introduite par le célèbre psychologue russe Vygotsky (Salomon, 1993). Ce dernier décrit l'apprentissage collaboratif comme un moyen incontournable d'acquérir des connaissances que l'on ne pourrait pas acquérir individuellement.

2.2. Assistance en e-learning :

Si l'usage des technologies reliées au travail coopératif est relativement récent dans le domaine du e-learning, celui relié au monde des agents ne s'est pas fait attendre car très vite l'on s'est aperçu de l'utilité d'utiliser des entités intelligentes dans un environnement dédié à l'apprentissage. En effet, l'Intelligence Artificielle (IA) en général et l'IAD⁶ en particulier ont trouvé dans le domaine pédagogique un terrain favori pour expérimenter différentes taxonomies d'agents allant des agents réactifs aux agents cognitifs.

Parmi les systèmes les plus connus, nous avons les Systèmes Tuteurs Intelligents (STI) qui se basent généralement sur une modélisation des domaines et acteurs impliqués dans la pédagogie tels que l'apprenant, le contenu, la démarche pédagogique ainsi que l'expert du domaine. L'ensemble du système est sensé assurer un tutorat automatique par l'interaction de ses différents composants. Ces systèmes ont cependant montré leur relative efficacité due essentiellement aux problèmes posés par l'IA elle-même.

D'autres approches sont nées de l'usage de l'IAD dont celle des EIAH⁷ se basant beaucoup sur la notion de Système Multi Agents (SMA) qui consiste à faire une répartition des rôles entre un ensemble d'agents. Le but étant d'offrir à l'apprenant un cadre d'apprentissage basé sur l'interactivité lui permettant de bénéficier du maximum de guidage possible (Pesty & al, 2001), (Mbala, 2003).

Cependant, le rêve de faire remplacer « complètement » le tuteur humain par un tuteur automatique s'est vite révélé être impossible car « Tutorer » c'est interagir avec l'apprenant en se basant entre autres sur son intuition, chose qu'un agent programmé ne pourra jamais faire aussi sophistiquée que sera sa conception. D'où l'orientation vers des systèmes de tutorat n'éliminant pas la composante « Tuteur-Humain », mais plutôt la confortant par des systèmes d'assistance sous forme de SMA.

En e-learning, le problème d'une assistance efficace reste posé notamment dans ses formes « *Blended Learning* » ou « *Asynchronous Learning* ». Dans le premier cas, l'enseignement est dispensé complètement via une plateforme avec pas ou vraiment très peu d'encadrement humain. Dans le second cas, le suivi y est périodique mais insuffisant pour assurer une bonne acquisition de connaissance. Ces deux formes sont donc celles qui nécessitent le plus d'efforts en matière d'assistance.

3. Notre Approche :

L'atout principal derrière l'usage du Workflow serait une meilleure coordination des échanges entre les acteurs impliqués et donc un guidage plus efficace des apprenants. Par ailleurs, une autre caractéristique importante dans l'acte d'enseigner en général, c'est la flexibilité, c'est-à-dire que nous n'enseignons jamais de la même manière. En effet, même dans une classe traditionnelle composée d'un nombre d'élèves limité, l'enseignant se trouve en face d'individus n'ayant ni le même niveau ni les mêmes aptitudes d'où son besoin de faire des réajustements et des adaptations de son

⁶ Intelligence Artificielle Distribuée

⁷ Environnement Interactif pour l'Apprentissage Humain.

enseignement. Nous voudrions alors pouvoir refléter les mêmes aspects dans un contexte de e-learning. Pour ce faire, nous avons adopté une approche par méta-modèle de workflow basée entre autres sur des travaux réalisés au sein de notre équipe de recherche (Alimazighi & Al., 2002) et ceux décrits dans (Saikali, 2001). Le méta-modèle est conforme aux normes du méta-modèle de la Workflow Management Coalition (*Wfmc*)⁸ qui est un consortium d'industriels, d'entreprises et d'institutions de recherche visant à standardiser les échanges entre différents systèmes de workflow.

Le méta-modèle enrichi des concepts liés au e-learning considère les processus de tutorat distant sous quatre facettes que sont l'organisation, le fonctionnement, le comportement et l'information. Il est conçu à l'aide du langage UML dont les bienfaits ne sont plus à redémontrer.

Se pose alors la question sur « *la démarche à suivre pour modéliser et mettre en œuvre de tels processus ?* ». Bien sûr, l'usage d'une des méthodes dédiées à la modélisation de processus workflow serait tout à fait possible, mais elle présenterait le désavantage d'utiliser une terminologie orientée plus vers le business. D'où nous préférons à cela une démarche qui s'inscrit plus dans le domaine de la pédagogie et dont le rapprochement avec la modélisation des processus permettra d'en faire une description plus rigoureuse. Le tutorat distant faisant partie des processus critiques, il est impératif de bien le prendre en charge au risque de voir s'anéantir la formation elle-même⁹. Dans les processus liés à l'enseignement, la mise en place de ce que certains spécialistes du e-learning appellent dispositifs de formation à distance que nous remplacerons volontiers par « dispositif de e-learning » (étant donné que ce dernier en fait partie), nécessite donc une méthodologie de conception de la même manière que l'on appliquerait une méthode de génie logiciel. Nous avons choisi d'appliquer certains aspects qui nous intéressent d'une démarche particulièrement intéressante dénommée ASPI¹⁰ dont les détails seront donnés en annexe de la thèse.

ASPI offre un cadre méthodologique ouvert permettant de spécifier toutes les facettes d'un dispositif de formation dans un sens beaucoup plus large que celui du tutorat distant. Elle couvre des aspects aussi divers que les activités pédagogiques en terme de temps et de délais, les scénarios d'apprentissage, le staff pédagogique impliqué, les ressources, les pré-requis et les méthodes d'évaluation. A partir de là, nous faisons une extraction des activités reliées au tutorat avec toutes les informations nécessaires pour modéliser le processus à l'aide des diagrammes d'activités.

Notre travail consiste donc à fournir une infrastructure statique et dynamique permettant de modéliser des processus de tutorat en e-learning par l'usage d'un méta-modèle permettant de générer les modèles et de les exécuter. La partie organisationnelle du méta-modèle est enrichie par la notion d'acteurs automatiques qui sont des agents logiciels dans notre cas dont la mission principale est de fournir l'assistance nécessaire aux acteurs « Apprenant » et « Tuteur ». Ceci se fait en impliquant les agents durant l'exécution de certaines activités et/ou tâches d'apprentissage ou de tutorat. L'ajout de cette dimension d'assistance automatique permettra de palier à l'inconvénient dû à l'exécution asynchrone des processus de tutorat car les apprenants et les tuteurs ne sont pas tous forcément connectés en même temps.

⁸ <http://www.wfmc.org>.

⁹ Le nombre d'échecs et d'abandons en e-learning en témoigne.

¹⁰ Accompagner, Soutenir et Piloter un projet d'Innovation techno-pédagogique.

4. Plan de la thèse :

Afin de mener ce travail à son aboutissement nous avons d'abord décrit un état de l'art constituant la première partie de la thèse et couvrant aussi bien les technologies workflow que celles des systèmes multi-agents notamment ceux appliqués dans le domaine de l'enseignement.

Dans la seconde partie de la thèse, nous abordons la problématique du tutorat distant par la définition des concepts y afférents et en proposant une démarche globale de modélisation. Nous présentons ensuite le méta-modèle de workflow pour le e-learning sans prise en compte des acteurs artificiels en détaillant les quatre aspects : organisationnel, fonctionnel, comportemental et informationnel. Pour la validation de notre méta-modèle, nous présentons une étude de cas détaillée dont le cahier des charges a été élaboré selon la démarche ASPI. Les détails de l'étude de cas concerne le tutorat à distance de TP réalisés dans le cadre de l'enseignement du module système d'information. Les détails du dispositif lui-même sous-forme de cahier des charges sont donnés en annexe C de la thèse. La partie étude de cas elle-même ne prendra en compte que les aspects reliés au processus workflow et mettra en évidence les phases de PAI pour plan d'apprentissage individuel et de PAC pour plan d'apprentissage collectif.

Suit alors un second volet proposant une description des acteurs artificiels formant un système multi-agents pour l'assistance aux apprenants et tuteurs humains dénommé « SMA-Assistance ». La conception du SMA s'est faite à l'aide du langage Agent UML (AUML)¹¹. Ce dernier constitue une extension d'UML en vue de supporter la modélisation d'agents. Le choix d'AUML comme langage de modélisation nous a permis de rester en conformité avec le méta-modèle de workflow, étant donné que la dynamique du SMA est décrite par les mêmes outils que ceux utilisés pour le reste des acteurs.

Nous terminons par la description d'une architecture fonctionnelle du système ainsi que quelques outils réalisés dans le cadre de ce projet.

¹¹ Agent UML : <http://www.auml.org>

Partie I

Etat de l'art



Chapitre I

Les Systèmes Coopératifs et les Workflows



Chapitre I

Les Systèmes Coopératifs et les Workflows

La problématique liée aux SIC (Systèmes d'Information Coopératifs) est présentée comme un nouveau domaine de recherche à la synergie des technologies des SI (Système d'Information), du TCAO (Travail Coopératif Assisté par Ordinateur) et des théories de la modélisation et de la planification organisationnelle. L'enjeu est d'élaborer des systèmes ayant pour objectif l'assistance de la coopération entre utilisateurs. Dans ce chapitre, nous proposons de traiter les deux composants clés des systèmes d'information coopératifs à savoir la coopération et les systèmes d'information, en passant bien évidemment en revue le TCAO ainsi que les workflows.

I.1 Le système d'information

I.1.1 Définition

Jusqu'à une époque récente, le développement des systèmes d'information se focalisait sur la nature organisationnelle de l'entreprise. De manière classique, l'entreprise est perçue comme un système au sens général.

Ce système regroupe des sous-systèmes qui agissent les uns avec les autres en fonction de *l'organisation générale* d'une part, c'est-à-dire, la répartition des tâches ou des fonctions sur les sous-systèmes, et d'autre part *la structure générale*, c'est-à-dire, le réseau de connexion entre les sous-systèmes.

Le système d'information (SI) est le reflet de l'entreprise, c'est un ensemble de moyens organisés devant servir la stratégie (la vision) et supporter les opérations quotidiennes (l'action) de l'entreprise. Il est considéré comme un système entre l'utilisateur et un ordinateur intégré qui produit de l'information pour assister les personnes dans les fonctions d'exécution, de gestion et de prise de décision.

Le SI utilise des équipements informatiques et des logiciels, des bases de données, des procédures manuelles, et des modèles pour l'analyse, la planification, le contrôle et la prise de décision. Ainsi le SI est un pont entre théories de l'organisation de l'entreprise et celles du traitement de l'information (LeMoigne, 1990) ce qui entraîne une confusion de la notion de SI au sens organisationnel et au sens informatique.

En sciences des organisations, on parle de système d'information organisationnel (SIO) qui se présente comme le système autour duquel s'organisent et se développent les activités de l'entreprise. Il constitue la mémoire de l'entreprise assurant l'apprentissage organisationnel. Le SIO se définit donc comme un outil intégré de performances organisationnelles, et d'une interface assurant le couplage opération-décision (LeMoigne, 1986).

En effet, la vision systémique de l'organisation de l'entreprise consiste à identifier trois sous-systèmes : un système de décision, un système d'opération et un système d'information.

- ✿ Le **système de décision** ou de pilotage a le rôle de contrôler et de prendre des décisions dans l'entreprise (stratégiques, tactiques et opérationnelles), à la connaissance d'événements provenant de son système d'information.
- ✿ Le **système opérant** ou de production exécute les tâches que le système de pilotage lui demande d'assurer, dans le but de produire (des biens ou des services). Les sous-systèmes de pilotage et de production s'échangent des flux d'informations ascendants et descendants.
- ✿ Le **système d'information** permet la communication entre ces deux systèmes et avec l'environnement. Il permet de recevoir et d'émettre des informations mais aussi de gérer les informations internes à l'entreprise.

La figure ci-dessous représente clairement la vision systémique de l'organisation de l'entreprise :

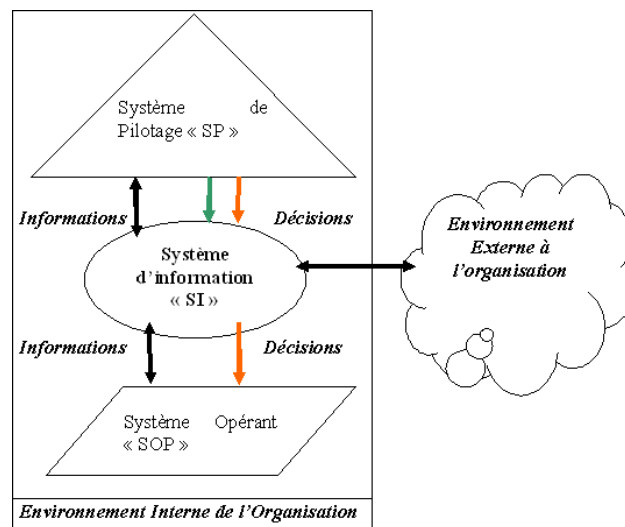


Figure I.1 : Représentation d'un SIO

Une des performances actuelles d'une organisation réside dans l'aptitude à corrélérer intelligemment l'activité stratégique et l'action opérationnelle (EncycMan, 1991). L'apport des sciences de l'informatique intervient dans la modélisation et la conception des SI(s) qui s'adaptent aux SIO(s) et qui assure cette corrélation.

Le SI est défini alors comme l'ensemble des ressources matérielles, humaines et informatiques contribuant à la collecte, la mémorisation, la recherche, la communication et l'utilisation des données nécessaires et suffisantes pour un pilotage et opérations au sein du système de l'entreprise (Tardieu & al, 1994).

Or, piloter l'entreprise c'est gérer un tout : un ensemble d'individus, de département, de directions, de plusieurs unités ou entités ayant chacune sa mission, son but, ses compétences, et par conséquent ses propres processus cognitifs (Baghdali, 1997).

Ainsi, certains auteurs comme (Alquier, 1993) (Courbon, 1993) affirment qu'il n'existe pas un seul et unique SI tel que préconisé par les méthodes classiques (Tardieu & al, 1994), mais plusieurs sous-ensembles du SI, qui ont des finalités différentes et sont conçus différemment. Dans (Alquier & al, 2000) on distingue trois sous-ensembles de SI selon une dimension dite « spatiale » :

- ✿ "SI collectif" : qui contient les données ayant une sémantique commune pour tous les membres de l'entreprise. Il constitue le langage de communication

compris pas tous. Il est généralement le support des objets de gestion fondamentaux de l'entreprise (clients, produits, ... etc.).

- ✿ "SI départementaux" : qui sont les langages spécialisés des différentes unités organisationnelles. Ils sont les supports des procédures de travail spécialisées.
- ✿ "SI individuels" : support de données individuelles utilisées pour l'aide à la décision individuelle.

Si les systèmes d'informations sont au cœur de tout changement, la raison est soit parce que le changement concerne directement ces systèmes, soit parce que le changement implique une action simultanée sur ces systèmes. Nombreuses sont les entreprises qui voient dans leurs systèmes d'information un véritable frein à leur flexibilité et à leur réactivité, elles se voient donc dans l'obligation d'effectuer les changements suivants :

- ✿ Le Passage d'une culture « *informatique de gestion de données* » à une culture « *informatique de gestion d'information* ».
- ✿ Le Passage de la culture verticale à la culture horizontale ou « l'entreprise aplatie ».

I.1.2 Passage d'une culture « *informatique de gestion de données* » à une culture « *informatique de gestion d'information* ».

La majorité des entreprises ne considèrent pas l'information comme un actif. Elles la stockent sur plusieurs systèmes, souvent incapables de communiquer entre eux. D'où la difficulté de partager les connaissances, la circulation et l'archivage de plusieurs informations essentielles sous des formes non intégrées : papier, téléphone, télécopie (Saadoun, 1996). Vient alors la nécessité du passage d'une culture « *informatique de gestion de données* » où l'informaticien est le principal acteur, à une culture « *informatique de gestion d'informations* » où les personnes sont les vrais et seuls acteurs.

L'informatique traditionnelle est en partie dépassée parce qu'elle se révèle incapable d'appréhender certaines exigences du monde tel qu'il se présente aujourd'hui. A ce niveau, il serait nécessaire de montrer la différence entre donnée, information et connaissance dont nous donnerons les définitions suivantes :

- ✿ Une **donnée** peut être définie comme étant un fait discret et objectif, résultat d'une acquisition, d'une mesure effectuée par un instrument naturel ou construite par l'homme. Elle peut être qualitative ou quantitative. Elle est objective et n'a pas de sens en elle-même.
- ✿ L'**information** est une collection de données organisées pour donner forme à un message (c'est une donnée additionnée d'un sens). L'information en tant que forme signifiante ne peut pas être définie en dehors de son contexte (elle est donc subjective).
- ✿ La **connaissance** est un ensemble de composites d'information, de comportement, de gestes, de mots, c'est un savoir théorique ou un savoir-faire.

Mener à bien les transactions de l'entreprise, avec son environnement s'appuie de plus en plus sur le préalable suivant : être capable de recueillir, d'analyser et de transmettre les informations pertinentes sur cet environnement. Cette activité de

collecte, d'analyse et de circulation de l'information prend aujourd'hui une telle importance qu'elle est devenue un problème majeur dans l'entreprise ou l'organisation.

L'encyclopédie "WikiPédia" définit une organisation comme :

"Un ensemble d'individus en interaction, regroupés au sein d'une structure régulée, ayant un système de communication pour faciliter la circulation de l'information, dans le but de répondre à des besoins et d'atteindre des objectifs déterminés".

D'après le sociologue français Alain Tourraine :

« Une grande organisation moderne est un type de production où l'énergie est de plus en plus remplacée par l'information, où la productivité dépend de plus en plus de la capacité à faire circuler des informations. L'activité professionnelle se définit de plus en plus comme la place occupée dans un réseau de communication ».

L'information est l'élément vital de toute organisation. Elle circule entre les éléments et les met en relation. L'information ne se contente pas de s'additionner à l'information déjà en stock, elle est facteur de changement. Le bon fonctionnement d'un système est assuré par la recherche permanente de la meilleure circulation. L'information permet de prendre des décisions et d'élaborer des stratégies dans l'organisation.

I.1.3 Passage de la culture verticale à la culture horizontale ou « l'entreprise aplatie »

La maîtrise industrielle passe aujourd'hui plus que jamais par la gestion de la qualité, la réduction des coûts et le respect des délais. Mais dans un contexte concurrentiel très exigeant, la satisfaction du client par rapport à un service, un produit ou un projet impose de repenser le savoir-faire de l'entreprise (Gutierrez, 2006).

L'exploitation des technologies de l'information et de la communication (TIC(s)) qui assurent un équilibre entre les composantes humaines (culture, valeurs, comportement), organisationnelles (processus, structures) et technologiques (système d'information et de communication) a conduit à repenser l'organisation de l'entreprise. Cette réorganisation consiste le plus souvent à faire transiter les processus et les structures d'une logique verticale centrée sur les fonctions à une logique transversale centrée sur les clients de l'entreprise. La satisfaction du client étant la condition majeure de la survie de l'entreprise (Gutierrez, 2006).

Selon Michael Hammer et James Champy :

« Le re-engineering est une remise en cause fondamentale et une redéfinition radicale des processus opérationnels pour obtenir des gains spectaculaires dans les performances critiques que constituent aujourd'hui les coûts, la qualité, le délai ».

Le ré-engineering est donc une action qui vise à reconfigurer le fonctionnement d'un processus et de l'organisation, les faisant passer d'un système orienté fonction à un système orienté client.

Un processus se définit comme un ensemble d'activités ainsi finalisées par un objectif global, un *output* matériel (produit) ou immatériel (service). Quant à la structure organisationnelle : c'est la manière dont sont organisés les ressources (les hommes, les informations), le temps (les véritables ressources) l'argent et les outils (les moyens). Fondamentalement la structure exprime la division du travail au sein d'une organisation.

Cette focalisation sur les processus a pour premier effet d'enlever leur raison d'être aux anciennes structures verticales et hiérarchiques, dites « *cloisonnées* » qui découpent l'entreprise selon ses métiers et ses fonctions : direction, division, département, bureaux, ...etc. C'est la reconfiguration des processus ou "Business Process Re-engineering" (BPR)¹ et des structures qui constituent donc le fondement de ce changement organisationnel.

Notons cependant, qu'une distinction est à faire entre le BPR et le CPI² (Saikali, 2001) dont l'objectif est d'essayer de toujours améliorer la qualité des processus métiers sans toucher à leur structure profonde. Le CPI permet d'apporter des correctifs et des redressements à certaines situations, mais il reste limité car les processus peuvent atteindre un seuil nécessitant leur remise en question complète.

Le BPR dépasse le cadre de l'informatisation des pratiques existantes, il nécessite une modélisation des processus d'entreprise ainsi qu'une spécialisation formelle des flux de travail qui les traversent. Ce constat se concrétise par le développement accéléré des concepts d'entreprise intégrée, d'ingénierie concurrente ou de gestion de la chaîne logistique globale. Aussi, on est de plus en plus amené à qualifier ces processus de « *socio-techniques* » car ils présentent une forte interaction entre les hommes et les outils d'information et de communication.

L'organisation par les processus implique une conception des structures en rupture avec les modèles classiques. Les formes structurelles favorisant la décentralisation de la prise de décision, la plus grande flexibilité, et la plus grande réactivité sont désormais privilégiées en entreprise. Le tableau ci-dessous est une comparaison des cultures « verticales » et « horizontales » en entreprise, d'après Sylvaine Brunet et Hervé Gardin tiré de (Saadoune, 1996).

¹ Business Process Reengineering ou Réingénierie des processus métiers.

² Continuous Process Improvement ou Amélioration continue des processus métiers.

Culture traditionnelle dite « culture verticale »	Nouvelle culture dite « culture horizontale »
Le processus de décision est dominé par les besoins des fonctions de l'entreprise.	Le processus de décision est dominé par les besoins des clients et des processus de l'entreprise.
Les collaborateurs de l'entreprise ne connaissent que les fonctions dans lesquelles ils travaillent.	Les collaborateurs de l'entreprise connaissent et comprennent le schéma global et ce que font les autres fonctions avec lesquelles ils ont besoin de coopérer.
Les collaborateurs de l'entreprise ne connaissent pas l'identité de leurs clients et les fournisseurs internes ainsi que les relations qu'ils ont avec eux.	Les collaborateurs de l'entreprise connaissent et comprennent les entrées et les sorties qui relient leurs fonctions aux autres fonctions.
Les fonctions ont des relations de concurrence	Les fonctions ont des relations de partenariat.
Les évaluations d'une fonction sont isolées des autres fonctions.	Les évaluations d'une fonction sont liées à la contribution qu'elle doit apporter à son client immédiat de l'entreprise.
Seuls les résultats sont mesurés.	Les résultats et les processus sont mesurés
L'information n'est pas régulièrement diffusée à toutes les fonctions.	L'information est régulièrement partagée par les fonctions concernées.
Les personnes ne sont pas impliquées dans les prises de décision ou leur implication est confinée à leur fonction.	Des groupes transversaux à tous les niveaux sont fréquemment réunis pour envisager les questions critiques des activités de l'entreprise.
Les personnes sont récompensées sur la base de leur contribution fonctionnelle.	Les personnes sont récompensées pour leur contribution à l'entreprise.

Table I.1 : Tableau représentant le passage de la "culture verticale" à la "culture horizontale" en entreprise.

I.2 La coopération

I.2.1 Définitions

Bien qu'il n'existe pas de définition unique et définitive de ce qu'est la coopération dans la communauté de recherche, nous essayerons d'abord de donner les définitions linguistiques du terme.

L'origine étymologique du terme coopérer est l'association de la racine « *operare* » et du suffixe « *co* » qui donne la signification de « *travailler ensemble* ». Cette notion de travail conjoint est retrouvée dans le dictionnaire LAROUSSE où coopérer est défini comme « *agir conjointement* ».

L'encyclopédie UNIVERSALIS définit la coopération comme :

« *Le fait, pour une personne, de s'adonner consciemment à une activité complémentaire à celles d'autres personnes dans le cadre d'une finalité commune, dans un groupe donné* ».

Dans (Dejours, 1993), on donne une définition commune qui considère la coopération comme :

« *Des liens que construisent entre eux des agents en vue de réaliser, volontairement, une œuvre commune* ».

Nous retrouvons donc trois traits caractérisant la coopération : le but commun, la forme consciente de la coopération et l'action collective.

▲ Le but commun

Pour qu'une coopération soit totale, il faut que le travail collectif corresponde à un but commun. Nous retrouvons ici la définition de Schmidt pour qui le travail coopératif (Schmidt, 1994) :

« Emerge quand plusieurs acteurs engagés dans la réalisation d'une tâche commune, sont mutuellement dépendants dans le travail et doivent se coordonner et rassembler leurs productions individuelles pour être en mesure d'effectuer la tâche qui leur a été confiée ».

Les acteurs coopèrent donc dans le but de réaliser une tâche commune. La coopération peut se restreindre à une collaboration. La Collaboration signifie uniquement travailler avec l'autre. Deux personnes peuvent collaborer, s'appuyer l'une sur l'autre et concourir pour l'accomplissement d'une tâche bien déterminée, sans pour autant partager le même but. Contrairement à la coopération, la collaboration ne prend pas en compte la solidarité dans les buts et les moyens.

▲ La forme consciente (ou le degré d'engagement)

Collaborer peut aussi signifier apporter son aide à l'autre sans pour autant chercher une récompense ou avoir une conscience que l'on est entrain de coopérer. Par ailleurs résoudre un problème en groupe implique l'engagement de tous les membres du groupe dans la réalisation de l'objectif, donc une conscience de coopération. Il faut qu'il y ait donc une volonté de coopérer, ce caractère de collectivité et de solidarité dans l'intention de coopérer nous permet de faire la distinction entre le terme anglo-saxon « *co-operate* » où des individus travaillent ensemble sans pour autant s'engager dans le groupe et « *coopérer* » où les acteurs (ou individus) travaillent ensemble dans le groupe.

▲ L'action collective (le degré de formalité ou de formalisation)

L'action collective est la composante essentielle faisant référence au travail du groupe. (Délfard, 2000) distingue la coopération formelle de la coopération informelle. Au niveau de l'individu la coopération est souvent informelle car elle est modulée par la confiance que l'individu porte envers les autres. Cette confiance peut revêtir trois formes :

- a. Elle peut être fonction des caractéristiques propres à la personne telle que l'appartenance à un groupe, à une famille ou à une ethnie.
- b. Elle peut également être liée aux précédents échanges, à la réputation de l'autre acteur (il s'agit là de la confiance relationnelle).
- c. Elle peut être liée à une structure formelle garantissant les activités de l'autre acteur (cette confiance est nommée institutionnelle).

Au niveau de l'entreprise la coopération est dite formelle car deux ou plusieurs entreprises partenaires, sous l'égide de contrats, mettent en commun des ressources et des moyens complémentaires pour la résolution de problèmes afin d'accomplir une ou plusieurs activités en commun.

I.2.2 Les formes de coopération

On peut alors se poser la question : "*Pourquoi les gens coopèrent-ils*" ? Les individus coopèrent à cause des capacités limitées de chaque individu. On considère essentiellement trois types de coopération : additive, intégrante et de débat.

▲ La coopération additive

Les capacités humaines individuelles sont limitées, sur un plan physique ou physiologique, mais aussi au plan psychologique. La "rationalité limitée" des processeurs d'information humains avait été mise en évidence, ce qui avait amené INBAR³ (1979) à cette conclusion désenchantée : "*Intellectuellement, l'homme apparaît comme un ordinateur particulier, caractérisé par une capacité de traitement de l'information lente, limitée et séquentielle*".

Cette capacité limitée peut être compensée par une "coopération additive" : en combinant capacités et efforts, un ensemble d'individus peut effectuer une tâche qui aurait été impossible à un individu seul. La division d'une tâche en sous-tâches précises et articulées dans le temps permet généralement de la réaliser avec un maximum d'efficacité. Paradoxalement, les innovations technologiques, en augmentant les capacités physiques et les possibilités de traitement de l'information de chacun, ont réduit l'impact de la coopération additive.

▲ La Coopération Intégrante

Avec la spécialisation croissante de chacun, il apparaît évident que chaque individu ne peut manier avec succès qu'un nombre limité d'outils. Il peut s'agir de deux types d'outils : les outils physiques ou matériels et les outils intellectuels qui sont plus difficiles à cerner car ce sont des compétences, des connaissances et des expertises

Cependant, dans l'un et l'autre cas, l'évolution actuelle s'effectue vers une spécialisation croissante et chacun se limite à un domaine d'activité. Cette différenciation du travail nécessite une activité coopérative d'intégration.

▲ La Coopération de Débat

La validité et la véracité d'une décision obtenue par un processus de travail basé sur la connaissance sont fragiles et contestables. L'objet de la coopération est de répondre à cette insuffisance. Le fait que le travail soit basé sur la connaissance entraîne l'existence de plusieurs stratégies possibles.

L'existence de ces différentes stratégies s'explique par des "points de vue" divers quant à la solution à apporter à un problème donné. La coopération de débat sert à sélectionner les bons ajustements, les bonnes solutions parmi l'infinie variation des points de vue. Même s'il existe des décisions erronées et des erreurs d'interprétation, des individus qui débattent de leurs problèmes avec leurs collègues de travail arrivent généralement à des décisions objectives et équilibrées dans des environnements complexes

³ INBAR

I.2.3 Mécanismes de la coopération

La coopération est une aptitude à la communication, la coordination et la collaboration d'un ensemble d'acteurs pour la réalisation d'un objectif commun.

- ▲ **Communication** : deux acteurs ne peuvent coopérer sans échanger des informations, soit directement ou indirectement par la modification de leur environnement commun.
- ▲ **Coordination** : la réalisation d'un objectif commun n'est possible que par une action coordonnée de l'ensemble des acteurs.
- ▲ **Collaboration** : c'est l'activité commune d'un ensemble d'acteurs, elle est l'expression d'une activité synchrone.

La coopération entre acteurs (individus) dans une entreprise ou dans un groupe résulte de leur capacité à collaborer, à communiquer, à coordonner leurs activités et à co-décider dans un souci de distribution des tâches.

Mais la « co-décision » mettant en jeu des mécanismes de négociation, peut être considérée comme une activité coopérative particulière.

I.3 Modalités de la coopération

La coopération s'organise à travers l'espace et le temps, la classification la plus connue, est celle de la matrice « Espace \ Temps » de Johansen (Khoshafian & al, 1998) ci-dessous :

TEMPS		Même (synchrone)	Différent (asynchrone)
		ESPACE	Même (local)
Différent (distant)	Interaction synchrone distribuée		Interaction asynchrone distribuée

Table I.2 : Matrice « Espace \ Temps » de Johanssen

➤ **Coopération locale ou à distance**

Les travailleurs qui coopèrent dans un même lieu (dans le même bureau, par exemple) sont en mesure d'interagir fréquemment, tandis que ceux qui coopèrent à distance sont limités dans leurs interactions par la disponibilité, la largeur de la bande passante et le temps de réponse du moyen de communication. Par exemple, pour un espace local, nous avons les réunions et à distance nous avons le téléphone ou la vidéoconférence.

➤ **Coopération synchrone ou asynchrone**

Les différentes tâches et sous-tâches d'un travail coopératif peuvent être effectuées simultanément ou reportées dans le temps. L'intervalle de temps entre deux tâches

coopératives varie. Les sous-tâches peuvent être réalisées comme une suite d'actions étroitement couplées ou comme une série d'actions interconnectées.

Par exemple : les réunions, les conversations téléphoniques ou encore la vidéoconférence sont des tâches synchrones. Alors que, la messagerie ou le fax, sont des tâches asynchrones.

➤ **Coopération collective et distribuée**

Dans le mode collectif du travail coopératif, les individus coopèrent ouvertement et consciemment : ils constituent un groupe qui a une responsabilité commune. Dans le mode distribué, au contraire, les individus sont semi-autonomes. Chacun peut modifier son comportement selon les circonstances et avoir sa propre stratégie : dans cette situation, chaque travailleur n'est pas nécessairement conscient des autres ni de leurs activités ; ils coopèrent au travers de leur espace de travail.

Les réunions présentent un exemple de coopération au mode collectif alors que les applications Workflow sont au mode distribué.

➤ **Coopération directe ou indirecte**

Dans le cadre de la coopération directe, les travailleurs interagissent en échangeant une information symbolique : ils communiquent.

Dans la coopération indirecte, les individus coopèrent via un appareillage technique, typiquement une machine. Par exemple, en fonction de l'état de fonctionnement d'une machine, un travailleur « A » va prendre une décision qui va modifier l'état de cette machine. En fonction de ce nouvel état, le travailleur « B » peut prendre une décision qui va entraîner une nouvelle modification. Dans ce cas, les travailleurs ne communiquent pas. Néanmoins, ils coopèrent.

Là encore, les réunions, le téléphone ou la vidéoconférence sont des exemples de coopération directe contrairement aux changements apportés à une BD qui est une coopération indirecte puisque l'accès y est en exclusif.

➤ **Les obstacles à la coopération**

Nous avons vu que la coopération est la participation intentionnelle et coordonnée des membres d'un groupe (de deux ou de plusieurs personnes) dans une action commune. Les obstacles habituels à la coopération sont : la compétition (rivalité entre des personnes ou des groupes), l'égoïsme, la non considération d'autrui, le manque de confiance, et les obstacles ordinaires de la communication (Saadoune, 1996).

De manière générale, au niveau du support de travail de groupe, les années 90 ont vu naître une nouvelle discipline de la Technologie de l'Information, la TCAO ou technologie de "Travail Coopératif Assisté par Ordinateur" (CSCW en anglais, pour "Computer Supported Cooperative Work") encore appelée "Collectique". La TCAO se propose de fournir des outils dédiés au support du travail de groupe. Quant aux outils permettant de supporter le travail coopératif, ils sont nommés *Groupware* ou *Collecticiels*.

I.4 Travail Coopératif Assisté par Ordinateur (TCAO)⁴

I.4.1 Le travail coopératif

On appelle « activités coopératives » tout ce qui entraîne des besoins de communication et de coordination entre personnes afin d'assurer conjointement une activité professionnelle ou sociale.

Le travail coopératif, le travail en équipe ("Teamwork") et la mise en réseau de compétences des collectifs de travail et des individus sont devenus des éléments clés de la société. Le management des entreprises, plus basé aujourd'hui sur la valorisation de la ressource humaine que par le passé souvent productiviste (Levan, 2000), évolue vers de nouveaux modèles moins hiérarchiques, valorisant la mise en réseau et la coopération.

En tant que domaine scientifique, le "Travail Coopératif" est encore immature et les problèmes théoriques sont nombreux. En voici quelques uns :

1. La définition de systèmes informatiques pour des environnements de travail coopératif est particulièrement difficile. On peut bien sûr décrire et spécifier un système d'information destiné à s'adapter à la nature sociale actuelle de l'organisation du travail ; mais réciproquement, ce changement technologique induira des modifications dans l'organisation sociale du travail. Les technologies de l'information ne peuvent être réalisées sans changements correspondants dans l'allocation des tâches. Il faudrait ainsi être en mesure de prédire le développement des processus de coopération et les modifications dans l'organisation qui résulteront du développement technologique.

2. Dans la recherche actuelle sur le TCAO, on présuppose habituellement que l'ensemble des personnes qui collaborent est un "groupe" ou une "équipe" ; i-e, un ensemble relativement fermé de personnes qui sont engagées dans un processus constant de communication et d'échange. Mais le concept de "groupe" ainsi défini est trop vague. L'interaction de différentes personnes d'une équipe est un processus complexe où chaque participant intervient avec des moyens et des procédures différenciées. La notion de groupe de travail ne couvre donc pas la riche complexité du "Travail Coopératif". Le groupe de personnes n'est pas l'unité spécifique de coopération des structures modernes. Dans ces dernières, la coopération utilise des moyens intermédiaires (machines, moyens de communication) et ne nécessite pas des contacts directs entre les personnes concernées. Dans les services administratifs, les agents coopèrent au travers d'espaces d'information virtuels, sans communication directe et sans nécessairement se connaître ni connaître les travaux effectués par chacun. Les systèmes informatiques destinés à supporter le travail coopératif doivent dès lors aider à utiliser les informations manipulées dans des contextes de travail différents par d'autres travailleurs, souvent inconnus. Le travail coopératif comporte donc des modes d'interaction directs aussi bien qu'indirects, distribués autant que collectifs.

3. La plupart des approches du Travail Coopératif sont orientées vers ou par les données, comme si l'information était quelque chose d'innocent et de neutre, d'où la nécessité, comme vu précédemment, d'adopter non pas une culture "*informatique de gestion de données*" mais plutôt une culture de "*informatique de gestion d'informations*". On pourrait ainsi penser qu'il est possible de concevoir un Système

⁴ En anglais CSCW : Computer Supported Collaborative Work.

d'Information permettant au manager de disposer d'un modèle unifié de son entreprise. Mais l'idée que les organisations sont des entités monolithiques est tout à fait naïve. Il faut plutôt considérer qu'une organisation est un mélange de collaborations et de conflits. La plupart des informations sont générées, traitées et communiquées, dans un contexte d'incohérence, de discorde et de conflits d'intérêts.

La notion de travail coopératif soulève donc une série impressionnante de questions et de sujets de recherche. Il s'agit en fait d'un thème multidisciplinaire impliquant entre autres : l'ergonomie, l'anthropologie, la sociologie des organisations, la sociologie scientifique, l'étude des processus industriels, la théorie du rangement, la théorie de la décision, la psychologie cognitive, les sciences cognitives, les théories de l'Intelligence Artificielle et l'étude des interactions Homme-Machine.

Cependant, la notion de travail coopératif est un concept très spécifique qui ne s'applique pas à tout système de production : il faut qu'il y ait production d'un produit particulier ou d'un ensemble de produits. En revanche, ce concept ne présuppose pas de forme particulière d'organisation :

- ▲ Il ne sous-entend pas l'existence de principes de travail, ni de travail de groupe, ni de communication face à face.
- ▲ Une organisation ou une corporation peut comprendre plusieurs processus de travail coopératif qui ne présentent ni lien, ni interférence.
- ▲ Le travail coopératif n'est pas limité à un métier mais peut franchir les limites d'un métier ou d'une fonction.

Dans une organisation peu complexe (Adam, 2000), le travail coopératif pourra impliquer plusieurs métiers qui collaborent en vue de l'exécution d'une tâche précise ou plusieurs sociétés qui servent à la mise en chantier ou à la production d'un produit élaboré en commun.

I.4.2 Les outils du TCAO : le Groupware et le Workflow

De manière générale, la littérature est très riche en classifications et taxonomies tendant à catégoriser les applications relatives au domaine du TCAO. Ainsi, certains ouvrages parlent indifféremment de TCAO ou Groupware comme le cas de (Levan, 2000) qui considère le groupware d'abord et avant tout comme un projet de changement au niveau du management avant d'être une simple introduction d'outils technologiques dans une organisation. Dans cette approche, le groupware est un projet comportant 1/3 de management, 1/3 d'organisation et enfin 1/3 d'informatique.

Cette approche se défend bien si l'on considère tout ce qui vient d'être dit sur les systèmes d'information coopératifs. Mais beaucoup de travaux relatent le groupware comme l'ensemble des outils dédiés au TCAO comme dans (Khoshafian & al, 1998) (Saikali, 2001) et autres. On parle aussi de "Collecticiels", i-e, l'ensemble des outils logiciels permettant le travail en groupe. Le terme "Collectique" revient souvent aussi pour désigner les applications de TCAO.

Pour notre cas, on se basera sur la classification assez courante des applications et outils relatifs au domaine du "Groupware" et qui peuvent se résumer dans la figure ci-dessous :

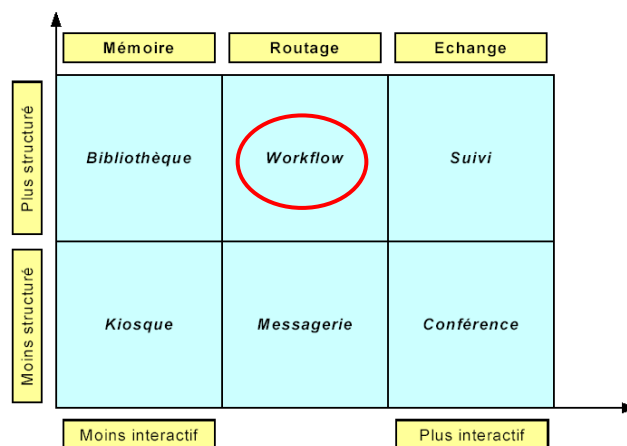


Figure I.2 : Les applications du groupware selon (Morand, 2001).

I.4.2.1 Définitions du groupware

Le groupware est l'ensemble des technologies et des méthodes de travail associées qui, par l'intermédiaire de la communication électronique, permettent le partage de l'information sur un support numérique à un groupe engagé dans un travail collaboratif et/ou coopératif (Courbon & al, 1999).

Le groupware est une technologie qui recouvre des domaines aussi vastes que la coopération, l'interaction homme-machine, et l'interaction interpersonnelle via les techniques numériques. Cette technologie a pour objectif principal de mettre en communication des personnes qui ont des intérêts ou des activités en commun sur la base d'informations qu'elles partagent. Autrement dit, le groupware a pour vocation de connecter et de partager. De ce fait, la plupart des produits de ce type sont construits sur des architectures client/serveur ou trois tiers.

Qui plus est, les applications de groupware utilisent et intègrent de nombreuses autres technologies. Dans cette perspective, le groupware peut être défini comme un système qui intègre du matériel et du logiciel existants en vue d'améliorer le travail en équipe, c'est-à-dire le travail coopératif.

I.4.2.2 Au cœur du Groupware : le Workflow

Comme on peut le remarquer dans la figure (I.2), il existe des liens étroits entre le Groupware et le Workflow puisque ce dernier est considéré comme un type d'applications groupware. En fait, ces deux technologies ont été développées dans le but d'améliorer le TCAO. L'introduction des logiciels de Groupware a permis de faciliter les interactions entre les groupes d'individus travaillant sur des sujets communs, en renforçant les aspects de communication, de coordination et de coopération, on parle des "3C" du groupware.

D'un autre côté, les logiciels de Workflow permettent d'automatiser les règles formelles (procédure établie) en vue de restructurer les procédures métiers de l'entreprise. On notera aussi que dans les technologies de Groupware, les logiciels de Workflow font partie des environnements les plus puissants pour automatiser les processus de travail. Ainsi, il serait légitime de définir le Workflow comme étant l'automatisation du travail coopératif visant un but commun et suivant une procédure établie, en ayant recours aux outils proposés par le Groupware.

En définitive, on peut dire que le Workflow est un cas particulier d'application de Groupware, et que la mise en place d'un environnement de Workflow dans une communauté de travail nécessite une bonne compréhension des concepts liés au Groupware, chose que nous allons aborder dans ce qui suivra.

I.5 Le Workflow

I.5.1 Origines du workflow

Même si cela ne fait pas l'unanimité, beaucoup considèrent que La Gestion Electronique des Documents (GED) est à l'origine du workflow. En effet, la GED est un ensemble d'outils et de méthodes développés dans le but de gérer et de partager des documents papiers de manière numérique. Cette gestion consiste à numériser les documents papiers afin de pouvoir les archiver sur des supports optiques après les avoir traités à l'aide de logiciels de reconnaissance optique des caractères (OCR). Les documents ainsi archivés sont indexés et accessibles aux utilisateurs à l'aide de leurs attributs ou de mots clés car dans un système de GED, un document est considéré comme un objet ayant des attributs et un contenu.

Ainsi, le Workflow est né avec la GED afin d'assurer le traitement automatique de ces documents et aussi leur routage vers leurs destinataires. La figure I.3 illustre les constituants de base d'un système de GED.

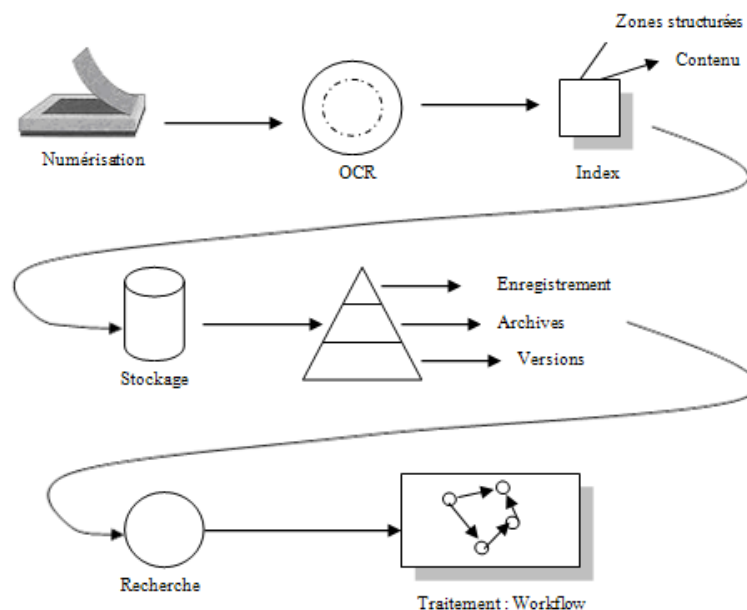


Figure I.3 : Constituants de base d'un système de GED (Khoshafian & al, 1998)

Cette première forme de workflow dits "papier" s'est progressivement développée vers d'autres formes plus à même de supporter les activités des individus impliqués dans une procédure métier donnée. On parle alors de workflow orienté activité ou processus où il ne s'agissait plus d'automatiser un routage de documents mais bien des procédures de travail impliquant des acteurs humains et des moyens ou ressources divers. Cette forme d'automatisation tire ses principes de la fameuse théorie de l'activité dont le célèbre psychologue russe "Vygotsky" en est une des figures de proue (Salomon, 1993). Dans ce travail, on s'intéressera à l'orientation processus des workflows.

I.5.2 Définitions

Dans la littérature Workflow, le sens du mot « Workflow » diffère selon qu'on l'utilise avec un « W » majuscule ou minuscule.

Définition 1 : Workflow (avec un « W » majuscule)

"Le Workflow est la technologie informatique du TCAO qui s'occupe de la gestion des processus des organisations".

D'après la définition donnée par la *Wfmc* (Workflow Management Coalition)⁵, le Workflow est l'automatisation de tout ou d'une partie d'un processus d'entreprise au cours duquel l'information circule d'une activité à l'autre, c'est-à-dire d'un participant (ou groupe de participants) à l'autre, pour action, en fonction d'un ensemble de règles de gestion. On parle aussi de « la gestion électronique des processus » (Levan, 2000).

(Saikali, 2001) en parle comme de la partie des technologies de l'information qui s'occupe de la coordination des acteurs des processus des entreprises, processus dont ils assurent également l'exécution et la gestion.

Définition 2 : workflow (avec un « w » minuscule)

Un workflow est un ensemble d'activités qui produit à terme un résultat tangible et observable (Conallen, 2000). Plus concrètement, un workflow est un processus d'une organisation, faisant intervenir plusieurs personnes qui coopèrent dans un but commun, en suivant une procédure établie.

Comme exemples de workflows, nous pouvons citer entre autres le traitement des prêts dans une banque, le traitement de sinistres dans une compagnie d'assurance, ...etc. Dans la littérature Workflow, on parle également de « processus Workflow » ou de « procédure Workflow », ou encore de « flux de travaux » pour désigner un workflow.

Enfin, dans un workflow on parle des "3R" relatifs aux rôles des acteurs, aux règles de fonctionnement du workflow et aux différents routages ou cheminement possibles dans un workflow.

I.5.3 Workflow et processus d'une organisation

I.5.3.1 Définition d'un processus

Un processus peut être défini comme étant un ordonnancement d'activités à travers le temps et les lieux, ayant un début et une fin, avec des entrées et des sorties pleinement définies (Saikali, 2001).

Suite à cette définition, on voit donc que toute organisation, quels que soient sa vocation et ses objectifs, est appelée à gérer des processus spécifiques à ses fonctions et à ses compétences. Ainsi, il est très important d'identifier et de classer clairement les processus censés être mis en œuvre par cette organisation afin de pouvoir les gérer facilement et développer des mécanismes pour l'optimisation de leur déroulement.

De manière générale, le cycle de vie de tout processus métier se résume en trois phases (Hollingsworth, 2004) essentielles comme le montre la figure suivante :

⁵ C'est un consortium d'industriels et d'organismes dont l'objectif est d'établir des standards permettant l'interopérabilité des systèmes Workflows.



Figure I.4 : Cycle de vie d'un processus métier

La première phase concerne la conception de la définition de processus en termes de ses activités, ressources et l'ordre d'enchaînement ; puis la génération de modèle représentant la définition (dans cette phase, on peut faire recours aux outils de conception et de modélisation existants sur le; ensuite, la mise en œuvre ou en exécution du modèle produit; et après, l'analyse des performances, l'évaluation et l'optimisation de processus métier (voire même la création d'une nouvelle définition).

I.5.3.2 Modèle de processus

Un modèle d'un objet du monde réel est un modèle s'il aide un acteur (individu) à répondre aux questions posées sur l'objet modélisé. Un modèle d'un processus métier, quant à lui, est une abstraction ou une substitution des activités, de l'ordre d'enchaînement et des ressources impliquées dans le processus.

La modélisation produit un modèle sur lequel on peut mener une analyse et une évaluation des performances qui aide à bien étudier et comprendre la chaîne de production des processus et par la suite identifier les problèmes.

(GREENWOOD & al, 1996) proposent de distinguer trois types de modèles de processus :

- ▲ **Modèle passif** : c'est un modèle statique qui représente l'état (structure et composants) d'un processus à un moment donné. Il n'est pas affecté si le processus a eu une modification ;
- ▲ **Modèle dynamique passif** : permet de modéliser et de représenter le fonctionnement et le comportement d'un processus métier. Il permet donc de servir de support à l'étude, l'analyse et la validation et l'optimisation ;
- ▲ **Modèle actif** : c'est un modèle qui représente l'état du processus de façon synchrone, c'est-à-dire, toute modification ou affectation de la définition du processus est capturée par le modèle. Il met en œuvre deux liens "réactif" et "actif" permettant respectivement de capturer le comportement du processus et de rester lié au processus en évoluant avec ses changements.

I.5.3.3 Taxonomie des processus

Plusieurs références dans ce domaine d'étude distinguent trois principaux types de processus qui entrent en jeu dans la vie d'une entreprise ; ce sont : les processus matériels (*material processes*), les processus informationnels (*information processes*) et les processus métiers (*business processes*) (Saikali, 2001).

➤ **Les processus matériels (*material processes*)**

Ce type de processus est basé sur des objets à caractère physique où les entrées sont des objets palpables et concrets. De plus, les activités constitutives d'un processus matériel peuvent être humaines ou automatisées, et se situent dans le monde physique. La mise en œuvre ou l'exécution d'un tel processus produit un résultat ou une sortie qui est aussi un objet physique.

➤ **Les processus informationnels (*information processes*)**

Contrairement aux processus matériels, dans les processus informationnels, le principal objet manipulé, en entrée comme en sortie, est l'information. Dans ce type de processus, les activités sont automatisées ou semi-automatisées (c'est-à-dire réalisées par des hommes collaborant avec des programmes). L'infrastructure de base pour les processus informationnels est fournie par le système d'information de l'entreprise, comme le SGBD par exemple.

➤ **Les processus métiers (*business processes*)**

Les processus métiers, aussi appelés "processus d'entreprise", ou "processus stratégiques", ou encore "processus opérationnels", sont les processus les plus importants de l'entreprise. En effet, ils sont directement liés au métier et aux objectifs principaux de l'entreprise qui les met en œuvre.

La *WfMC* définit un processus d'entreprise comme un ensemble de plusieurs activités reliées les unes aux autres pour réaliser un objectif, dans un contexte généralement organisationnel qui définit des rôles et des relations (Levan, 2000). Notons aussi que dans l'expression "processus métier", le terme "métier" représente l'offre d'un ensemble de services de l'entreprise aux clients. Donc la réussite de ce type de processus influe fortement sur le succès de l'entreprise.

Les processus métiers consomment différents types d'entrées (matériels, finances,...) et produisent des résultats (des sorties) à orientation le plus souvent économique. En effet, le terme "économique" est à prendre au sens large du terme car les résultats attendus ne s'expriment pas toujours directement en bénéfices financiers réalisés par exemple, mais peuvent prendre d'autres formes comme augmenter la compétence du personnel par exemple.

De plus, un processus métier, au cours de son exécution, peut faire appel aux deux types de processus dont nous avons parlé précédemment.

I.5.3.4 Langages de modélisation des processus métier

Plusieurs sont les langages de modélisation des processus métiers, issus de l'effet de la concurrence entre organismes et groupe. On trouve des langages destinés à l'exécution de processus tels que XPDL, BPEL, d'autres sont dédiés à la description tel que BPMN (Lonjon, 2004).

- **BPMN** (Business Process Modeling Notation) : initiative de BPMI (Business Process Management Initiative); c'est un langage destiné à la description statique et la conception des processus, il fournit une notation graphique pour la modélisation des processus;
- **XPDL** (*XML Process Definition Language*): lancé par la WfMC, il est considéré comme une couche intermédiaire (couche logique d'abstraction) entre les deux langages BPMN et BPEL. Il permet de transporter un ensemble

d'informations (même très spécifiques) à partir d'un outil de modélisation sans qu'aucun attribut ne soit perdu.

- **BPEL** (Business Process Execution language) : c'est le standard des langages soutenu par OASIS (Organization for the Advancement of Structured Information Standards). Il est dédié à l'exécution de processus et n'est pas utilisé dans la phase de conception. Il est vu comme une progression du langage BPML (Business Process Modeling Language) développé par BPMI ;

Les langages de modélisation connaissent continuellement des progrès afin de mieux maîtriser tout le cycle de vie des processus métiers.

I.5.4 Systèmes de Gestion de Workflow (SGWF)

I.5.4.1 Définition

Un SGWF est un système informatique dédié à la gestion des processus. C'est un système qui définit, implémente et gère l'exécution d'un ou de plusieurs workflows à l'aide d'un environnement logiciel fonctionnant avec un ou plusieurs moteurs de Workflow et capable d'interpréter la définition d'un processus, de gérer la coordination des participants et d'appeler des applications externes.

La *WfMC* attribue au SGWF un ensemble d'appellations synonymes et pratiques telles que « le système Workflow », « le Workflow », « le système de gestion de flux », « l'automatisation de workflow », « logiciel de gestion des procédures », « produit Workflow », « logiciel de Workflow » ou encore « fluxgiel ».

I.5.4.2 Environnements d'un SGWF

Un SGWF se compose de trois environnements distincts qui expliquent globalement son principe de fonctionnement :

- ✱ Un environnement de modélisation permettant le paramétrage d'un modèle de processus et des ressources associées.
- ✱ Un environnement d'exécution permettant le pilotage des processus en production.
- ✱ Un environnement de communication permettant au SGWF de communiquer avec d'autres SGWF.

➤ **Environnement de modélisation**

Aussi appelé « Build-Time », cet environnement se compose :

- ✱ de la base de données de modélisation, dans laquelle sont stockés les objets définis avec le client de modélisation ;
- ✱ du client de modélisation qui fournit une interface graphique permettant de définir les modèles de processus et les ressources associées.

➤ **Environnement d'exécution**

Aussi appelé « Run-Time », l'environnement d'exécution comprend :

- ✱ la base de données d'exécution dans laquelle on trouve toutes les données sur les processus en cours et les tâches qu'ils comptent.

- ✿ le moteur de Workflow : c'est le cœur du système Workflow, assurant le pilotage et le suivi des processus ; autrement dit, il constitue un outil précieux pour formaliser et automatiser les processus ;
- ✿ le client d'exécution : c'est l'interface avec l'application ou avec l'utilisateur pour dialoguer avec le moteur de Workflow et recevoir les tâches à exécuter.

La figure I.5 illustre la mise en correspondance de l'environnement de modélisation avec celui d'exécution (Lucas, 2001).

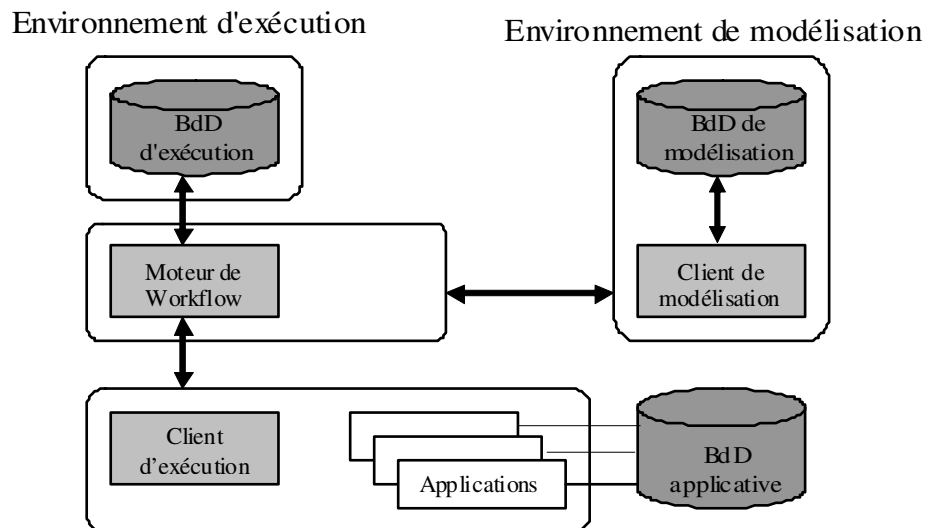


Figure I.5 : L'environnement d'exécution et de modélisation workflow (Lucas, 2001)

➤ Environnement de communication

Cet environnement comprend l'ensemble des outils et des fonctions d'API⁶, qui permettent au SGWF de s'ouvrir vers des applications extérieures, en particulier d'autres SGWF(s).

Les fonctions d'API permettent également aux utilisateurs de personnaliser le système et ses services en fonction de leurs besoins spécifiques.

I.5.4.3 Classification des SGWF

Dans le domaine du Workflow, on retient généralement deux principaux critères pour la classification des SGWF(s).

Il s'agit du domaine d'application du SGWF et des objectifs que le SGWF permet d'atteindre.

➤ **Classification par domaines d'application**

Cette classification met en évidence quatre types de systèmes Workflow : les systèmes Workflow administratifs (ou *Administrative Workflow Management Systems*), les systèmes Workflow de production ou transactionnels (ou *Production Workflow Management Systems*), les systèmes Workflow Ad-hoc (ou *Ad-hoc Workflow Management Systems*) et les systèmes Workflow collaboratifs (ou *collaborative Workflow Management Systems*).

⁶ Application Programming Interface.

A/ Les Systèmes Workflow administratifs

Les SGWF(s) dits administratifs sont orientés vers la gestion de processus administratifs simples, mais répétitifs ayant pour vocation de simplifier des circuits et de décharger les employés de tâches fastidieuses (Courbon & al, 1999). On évoque ici tout ce qui concerne le remboursement de frais, les comptes rendus d'activités, les demandes d'achats, les demandes de billets pour les voyages, les vacances, ...etc. Les systèmes Workflow administratifs prennent eux-mêmes en charge le routage des documents nécessaires et le remplissage de formulaires.

B/ Les Systèmes Workflow de production

Appelés aussi systèmes Workflow transactionnels, ou encore systèmes Workflow procéduraux, les SGWF dits de production gèrent des processus lourds, complexes et très répétitifs pour lesquels l'automatisation de nombreuses tâches administratives permet d'espérer une augmentation forte de la productivité administrative (Courbon & al, 1999).

Les termes « production » ou « transactionnel » signifient que l'entreprise a défini et mis en place des règles de fonctionnement et des procédures contraignantes. Les processus - généralement complexes - gérés par ce type de SGWF, sont au cœur même du métier de l'entreprise et représentent pour elle un niveau de risque élevé. Les tâches exécutées par les logiciels de Workflow dits de production sont fréquemment des transactions récurrentes qui impliquent la participation de plusieurs départements de l'entreprise et l'existence d'une structure créée pour les mettre en place et les contrôler.

Parmi les processus gérés par les systèmes Workflow de production, nous pouvons citer entre autres exemples la gestion des dossiers de prêts, d'assurances ou de réclamations.

C/ Les Systèmes Workflow Ad-hoc

Contrairement aux systèmes Workflow de production qui gèrent des tâches répétitives en fonction des circuits et des règles plus ou moins définis à l'avance, les systèmes Workflow dits "Ad-hoc" sont soumis à des objectifs dont les étapes et les niveaux d'interaction entre les intervenants dans le circuit global du processus sont plus difficiles à définir en détail et à prévoir (Khoshafian & al, 1998) (Levan, 2000).

Parmi les processus gérés par un SGWF Ad-hoc, nous pouvons citer entre autres exemples, l'embauche d'un candidat, un suivi de la clientèle, un service de dépannage, ...etc.

Le fait que l'exécution d'un workflow pour ce type de SGWF soit soumise à des délais et des objectifs implique que les responsabilités des intervenants dans le workflow peuvent changer dynamiquement. Le SGWF Ad-hoc fait souvent intervenir des utilisateurs plus créatifs et mieux formés que le SGWF de production.

D/ Les Systèmes Workflow collaboratifs

Les SGWF(s) dits collaboratifs permettent de réunir un certain nombre d'intervenants dans le but d'atteindre un objectif commun. Ils sont dédiés au support de travail de groupes tels que la conception ou la gestion de projet faisant appel à différents niveaux d'expertise (Saikali, 2001).

En effet, les SGWF(s) collaboratifs recouvrent la circulation de l'information entre membres d'un groupe dont le mode de communication privilégie le partage, c'est-à-dire, une certaine uniformité des rôles.

Néanmoins, certains rôles dans le groupe peuvent être différenciés (Courbon & al, 1999). Un exemple typique de cette situation est celui de l'édition conjointe d'un document entre divers contributeurs.

La segmentation de la classification des SGWF par domaines d'application s'appuie sur de nombreux critères convergents (Levan, 2000). Elle peut être représentée par la figure I.6 :

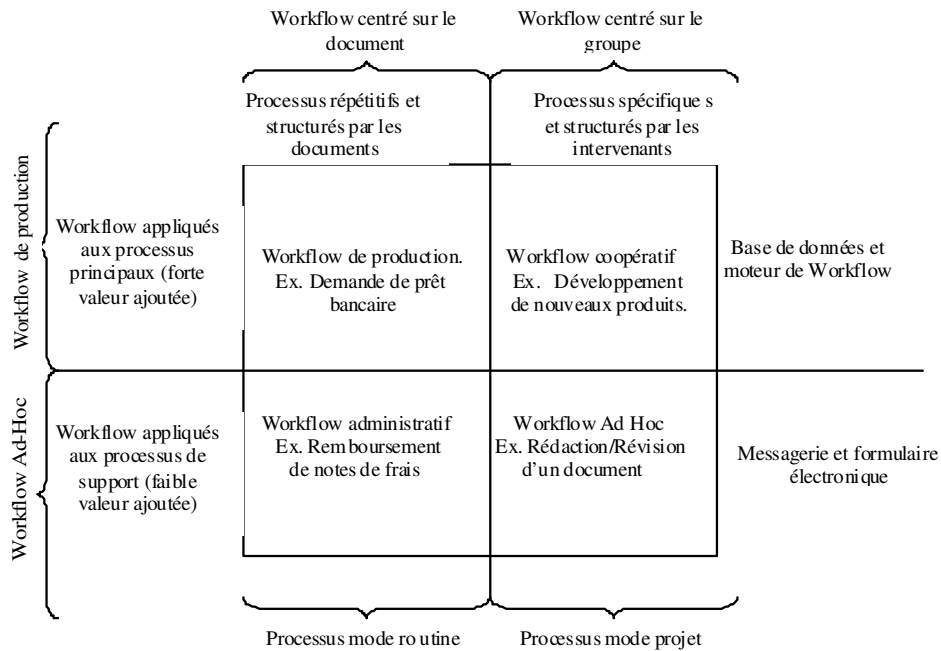


Figure I.6 : Classification de systèmes Workflow par domaines d'application (Levan, 2000).

➤ **Classification par objectifs**

La classification par objectifs met en évidence trois types de SGWF (Saikali, 2001) : les Systèmes Workflow orientés processus (ou *Process Workflow*), les Systèmes Workflow orientés documents (ou *Documents Workflow*) et les Systèmes Workflow de communication (ou *Communication Workflow*).

A/ Les Systèmes Workflow orientés processus

Les SGWF dits orientés processus sont dédiés à l'automatisation de l'exécution et la gestion des processus métiers des entreprises. Pour cela, les SGWF orientés processus basent l'automatisation des processus métiers sur une modélisation précise de ces derniers.

B/ Les Systèmes Workflow orientés documents

Les Systèmes Workflow orientés documents sont directement issus de la GED (la Gestion Electronique des Documents) dont ils se démarquent par des fonctionnalités supplémentaires d'assemblage, de création et de routage de documents.

C/ Les Systèmes Workflow de communication

Les SGWF sont dits de communication si les étapes des processus qu'ils gèrent sont représentées non pas en termes d'activités, mais plutôt en termes de transactions entre un client et un fournisseur. A chaque étape du processus qualifiée de transaction, on identifie un client et un fournisseur qui peut prendre à son tour le rôle de client par rapport à un autre intervenant. Vu que l'objectif principal est la réussite de la transaction, ce type de SGWF offre les moyens de trouver le meilleur chemin pour que la transaction ait lieu de manière satisfaisante pour le client.

Dans le cadre de ce travail, notre intérêt se portera sur la catégorie des workflows orientés processus.

I.5.5 Modèle de référence des SGWF(s)

Vu l'importance de la technologie workflow et ses incidences sur divers domaines de l'industrie et du management, une pléthore d'outils ont inondés le marché dès la fin des années 80.

S'en est suivie une période de régression due à des problèmes de différents niveaux d'où la communauté workflow a senti le besoin de se mettre d'accord autour d'un ensemble de points afin d'assurer la longévité de cette nouvelle technologie.

Pour cela, différents organismes sont nés, leurs objectifs étant d'établir des standards du workflow. Parmi ces organismes, nous avons à titre d'exemple la *WfMC* (citée précédemment) et la *WARIA*⁷.

Ce sont en général des regroupements d'industriels et d'institutions publiques et/ou privées dont le rôle et le but est de synthétiser, coordonner et de définir des standards dans le domaine de développement des produits Workflow, et de normaliser les différents concepts rattachés au Workflow.

En définissant une terminologie commune et des interfaces de programmation pour un Workflow standard, la *WfMC* espère promouvoir l'utilisation du Workflow grâce à la mise en œuvre de standards d'interopérabilité et de connectivité entre les multiples produits de Workflow et l'adoption de normes communes pour le déploiement du Workflow dans les industries. De plus, cette coalition n'exclut pas les travaux des autres associations, ce qui donne un caractère plus général à ses travaux.

Nos travaux se sont particulièrement basés sur ceux de la *WfMC* à cause de leur richesse et de leur disponibilité.

Donc la *WfMC* est l'un des regroupements les plus connus pour ses efforts visant à proposer des modèles de référence sur les divers aspects du workflow, allant du méta-modèle de base que nous présenterons plus loin, au modèle de référence d'un SGWF que nous allons décrire ci-après.

I.5.5.1 Le modèle de référence de la WfMC pour les SGWF(s)

Dans les sections précédentes, nous avons présenté les trois environnements qui expliquent le fonctionnement global d'un SGWF (Système de Gestion de Workflow). Le modèle de référence des SGWF (figure I.7) mis en place par la *WfMC* illustre cinq composantes que doit comporter tout SGWF.

⁷ Workflow And Reengineering International Association.

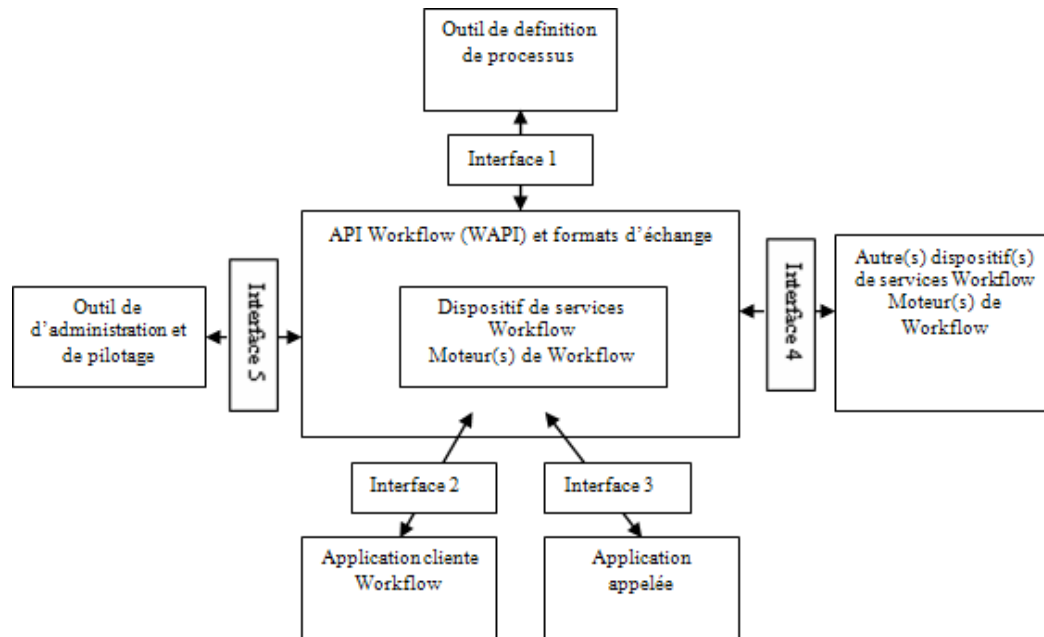


Figure I.7 : Modèle de référence du Workflow (WfMC)

La WfMC a mis sur pied un modèle de référence (appelé en anglais, *Reference Model for Workflow Management Systems*) sur lequel sont focalisés tous les efforts de chacun des groupes de travail constitutifs de la Coalition. Ce modèle est un schéma qui spécifie un cadre référentiel pour les SGWF(s) en définissant leurs caractéristiques communes, leurs interfaces et leurs fonctions. La Coalition l'a établi afin de définir les principales composantes de tout SGWF en partant d'une hypothèse toute simple : tous les SGWF reposent sur les mêmes composantes génériques qui interagissent selon diverses modalités (Levan, 2000).

I.5.5.2 Les composantes de base du modèle

Comme un SGWF est constitué d'un regroupement de plusieurs technologies, la WfMC a défini cinq composantes de tout SGWF, faisant elles-mêmes l'objet de standards. Ces composantes sont :

- ❖ les outils de définition de processus (ou *Process Definition Tools*).
- ❖ le moteur de Workflow (ou *Workflow Engine*).
- ❖ les outils d'administration et de pilotage (ou *Administration and Monitoring Tools*).
- ❖ l'application cliente Workflow (ou *Client Application*).
- ❖ l'application appelée par le Workflow (ou *Invoked Application*).

A/ Les outils de définition de processus

Ce sont des outils qui permettent d'analyser, de modéliser et de décrire les processus d'une entreprise. Ces outils permettent de spécifier dans une notation abstraite la logique de fonctionnement des processus.

B/ Le moteur de Workflow (ou serveur Workflow)

C'est le centre nerveux d'un SGWF. Il correspond à un environnement « Run-time » capable d'exécuter un ou plusieurs workflows. Cet environnement peut

impliquer un ou plusieurs moteurs de Workflow, c'est-à-dire des produits Workflow différents.

C/ Les outils d'administration et de pilotage

Ce sont des outils utilisés pour suivre l'exécution et le contrôle des processus.

D/ L'application cliente Workflow

Aussi appelée « client Workflow », l'application cliente Workflow est le module logiciel qui présente les bons de travail à l'utilisateur et peut appeler les applications et les outils nécessaires à l'accomplissement des tâches du même utilisateur. L'utilisateur rend ensuite la main au moteur de Workflow pour poursuivre l'exécution du processus. Le client Workflow peut faire partie intégrante d'un SGWF, comme il peut être un produit tiers (par exemple une messagerie) ou bien encore une application spécifique.

E/ L'application appelée par le Workflow

C'est une application externe appelée par le moteur de Workflow pour compléter le déroulement des activités du processus : on peut citer entre autres l'appel de service de messagerie, l'envoi d'une télécopie, l'utilisation des fonctions de gestion de documents, des outils bureautiques, des applications de production, ...etc.

Parmi ces cinq composantes, une revêt un rôle capital dans le fonctionnement du SGWF, il s'agit du moteur de Workflow (MWF). Ce dernier est le centre névralgique de tout système dédié au final à orchestrer des flux. Autrement dit, c'est le noyau central de tout système Workflow et qui fournit un environnement d'exécution des processus.

I.5.5.3 Description d'un moteur de Workflow (MWF)

Le MWF est un moteur logiciel qui fournit le run-time, c'est-à-dire, l'environnement d'exécution des instances de processus (WfMC, 1996). Le MWF est le module du SGWF qui permet d'automatiser l'enchaînement des tâches au sein du processus et d'en superviser le déroulement.

➤ **Caractéristiques du MWF**

Modularité : Un MWF doit être aussi modulaire que possible. Il doit permettre d'intégrer de nouveaux workflows aussi simplement que possible.

Intégration : Un MWF doit s'intégrer (ou s'interfacer) avec un serveur Web (en mode connecté) ou un SGBD.

➤ **Le moteur de Workflow vu comme un middleware**

(Lucas, 2001) regroupe dans la catégorie « middleware », les moteurs de Workflow. En effet, les cinq interfaces du modèle de référence des SGWF permettent au moteur de Workflow de communiquer avec les autres composants du SGWF et avec les autres SGWF(s). A ce stade, le moteur de Workflow est un véritable middleware.

Le Workflow constitue un moyen d'intégrer, au niveau des processus métiers, des applications hétérogènes avec un couplage lâche et opérant éventuellement dans des environnements différents.

Qui plus est, en revenant à la définition d'un SGWF, on peut dire que techniquement, un SGWF est un assemblage de composants secondaires qui sont intégrés au composant principal qu'est le moteur de Workflow au travers des interfaces.

Dans cette perspective, le moteur de Workflow occupe donc une place centrale et névralgique dans l'anatomie et le fonctionnement des SGWF. La figure I.7 illustre donc bien la position du MWF dans le modèle de référence des SGWF(s). En effet, le MWF est le composant principal de tout système Workflow, qui doit fournir toutes les fonctions permettant l'exécution d'instances de processus qu'il crée et gère lui-même.

➤ **Fonctionnalités d'un MWF**

Le moteur de Workflow permet l'exécution d'instances de processus qui sont basées sur des définitions de processus. Parmi les fonctions du moteur de Workflow, nous pouvons citer entre autres (Levan, 2000) :

- ✱ **L'interprétation des modèles ou définitions de processus** : Cette fonction consiste à réaliser l'exécution proprement dite des instances de processus générées à partir des modèles ou définitions de processus.
- ✱ **La création d'instances de processus et leur gestion du début à la fin** : A partir des modèles de workflow stockés dans la base de données de modélisation, le MWF crée et lance des instances de processus qu'il gère du début à la fin de leur déroulement. Les instances de processus seront créées conformément au modèle décrivant ce processus.
- ✱ **La navigation entre les activités et la création des bons de travail appropriés** : Le MWF doit assurer le routage des documents, des informations ou des tâches. Après la terminaison de chaque tâche, le MWF doit préciser la tâche suivante et l'utilisateur auquel elle est affectée.
- ✱ **La supervision et le contrôle général du workflow** : cette fonction consiste à suivre le déroulement des processus workflow. Pour chaque instance de processus créée, le MWF doit préciser la date et l'heure de création, de sa terminaison, l'état de déroulement des différentes tâches rattachées au processus, etc. La supervision de processus, aussi appelée suivi de workflow ou pilotage de processus sert à mesurer les performances (statistiques, etc.) du processus, ou tout simplement vérifier le bon déroulement des activités constitutives dudit processus.
- ✱ **La gestion des utilisateurs, des groupes et des rôles** : un "MWF" doit préciser à chaque instant, à quel groupe un utilisateur est affecté et le rôle qui lui est confié. Pour ce, le MWF doit généralement s'interfacer avec un SGBD.

➤ **Fonctionnement d'un MWF**

La mise au point d'un MWF permet de formaliser la définition de processus et d'automatiser l'enchaînement des tâches au sein du processus et d'en superviser le déroulement.

Dans sa fonction d'automatisation de l'enchaînement des tâches, le MWF détermine, dès qu'une tâche est terminée, quelles sont les tâches suivantes et à qui les attribuer en se référant à un modèle défini de manière préalable et formalisant le scénario du processus. Puis, il distribue immédiatement la tâche à l'utilisateur et/ou l'application sélectionnée.

Effectivement, le MWF affecte des tâches aux participants en fonction de leurs rôles et de la priorité des activités dans les instances de processus. Ces tâches sont notifiées aux participants via une application cliente Workflow qui fournit un environnement permettant de les accomplir.

Le MWF connaît aussi l'état d'avancement des différents processus qu'il contrôle. A tout moment, on peut l'interroger pour connaître l'état d'avancement d'un processus en cours.

Dans sa logique de fonctionnement, le MWF doit s'interfacer avec un SGBD afin de pouvoir gérer les définitions de processus et les différents utilisateurs impliqués au cours du déroulement de ces processus. Cependant, il convient de signaler qu'un MWF peut être conçu et réalisé pour fonctionner tant en environnement connecté qu'en environnement déconnecté.

➤ **Communication du MWF avec son environnement**

Le MWF se trouvant au centre de la plate-forme d'intégration, il se doit d'être extrêmement interopérable. La WfMC a bien compris cette nécessité. Rappelons que la WfMC est à l'origine d'une spécification concernant les systèmes de Workflow. Cette spécification se découpe en cinq interfaces qui identifient bien les différentes interactions du MWF avec son environnement comme nous le verrons dans la section suivante. L'environnement peut être interne ou externe au SGWF dont le MWF fait partie.

I.5.5.4 Les interfaces du modèle de référence

Afin d'obtenir l'interopérabilité entre plusieurs produits Workflow, il faut définir des standards d'interfaces et d'échange de données. Les interfaces définies dans le modèle de référence de la figure I.6 sont centrées sur l'utilisation des mécanismes de transport standard et permettent au moteur de Workflow – le centre nerveux du SGWF – d'inter-opérer avec les autres composants du SGWF et avec les autres SGWF(s). Plus concrètement, une interface est un mode de communication entre composants d'un même système ou entre deux ou plusieurs systèmes différents. Le modèle définit les cinq types d'interfaces suivants :

Interface 1 : Interface des outils de définition des processus (*Process Definition Tools Interface*)

Désignée aussi sous le nom d'« interface d'import/export de définition de processus », cette interface fournit un format commun pour l'échange des spécifications statiques entre l'outil de définition des processus et le moteur de Workflow. Parmi ces spécifications, nous pouvons citer :

- ❖ Les conditions de déclenchement et de terminaison de processus.
- ❖ L'identification des activités dans le processus incluant les applications externes associées et les données d'ordonnancement de processus.
- ❖ L'identification des types de données et des chemins d'accès.
- ❖ La définition des conditions de transition et des règles de routage.
- ❖ Les informations relatives aux décisions d'allocation de ressources.

Interface 2 : Interface d'application du client Workflow (*Workflow Client Application Interface*)

Cette interface supporte les interactions entre le client Workflow et le moteur de Workflow. Elle définit des standards appliqués au moteur de Workflow pour la gestion des bons de travail (*work items*) que le client Workflow présente à l'utilisateur.

Interface 3 : Interface d'application appelée (*Invoked Application Interface*)

Elle décrit comment différentes applications externes sont invoquées par le moteur de Workflow. L'interface d'application appelée permet au moteur de Workflow d'activer une application externe spécifique d'une activité donnée. Aussi faut-il ajouter que cette interface, intégrée au moteur de Workflow, ne demande pas d'action particulière de la part de l'utilisateur (par exemple l'appel à une messagerie électronique ou l'exportation des données vers une base de données).

Interface 4 : Interface d'interopérabilité entre Workflow (*Workflow Interoperability Interface*)

Cette interface décrit les interactions entre les moteurs de Workflow. Ces interactions incluent l'initiation, la demande d'information et de contrôle des processus Workflow et de leurs activités et les fonctions administratives.

L'interface d'interopérabilité entre Workflow définit des standards permettant à des SGWF(s) conçus et produits par différents éditeurs, de travailler ensemble sur les mêmes bons de travail. Ces standards d'interopérabilité peuvent agir à différents niveaux : du simple transit de tâche d'un produit workflow à l'autre jusqu'à l'échange intégral de définitions de processus avec des données d'ordonnancement.

Interface 5 : Interface entre outils de pilotage et d'administration (*Administration and Monitoring Tools Interface*)

Cette interface décrit les fonctions de pilotage et d'administration d'un moteur de Workflow. L'interface entre outils de pilotage et d'administration permet à un outil de pilotage et d'administration de travailler avec n'importe quel moteur de Workflow. Cela permet d'obtenir une vision complète de l'état d'un workflow cheminant à travers une organisation, indépendamment des systèmes Workflow mis en œuvre.

I.5.6 Architecture client/serveur d'un SGWF

En analysant le modèle de référence des SGWF(s), une architecture fonctionnelle de ceux-ci peut être facilement établie. Les architectures des produits de Workflow sont généralement différentes, mais cela n'empêche pas que certains modules fonctionnels apparaissent de manière constante. Dans ce qui suit, nous allons présenter une architecture de type client/serveur.

Cependant, il faut noter que tous les produits Workflow ne sont pas forcément client/serveur et que les composants présentés dans cette architecture ne se retrouvent pas dans tous les systèmes Workflow (Khoshafian & al, 1998) (Levan, 2000). L'architecture d'un SGWF présentée par la figure I.8 distingue deux parties essentielles : la partie client et la partie serveur.

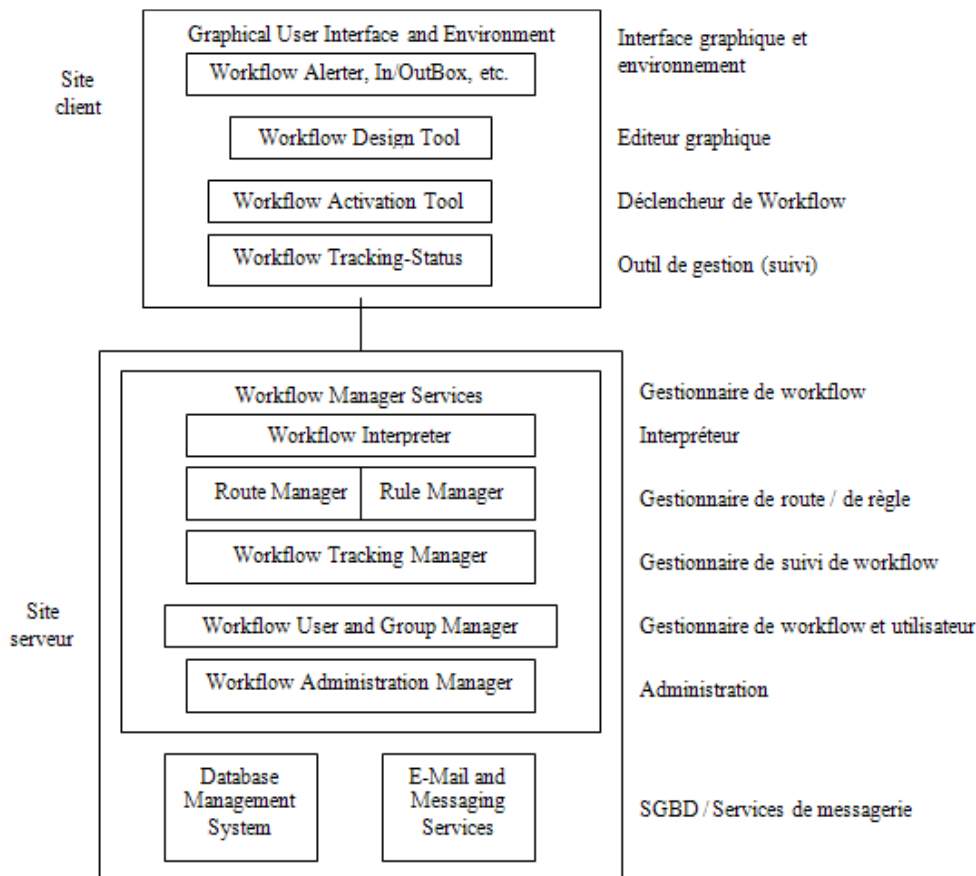


Figure I.8 : Les différents composants d'une architecture de Workflow

a) Partie client

Cette partie comprend trois composants principaux :

- ✱ **L'éditeur graphique (ou outil de définition des processus)** : ce composant permet la définition des modèles de workflow. C'est un outil de conception généralement graphique.
- ✱ **Le déclencheur du workflow** : ce composant est chargé de déclencher ou d'activer les fonctionnalités du workflow. Il gère les interactions qui se déroulent entre le moteur de Workflow et la partie client lors du démarrage, de l'arrêt ou de la suspension du workflow.
- ✱ **L'outil de gestion (suivi, statut, etc.)** : cet outil affiche l'état des différents workflows en cours, il indique la durée des tâches en cours et dialogue avec le moteur de Workflow pour connaître l'état du ou des workflows en cours (informations contenues dans la base de données).

b) Partie serveur

Cette partie comprend les composants suivants :

- ✱ **Les services de Workflow (ou le moteur de Workflow)** : il regroupe toutes les fonctionnalités propres au moteur de Workflow telles que l'interprétation des modèles de workflow, suivi du statut, gestion des utilisateurs, des groupes, des rôles et définition des workflows.
- ✱ **Les services des bases de données** : ce sont des modules avec lesquels les systèmes de Workflow dialoguent pour que le travail s'accomplisse efficacement. Généralement, un système de gestion des bases de données (SGBD) est associé au

moteur de Workflow. Grâce au SGBD, le moteur de Workflow peut définir, créer, rechercher ou mettre à jour des objets qui reflètent son activité. La base de données permet à plusieurs utilisateurs de lire ou de modifier les objets workflow de manière simultanée.

- ✱ **Les services de messagerie, de transport et de communication** : ce sont des outils (par exemple la messagerie) que les logiciels de Workflow utilisent pour échanger des messages, des données, des notifications, etc.

I.5.7 Conceptualisation du Workflow

Dans ce paragraphe, nous présentons les principaux concepts du workflow contenus dans le méta-modèle Workflow pour la définition de processus proposé par la *WfMC*, afin d'expliquer les concepts accompagnant la gestion de processus par un système Workflow. En effet, nous décrivons deux groupes de concepts : les concepts de base et les concepts complémentaires. La figure I.9 suivante représente le méta-modèle de la *WfMC* :

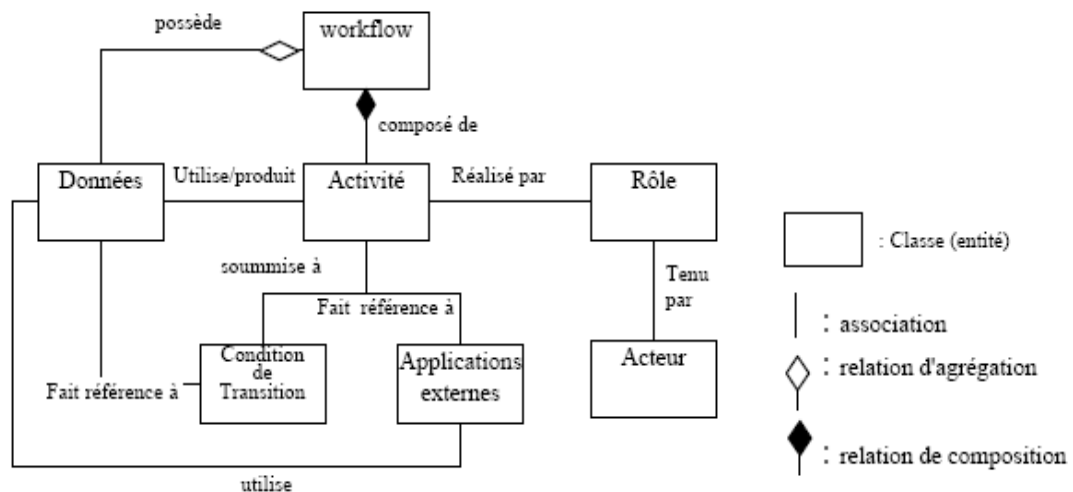


Figure I.9 : Méta-modèle Workflow pour la définition de processus de la WfMC

I.5.7.1 Les concepts de base

La figure précédente nous permet de décrire les concepts de base suivants :

- **Rôle** (*organizational role*) : c'est la liste des attributs, des compétences et du savoir-faire d'un acteur du workflow (participant) et qu'il met en pratique. Ce rôle définit la position de l'acteur dans une organisation.
- **Règle** : c'est le principe de comportement que l'acteur doit suivre pour exécuter une activité. En cours ou à la suite d'une opération sont mises en jeu des règles de gestion qui définissent la forme prise par l'information et le cheminement ultérieur vers les opérations suivantes.
- **Activité** (*process activity*) : d'après la *WfMC*, une activité est une étape d'un processus au cours de laquelle une action élémentaire (c'est-à-dire non décomposable) est exécutée. La *WfMC* distingue deux types d'activités : l'activité manuelle et l'activité workflow (ou automatisée). Dans la littérature Workflow, on parle aussi de « tâche », d'« action », d'« opération », de « nœud » ou d'« instruction » à la place d'« activité ».

- **Acteur** (*actor*) : aussi appelé « agent », « participant » ou « utilisateur », l'acteur est une entité (personne ou matériel), membre d'une organisation, chargée de réaliser les activités qui lui sont confiées, à travers ses rôles.
- **Donnée** (*relevant data*) : c'est l'objet qui est manipulé pendant la réalisation d'une activité ; la donnée peut être utilisée en entrée comme elle peut aussi être générée par la tâche.
- **Condition de transition** (*transition condition*) : c'est le critère qu'il est nécessaire de remplir soit afin de changer l'état d'une activité, soit pour passer à l'activité suivante ; elle s'exprime sous forme d'évènement ou d'expression logique.
- **Application externe** (*invoked application*) : c'est une application informatique que l'on appelle pour réaliser la tâche. Si l'application n'est pas uniquement informatique, on parle de ressource.

I.5.7.2 Les concepts complémentaires

Pour ce qui est des concepts complémentaires, nous pouvons soulever les points suivants :

- **Cas d'un Workflow** (*workflow case* ou *workflow instance*) : c'est l'instanciation d'un workflow, c'est-à-dire l'exécution d'un workflow sur une situation spécifique.
- **Bon de travail** (*work item*) : c'est la représentation du travail à effectuer par un acteur dans le cadre d'une instance d'activité dans une instance de processus.
- **Liste de tâches** (*worklist*) : appelée aussi corbeille, la liste des tâches est la liste des bons de travail qui attendent d'être réalisés par un acteur du workflow.

I.5.8 Flexibilité et Workflow ou Workflow avancé

Le problème de la flexibilité du Workflow ne recouvre pas un seul mais plusieurs domaines et aspects, que sont : les processus, les données et l'organisation. Par conséquent, les contraintes de flexibilité du Workflow ne s'appliquent pas à un seul domaine mais à la combinaison des trois. Il s'agit donc d'une part d'absorber tous les événements pouvant inquiéter la cohérence et l'intégrité des données mais également de conserver la cohérence du déroulement du processus en toute circonstance, ainsi que d'intégrer les modifications qui peuvent avoir lieu dans l'organisation de l'entreprise.

De manière générale, la flexibilité d'un système peut être définie par deux propriétés : celle de s'adapter à des situations ou des configurations nouvelles, et la capacité d'absorber les fautes ou les problèmes. Les domaines des bases de données et de la gestion de production sont de ceux où la recherche de la flexibilité est très développée (Saikali, 2001).

Dans la pratique, on constate cependant que la flexibilité du Workflow se reflète principalement au niveau des processus, bien qu'elle implique et s'appuie également sur les deux autres niveaux : données et organisation.

Ainsi (Weske & al, 1996) définissent la flexibilité du Workflow comme étant la capacité de pouvoir changer la définition d'un workflow en cours d'exécution ainsi que celle de réutiliser les modèles pour la conception d'autres workflows. Plus globalement, nous pouvons dire que la flexibilité du Workflow se définit par, et s'applique à deux niveaux complémentaires : la flexibilité du modèle Workflow et la flexibilité du système Workflow. On parle alors d'adaptabilité et d'adaptativité du workflow, i-e, l'adaptabilité concerne la flexibilité au niveau des instances de modèles workflow (cas

de workflow) et l'adaptativité est relative à la flexibilité du modèle workflow lui-même. La figure I.10 résume les différents cas de figure de la flexibilité :

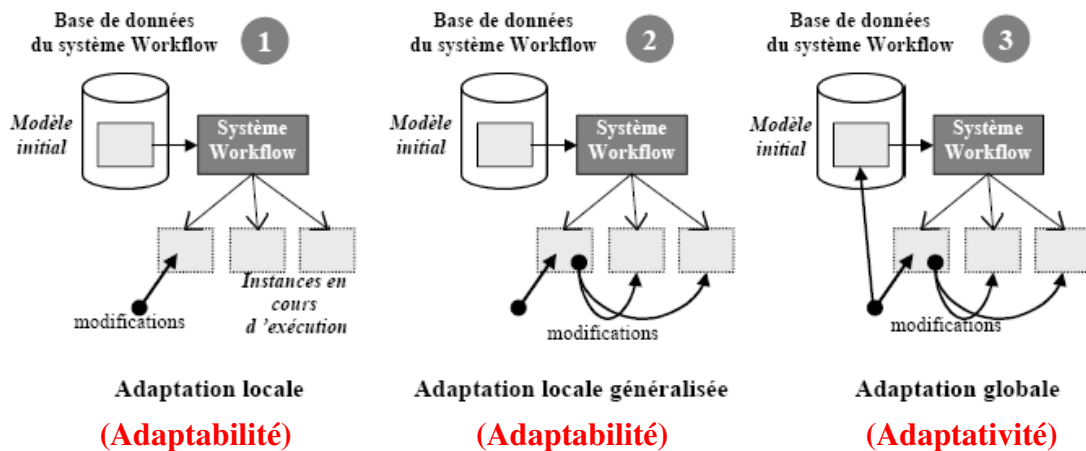


Figure I.10 : Les différents cas de flexibilité du workflow d'après (Saikali, 2001)

La flexibilité est donc l'une des principales caractéristiques de la "nouvelle génération" de systèmes Workflow. Ceux-ci doivent prendre en charge entre autre la mise en œuvre de la flexibilité.

Au niveau du Workflow, l'approche par méta-modèle a pour but de permettre d'apporter des modifications sur les modèles workflow. Plus généralement, une approche par méta-modèles tend à définir un méta-modèle Workflow qui sert de fondement à la construction et à la réutilisation de modèles Workflows. Ceux-ci sont donc construits à partir des composants du méta-modèle (ou de leur spécialisation si la réutilisation est possible) et en appliquant les règles de conception et de modélisation qui y sont définies.

I.5.9 Interopérabilité des SGWF(s)

I.5.9.1 Définition

L'interopérabilité des SGWF est la possibilité des MWF de communiquer afin d'exécuter de manière coordonnée, des instances de processus (WfMC, 1996).

C'est la possibilité de mettre simultanément en œuvre deux ou plusieurs MWF pour constituer un seul environnement de service Workflow capable d'exécuter, de façon partagée, des instances de processus. Autrement dit, deux ou plusieurs MWF peuvent coopérer pour partager l'exécution de processus workflow.

A cet effet, l'interface 4 du modèle de référence de la WfMC définit la spécification de l'interopérabilité entre des MWF hétérogènes.

I.5.9.2 Types d'interopérabilité de SGWF

Durant l'exécution d'instances de processus, quatre types d'interopérabilité sont à distinguer :

- ✱ **Interopérabilité hiérarchique** : quand on a un sous-processus à l'intérieur d'un processus général ;

- ✱ **Interopérabilité chaînée** : elle permet l'enchaînement de processus distincts mais complémentaires ;
- ✱ **Interopérabilité point à point** : elle permet le routage d'une activité à l'autre, chacune pouvant être exécutée par un MWF différent ;
- ✱ **Interopérabilité parallèle** : elle permet la synchronisation de workflows exécutés en parallèle sur des MWF différents.

Les besoins d'interopérabilité entre les systèmes workflows ont fait émerger une nouvelle technologie actuellement en pleine expansion, il s'agit du Web-Service dont nous allons décrire quelques notions dans ce qui suit.

I.5.10 Le Web-Service

L'émergence du concept de web service a suscité une évolution dans le monde de l'informatique distribuée et de l'interopérabilité entre systèmes hétérogènes de diverses plateformes. Ceci a été rendu possible grâce à l'envoi de messages XML facilitant la compréhension et l'échange de données, l'utilisation de standards (TCP/IP, SMTP, SOAP) acceptés par un grand nombre de fournisseur de logiciels (Microsoft, W3C, OMG, OASIS).

Basés sur le protocole HTTP qui utilise le port 80, les web services traversent les pare-feux de sécurité des entreprises, ce qui est intéressant dans les trafics B2B (Business-to-Business) ou B2C (Business-to-Customer). Dans le cadre de ce travail, les web services seront utilisés pour la composition ou la constitution du processus métier via le langage BPEL qui est le langage d'orchestration des web services et d'exécution des processus métier.

Ainsi, les besoins des entreprises changent suivant le développement technologique et la complexité de leurs systèmes informatiques. D'où l'émergence de technologies telles que CORBA (Common Object Request Broker Architecture), DCOM (Distributed Component Object Model), et bien d'autres, ont permis la conception et le déploiement d'applications sur des environnements distribués (intranet, internet). En dépit de ces technologies complexes et fortement couplées, il reste encore le problème de l'interopérabilité entre systèmes hétérogènes, issus de constructeurs différents. La tendance actuelle est donc vers l'architecture orientée services.

Au début, fut utilisée l'architecture client/serveur, basée sur les mécanismes RPC (Remote Procedure Call) permettant au client de récupérer ou de déclencher auprès d'un serveur et par l'envoi d'un message, un service donnée. Ensuite, on a vu le passage vers à un modèle d'architecture supportant l'exécution d'applications réparties sans passer par un middleware, c'est l'architecture orientée services. Celle-ci permet de faire continuer à fonctionner les applications déjà mise en œuvre (qui se base sur l'ancienne architecture) et d'avancer plus vers l'informatique distribuée en rendant possible l'interopérabilité entre systèmes divers, voir même la conception d'application répartis via les web services.

Le principe de l'architecture orientée service (AOS) consiste à offrir sur le web des services qui réalisent chacun une tâche bien précise, et des applications leur faisant appel si elles en ont besoin. Ces applications elles mêmes peuvent être déployées comme des services réalisant un grand rôle (Rampacek, 2006), (Salatgé, 2006).

Les web services constituent la pierre angulaire et la technologie de base de l'architecture orientée services. Apparus vers les années 1990, ils sont des portions

logicielles reliant des applications, des bases de données ou des processus d'affaires, en utilisant les protocoles et langage standards (HTTP, XML). L'interopérabilité et l'échange sont alors rendue possibles entre systèmes issus de concepteurs divers (Babin & al, 2003). L'apparition du XML codant tout type de données et favorisant le partage sur le Web de données entres différents systèmes hétérogènes, l'Internet et la recherche de l'interopérabilité sont autant de facteurs qui ont contribué à la propagation des web services.

I.5.10.1 Définition

On traduit un web-service comme une entité logicielle destinée à faire inter-opérer des systèmes hétérogènes dans un environnement distribuée (internet, intranet). D'après le consortium de la W3C : un web service est une entité logicielle identifiée par une URI (Uniform Ressource Identifier), dont l'interface est publique et les liaisons sont possibles grâce aux protocoles standards basés sur le fameux langage XML, qui permet l'interaction entre applications, en utilisant des messages XML et protocole de l'internet.

Un web service correspond donc à une application logicielle, à interaction dynamique, identifiée par un URI, pouvant interagir avec d'autres composantes logicielles et dont les interfaces et liaisons ont la capacité d'être localisées et invoquées via XML et par l'utilisation des protocoles Internet communs (Babin, 2003). Il s'agit alors d'une collection de fonctionnalités mises à disposition aux applications en temps réel, accessibles via le réseau par des messages écrits en XML en utilisant des protocoles standards (HTTP). Ces fonctions permettent d'instaurer des communications inter-application, voire même la construction rapide et moins chère d'applications distribuées plus flexibles et évolutives.

Contrairement aux plateformes déjà employées (CORBA, DCOM) qui sont fortement couplées, c'est-à-dire, la modification d'une composante logicielle entraine forcément la modification des autres, les web services présentent l'avantage d'être faiblement couplés et peuvent être modifiés indépendamment les un des autres. D'où l'utilisation des web services se voie nécessaire pour une meilleure interopérabilité des plateformes.

I.5.10.2 Le Modèle fonctionnel du Web-Service

Le modèle fonctionnel sert à fournir une vue descriptive du la façon dont les web services sont publiés, localisés et utilisés avec les divers protocoles impliqués.

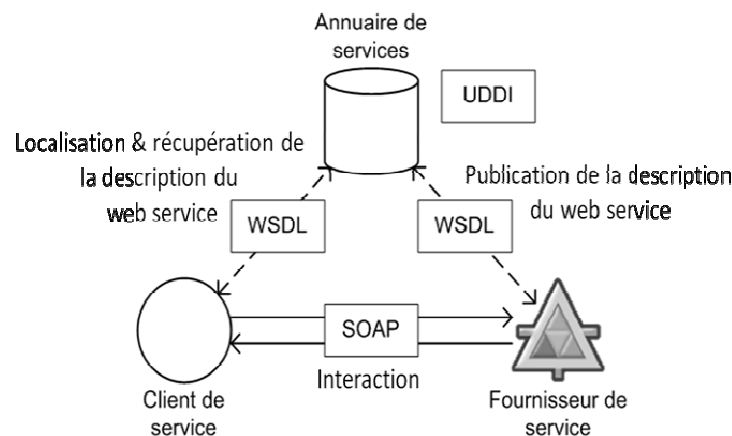


Figure I.11 : Modèle fonctionnel des web service

Ainsi illustré dans la figure I.11, un client ou une application peut interroger un web service auprès d'un fournisseur de web service, par l'envoi de messages écrits en SOAP. La communication se fait par étapes, la localisation et la consommation des web services.

La localisation ou la découverte consiste à l'émission d'un message, en utilisant le protocole WSDL, à l'annuaire UDDI, celui-ci contient des fichiers décrivant les web services à l'aide du langage WSDL, permettant de retourner toutes les informations (URI, paramètres) nécessaires à l'invocation et la consommation du web service (Rampacek, 2006), (Salatgé, 2006).

I.5.10.3 Les technologies des web services

Les web services font référence à une pléthore de technologies qui s'enrichissent au fil du temps, elles tendent à devenir des standards tels que XML, SOAP, WSDL, UDDI.

➤ XML (eXtensible Markup Language)

XML est depuis 1998, une recommandation du World Wide Web Consortium (W3C), il a été développé dans le but de rendre simple et utilisable sur le Web le SGML (Standard Generalized Markup Language) utilisable dans la gestion électronique de documents (Rampacek, 2006).

Le langage XML à balises extensibles est donc une amélioration du langage de balises HTML. Il permet de définir de nouvelles balises pour la description de données et leur texte de présentation. Contrairement au langage HTML, qui est un langage de présentation figé, XML est un métalangage définissant d'autres langages flexible selon les besoins des utilisateurs et extensible car permettant de décrire n'importe quel domaine de données.

Le descendant de SGML utilise deux fichiers textes, le premier sert pour la description de types de données que contient deuxième. Il est lisible par le non spécialiste et aucune connaissance n'est requise.

Ainsi, il est indépendant des plateformes et considéré comme un format adapté à l'échange de données et documents à travers les réseaux notamment le Web.

De ce fait, il est largement utilisé et plusieurs sont les formats et les standards issus de ce langage (MathML, CML, SMIL, XHTML, ...), chacun relatif à un domaine particulier.

➤ SOAP (Simple Object Access Protocol)

La communication avec les web services se fait via le protocole SOAP, qui est une technologie répandue pour la transmission et l'échange d'informations. SOAP a été proposé par Microsoft et IBM et fût adopté par le W3C.



Figure I.12 : Schéma d'un message SOAP

Dans SOAP, le moyen d'échange est le message. Un message SOAP est formellement spécifié comme un ensemble d'information XML fournissant une description abstraite de son contenu. Un message renferme trois composants (figure

I.12) : une enveloppe qui englobe le contenu et son traitement, un entête optionnel regroupant les règles et mécanismes de représentation des données et enfin un corps obligatoire contenant les requêtes et les réponses échangées entre systèmes (Mitra & al, 2007). Le tout est encapsulé dans un protocole applicatif avec son entête (version, type d'encodage). Le protocole applicatif le plus utilisé est HTTP mais bien évidemment il est possible d'utiliser d'autres tels que SMTP ou FTP (Salatgé, 2006).

➤ **WSDL (Web Services Description Language)**

C'est un langage de description des web services, initiative de différents constructeurs (Microsoft, IBM, ...) afin de fournir un langage de description unifié pour les web services et normalisé par le W3C. Le WSDL est une norme dérivée de XML, il fournit les spécifications nécessaires à l'utilisation des web services en décrivant les méthodes pouvant être invoquées, leurs signatures (paramètres), en précisant le format de messages et les URI des web services (Salatgé, 2006). La figure I.13 décrit la structure d'un fichier WSDL :

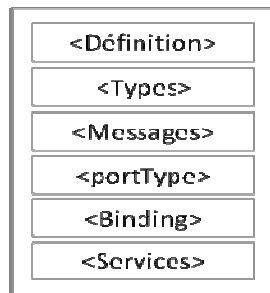


Figure I.13 : Structure générale d'un fichier WSDL

Le fichier WSDL décrit les web services indépendamment de leur implémentation, il renferme 6 éléments essentiels en plus d'autres optionnels (Booth & al, 2007) :

- ✱ **définition** : C'est la racine du fichier, elle précise le nom du web service ;
- ✱ **types** : indique tous les types de données utilisées entre le client et le fournisseur ;
- ✱ **message** : décrit un message unique, qu'il soit le message d'une requête ou d'une réponse, cet élément peut comporter l'élément **part** qui indique les paramètres d'un message ou ses valeurs retournés ;
- ✱ **portType** : combine plusieurs messages pour former une opération, celle-ci se réfère en entrée à un message et à plusieurs en sortie ;
- ✱ **binding** : regroupe des spécifications relatives à l'implémentation du service : protocole de communication, format de données pour les opérations, et messages définis par un type de port particulier ;
- ✱ **service** : renferme les adresses nécessaires à l'invocation d'un service donnée : ensemble de ports reliés ou URL invoquant le service SOAP ;

Ce langage de description ne possède pas une sémantique. Pour la supporter, une extension a été proposée par le W3C, c'est la SAWSDL (Semantic Annotations for WSDL and XML Schema) qui est accomplie en utilisant des modèles sémantiques telles que les ontologies. SAWSDL définit comment ajouter des annotations sémantiques dans différentes parties d'un document WSDL tels que les entrées et la sortie de structure des messages ou les opérations (Farrell & al, 2007).

➤ **UDDI (Universal Description, Discovery and Integration)**

Un annuaire offre des mécanismes afin de répertorier les web services, il détermine comment organiser et localiser les informations liées la description web services d'une entreprise, d'un organisme ou d'une seule personne afin qu'ils puissent être découverts par les divers clients et applications.

La version 3.0 du UDDI (Bellwood & al, 2002) est adoptée par OASIS (Organization for the Advancement of Structured Information Standards). Elle a été développée par un groupe d'industriels en l'occurrence Ariba, Microsoft & IBM. L'UDDI assure trois fonctionnalités : la publication des web services des fournisseurs en enregistrant leurs descriptions dans un fichier XML, la localisation et récupération de ces descriptions et la troisième fonction gère comment se connecter et utiliser le web service une fois celui-ci localisé.

UDDI fournit pour les environnements distribués, une infrastructure d'interopérabilité, il utilise "XML Schéma Language" et sa sémantique afin de décrire formellement les structures de données.

Il présente un modèle d'information composé d'instances de structures de données persistantes appelées entités, décrites en XML et stockées dans les nœuds d'UDDI. On distingue plusieurs types d'entités :

- ✱ **businessEntity** : sert à représenter les industriels ou organisation fournissant des web services;
- ✱ **businessService** : décrit une collection de web services fournies par un fournisseur représenté dans "businessEntity" ;
- ✱ **bindingTemplate** : concerne les informations techniques nécessaires à la consommation d'un web service particulier ;
- ✱ **tModel** : modèle technique renfermant les concepts réutilisables tels que le type de web service et le protocole utilisé par web service ;
- ✱ **publisherAssertion** : décrit une vue d'une "businessEntity" et ses relations avec d'autres "businessEntity" ;
- ✱ **subscription** : décrit les requêtes servant à localiser les changements des entités décrites par cette entité (subscription).

Etant donné l'objectif central des UDDI et qui est la représentation des données et métadonnées des web services et des interfaces regroupant les informations techniques relatives, le registre UDDI fournit un mécanisme standard pour classifier, répertorier et gérer les web services afin qu'ils puissent être découverts et consommés.

I.5.10.4 Composition de Web services

Par la technologie des web services, les entreprises peuvent offrir leurs clients une très large gamme de services sur Internet tel que la réservation d'un billet d'avion d'une compagnie aérienne ou la location d'une voiture, ou encore réserver une chambre dans un hôtel, faire l'achat et le paiement en ligne.

Dans certains cas, un web service peut assurer à lui seul un service donné, comme par exemple l'utilisation d'un dictionnaire en ligne pour la traduction immédiate d'un mot. Dans d'autres cas, le web service est forcé de se combiner et de se compléter avec d'autre web services afin de fournir un service donné : par exemple la réservation d'une chambre dans un hôtel qui nécessite un certain temps pour la négociation du prix et de son paiement. De là découle la notion de Web Service Composite dont nous donnons la définition ci-après.

➤ Définition

Un web service composite ou complexe est un service composé de plusieurs services et dont son exécution entraîne l'exécution des autres, en termes de faire appel à leurs fonctionnalités. Elle spécifie - afin d'accomplir une tâche qui nécessite la participation de plusieurs web service - la composition des web services invoqués et comment ils sont impliqués.

L'utilisation de plusieurs web services par un même web service permet de modéliser un processus métier. D'où la relation avec les workflows qui, nous le rappelons, permettent aussi de décrire des processus métier.

➤ **Mode de composition**

La composition des web services sert à déterminer la combinaison, la coordination et l'ordre d'exécution des web services, elle peut se faire de deux manières que sont : l'orchestration et la chorégraphie (Rampacek, 2006).

✿ **L'Orchestration**

Cette composition consiste en l'instauration d'un coordinateur de l'orchestration qui se charge de diriger le déroulement des différentes opérations des web services et interactions au niveau des messages en incluant la logique métier. Les web services n'ont pas connaissance qu'ils sont dans une même composition. La figure I.14 schématise une composition :

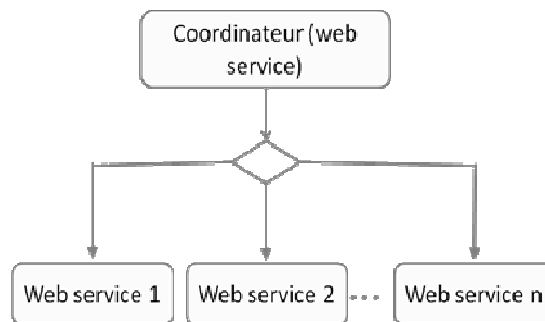


Figure I.14 : Orchestration des web services

Cette façon de faire la composition semble plus flexible, l'implémentation du coordinateur est facile car il régit des interactions connues entre les web services.

✿ **La Chorégraphie**

La chorégraphie n'implique pas un coordinateur qui gère les interactions. Les web services ont connaissance de leur appartenance à une composition, de leurs opérations qui doivent être exécutées et avec qui leurs interactions peuvent avoir lieu. La collaboration dans la chorégraphie des services web peut être représentée de la manière suivante (Figure I.15) :

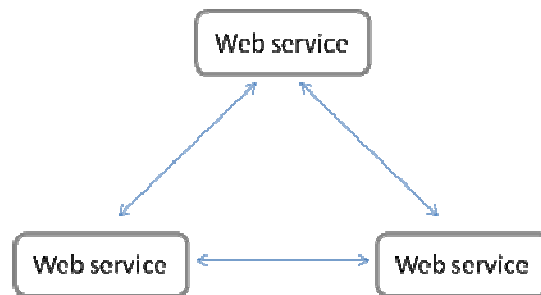


Figure I.15 : chorégraphie des web services

La chorégraphie présente l'avantage de la non-centralisation du traitement. Cependant, les web services sont sensés mener la coordination entre eux et être développés en tenant en compte de la composition qu'ils formeront (Rampacek, 2006).

➤ Les types de composition de Web Services

Trois types de compositions sont possibles suivant le degré d'automatisation (Barreiro, 2006) :

✿ Composition manuelle

Dans une composition manuelle, l'ensemble des web services est prédéterminé et l'ordre de déclenchement des web services est préfixé. L'utilisateur intervient dans la composition par un outil dédié. Ceci est très souhaitable car il permet aux utilisateurs de composer les services suivant leurs besoins, cependant cela requiert des connaissances dans la programmation.

✿ Composition semi-automatique

Parfois, les web services qui participent à une composition ne sont pas obligatoirement fixés à l'avance. Par exemple, la non-disponibilité d'un web service nécessite de le remplacer avec un autre assurant les mêmes fonctionnalités. La composition des web services est alors guidée par des suggestions que l'utilisateur peut tenir en compte dans la définition de son propre processus, en utilisant des outils graphiques pour la modélisation et la conception de web services composites.

✿ Composition automatique

Les web services d'une composition automatique ne sont connus qu'au fur et à mesure avec le temps y compris l'ordre de leur exécution. Le processus de composition est entièrement automatisé et n'est pas flexible car l'utilisateur ne participe pas dans son élaboration, c'est par exemple le cas d'un système workflow.

1. Langage de description des web services composites

Les langages de description ou de programmation des web services composite sont plusieurs, parmi les plus cités, nous avons :

- ✿ **WSFL** : Web Services Flow Language (Snell, 2001) a été développé par IBM dans le but de permettre la définition des processus métier. C'est une grammaire dérivée de XML permettant la description de la composition de web services. La définition de processus métier en WSFL revient à deux niveaux : dans le premier, on utilise une approche basée sur un modèle de graphe dirigé définissant un flux de travail. En deuxième niveau, une

interface publique permet la publication des processus métier en tant que web services.

- ✿ **BPML** : Le Business Process Modeling Language (Bpml, 2002) est un métalangage mis en place par BPMI (Business Process Management Initiative), il sépare les interfaces publiques (description) des interfaces privées (implémentation). Il permet la description des paramètres nécessaires à l'exécution par un moteur d'orchestration de processus métiers. Ces derniers sont modélisés et présentés par des flux de données des flux d'évènements. On peut exercer un contrôle sur ces flux en entraînant des règles métier, des règles de sécurité ou des règles de transactions.
- ✿ **XLANG** : A été développé en 2001 par Microsoft pour les besoins de l'outil de gestion de processus "BizTalk" et pour combler les lacunes du langage de description des web services WSDL en matière d'information concernant le comportement. Il ajoute quatre types d'opérations (requêtes/réponses, sollicitation de la réponse, le sens unique et la notification), et deux types d'actions à savoir : arrêt et exceptions. XLANG ressort de la première génération des langages de description des web services complexes (processus métier).
- ✿ **BPEL4WS** : Le BPEL For Web Service (Alves & al, 2007) est la fusion des langages précédents et présente donc l'évolution dans le temps. Il a été développé par un groupe d'industriels (Microsoft, IBM) pour un langage plus performant que les langages WSFL et XLANG, dont il tire profit.

I.5.11 Exemples d'outils dédiés au workflow

I.5.11.1 W4⁸ Suite (World Wide Web Workflow)

"W4 Suite" (en abrégé W4), est un système Workflow conçu dès l'origine par la société "W4 SA" pour l'environnement Internet. De ce fait, il intègre toutes les évolutions technologiques telles que XML, LDAP⁹, la sécurité, etc. Il dispose d'une architecture multi-tiers lui permettant de fonctionner en client/serveur Java et C/C++ et s'appuie sur une base de données garantissant l'intégrité des modèles et des instances de processus. Il est doté de son propre moteur de Workflow appelé le "W4 Engine" et d'une couche client distribuée représentée par des programmes clients.

Le système est multi plates-formes et peut fonctionner sous les environnements suivants :

- ✿ Microsoft Windows : NT, 2000 et 2003
- ✿ Unix : Solaris, AIX, Linux et HP-UX
- ✿ Java : Tomcat, Websphere, WebLogic, etc.

W4 s'appuie sur SQL Server et DB2 en environnement Microsoft et sur Oracle dans tous les environnements. Il permet l'activation de plusieurs dizaines de processus en même temps.

⁸ <http://www.w4.com>

⁹ Lightweight Directory Access Protocol, en français : Protocole d'accès aux annuaires légers.

W4 étant un système développé pour Internet, son utilisation se fait entièrement via un navigateur Web : toutes ses fonctionnalités sont ainsi disponibles à travers un navigateur. Dans ce qui suit, nous donnons quelques éléments essentiels de la composition et des fonctionnalités de W4.

1. Environnement de modélisation

L'outil de modélisation utilisé par W4 est appelé "W4 Studio V2". Il supporte la conception, la modélisation et la génération d'applications W4. Il permet une conception entière des processus de manière graphique. Il peut fonctionner en mode connecté au moteur de Workflow "W4 Engine" ou en mode déconnecté. On peut, à l'aide de cet outil, définir le graphe des processus, les différentes conditions de transitions, les rôles (et les acteurs dans la suite), ...etc.

Notons que les processus peuvent être composés (en nombre non limité) d'étapes, d'activités, de points de rendez-vous, de boucles, de sous-processus, de plusieurs points de terminaison, etc. W4 prend aussi en charge la notion d'acteur humain et d'acteur automatique car une étape du processus peut être exécutée par l'un ou l'autre des deux types d'acteurs. A l'aide de cet outil, il est possible de générer, au format HTML, une documentation du processus. La richesse fonctionnelle de la modélisation est un des points forts de W4.

2. Environnement de développement

L'environnement de développement intégré "W4 Studio" s'appuie sur une bibliothèque extensible de modèles d'écrans modifiables nommé "W4 Library". Il permet ainsi une très grande souplesse et une grande rapidité de programmation car ces modèles peuvent être modifiés à l'aide des éditeurs HTML standard.

Les formulaires et écrans d'application (corbeille, écran de recherche, etc.) sont générés à partir du processus, en suivant à partir d'un jeu de modèle l'écran sélectionné dans W4 Library. Seuls HTML et JavaScript sont nécessaires et donnent accès à toutes les fonctionnalités du moteur de Workflow "W4 Engine". Toutefois, W4 laisse le choix des technologies de construction des formulaires. On peut donc utiliser Java, JSP, ASP, Visual Basic, ... etc.

3. Exécution des instances de processus sur le serveur

Avec W4, il est possible d'exécuter en nombre non limité, les instances d'un processus quelconque, dont le nombre de tâches créées simultanément n'est pas limité. De plus, il permet de définir les rôles statiquement pendant l'étape de conception. On peut aussi déterminer les rôles et les acteurs de manière dynamique à l'exécution de l'instance de processus.

En ce qui concerne les tâches, elles sont créées par le composant d'ordonnancement "Scheduler" du serveur par l'interprétation du graphe associé au processus. Une tâche peut être refaite ou sautée pendant l'exécution du processus. Un acteur a le droit de suspendre une tâche qui lui est assignée ; il peut aussi la réaffecter à un autre acteur s'il en a les droits. En cas d'absence d'un acteur, ses tâches sont assignées à son remplaçant. La souplesse de la gestion des tâches est un autre point fort de W4.

4. Exécution des instances de processus pour le client

Rappelons que tout l'ensemble des fonctions utilisateur final, administrateur et coordinateur est accessible à travers un navigateur.

W4 offre la possibilité de personnaliser l'apparence, le contenu et l'organisation de la corbeille acteur par le client. Cela permet d'avoir toutes les informations nécessaires selon les critères jugés utiles par le concepteur. W4 est aussi capable de gérer les connexions/déconnexions des acteurs. Ainsi, ils peuvent se connecter et se déconnecter à tout moment. Les tâches non terminées par un acteur restent dans sa corbeille et lui sont présentées lors de sa prochaine connexion.

5. Administration, supervision et gestion

En ce qui concerne l'authentification, un module assure les mécanismes d'authentification unique soit par les interfaces Web de W4, soit en s'appuyant sur LDAP. Une représentation visuelle du suivi de chaque processus à l'aide de techniques permettant une variété de représentations personnalisées telles que : graphes, tableaux, schémas, etc.

De plus, il est maintenu un journal contenant l'historique complet de tous les processus et des tâches : temps d'activation, temps d'exécution, liste des participants, liste des tâches annulées, liste des tâches suspendues, etc. Enfin, toutes ces informations peuvent être exportées via des outils de bases de données.

I.5.11.2 Oracle¹⁰ Workflow

Oracle Workflow est un outil complet conçu pour la réalisation de l'automatisation des processus de gestion des entreprises. Il permet la modélisation et l'automatisation des processus quel que soit le niveau de leur complexité. Il assure une parfaite collaboration entre les différents rôles concernés et supporte le routage des fichiers pour approbation et/ou revue. Oracle Workflow intègre l'ensemble des composants suivants :

1. Oracle Workflow Builder

"Oracle Workflow Builder" est un outil graphique qui permet la modélisation, la visualisation et la modification de processus d'entreprise à l'aide de simples opérations de glisser/déplacer accessibles tout au long de la phase de développement. Il est utilisé pour développer des processus de services spécifiques à chaque organisation en spécifiant les activités, les types d'objets, les messages et les notifications qui leurs sont propres.

Chaque diagramme est la représentation graphique du processus de workflow, il contient obligatoirement un nœud de début pour marquer le lancement du processus et un nœud de fin pour marquer sa terminaison. Chaque nœud intermédiaire du diagramme représente une activité, elle peut être de type fonction, notification, sous-processus ou événement. Chaque trait reliant deux nœuds représente une transition. Une transition détermine la règle de passage d'une activité à l'autre. La figure I.16 présente un exemple de processus workflow :

¹⁰ <http://www.oracle.com>

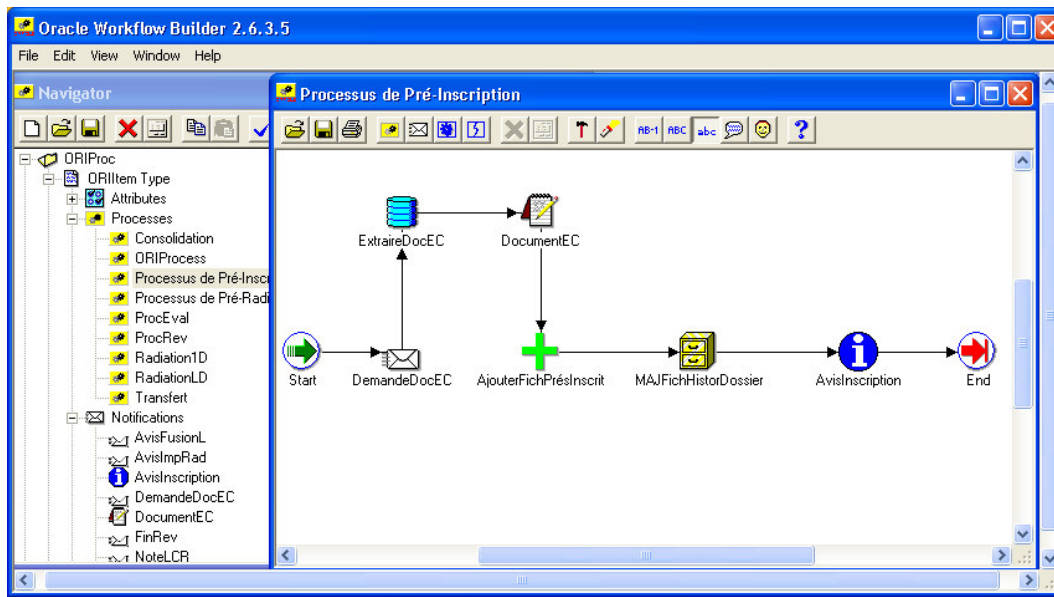


Figure I.16 : Modélisation d'un processus workflow avec Oracle Workflow Builder.

2. Workflow Engine

Le "Workflow Engine" est le noyau d'oracle workflow. C'est un package ou un ensemble de procédures écrites en PL/SQL. Ces procédures peuvent être appelées depuis n'importe quelle instance de processus de Workflow. Les principales fonctionnalités du "Workflow Engine" sont :

- ✿ Le contrôle du flux d'exécution du workflow : sachant qu'un processus workflow est une suite d'activités reliées les unes aux autres, le "Workflow Engine" contrôle le résultat de chaque activité pour déterminer la prochaine activité à exécuter.
- ✿ La gestion de l'exécution des branchements parallèles et des points de rendez-vous suivant les règles imposées par les transitions définies au moment de la modélisation.

3. Background ENGINE

C'est aussi un package écrit en PL/SQL. Il accède aux tables du workflow à la recherche d'activités en attente d'exécution. Par exemple pour les activités dont le temps est écoulé (« *Time Out* ») et dont l'état sera égal à « *Defferred* », le "Background Engine" complètera leur exécution et mettra leur état à « *complete* ».

Le "Background Engine" est configuré pour être exécuté (intervenir) à des intervalles de temps choisis au moment de l'installation, cependant il est possible de le configurer pour un item type spécifique selon le choix du développeur. Il peut exister plusieurs backgrounds Engine en exécution au même moment, la charge du travail sera alors répartie entre eux.

4. Système de gestion des Événements

Le système de gestion des événements est un service qui utilise l'infrastructure avancée des queues « Oracle Advanced Queuing (AQ) » pour garantir l'échange des événements entre systèmes. C'est un composant qui assure aux applications de Workflow la possibilité de :

- ✿ Définir une gestion des événements intéressante.
- ✿ Enregistrer d'éventuels abonnements à des événements choisis.
- ✿ Référencer voire appeler des événements à partir d'applications externes.
- ✿ Personnaliser les packages des applications sans avoir à modifier leur code.
- ✿ Gestion et routage des messages inter et extra applications.

5. Workflow Définitions Loader :

"Workflow Définition Loader" est un programme qui extrait la définition d'un workflow de la base de données et met à jour sa copie dans le fichier correspondant. Il peut être utilisé pour transporter les définitions du processus workflow de la base de données de développement à celle de production ou bien pour appliquer les mises à jour à un processus déjà défini. "Oracle definitions loader" est intégré à "Oracle Workflow Builder" pour permettre d'ouvrir et de sauvegarder la définition des processus dans la base de données et/ou dans un fichier.

6. Electronic Notifications

Oracle Workflow permet l'intégration des utilisateurs (Directory Service) pour exécuter les activités non automatisables. Le système de notification envoie et traite les réponses des utilisateurs workflow et une notification électronique est routée au rôle concerné. Tout utilisateur associé à ce rôle pourra visualiser cette notification. A chaque notification est rattaché un message qui contient toutes les informations nécessaires à l'accomplissement des tâches du rôle récepteur. Ces mêmes informations peuvent être incluses dans un document rattaché au message. Dans le cas de notification avec attente de réponse, Oracle workflow interprète la réponse pour décider du branchement vers la prochaine activité.

7. Electronic Mail Integration

Chaque acteur peut recevoir et répondre aux notifications via sa messagerie personnelle. La notification par e-mail peut inclure un document rattaché ce qui offre un autre moyen de réponse aux notifications.

8. Workflow Monitor

Les utilisateurs workflow peuvent visualiser la progression du travail d'une instance de processus workflow en se connectant au workflow Monitor, via un standard navigateur web qui supporte Java. Le workflow Monitor affiche des vues du diagramme d'une instance particulière du processus workflow et une description graphique de l'état de chaque objet : processus, activités, ...etc.

9. Internet utilisateur de workflow

Tout utilisateur qui accède à Internet peut être inclus au workflow. L'utilisateur du web peut accéder à la page de notification pour visualiser les résultats de son objet de travail, puis naviguer vers d'autres pages, pour voir d'autres détails.

10. Quelques éléments relatifs au processus workflow

➤ Notion de flux synchrone et asynchrone

Oracle workflow gère les processus de workflow suivant des règles préalablement spécifiées. Un processus workflow est composé d'activités et de relations reliant ces

activités entre elles. Une activité peut être une procédure PL/SQL enregistrée dans la base de données ou une fonction externe, une fonction de notification à un utilisateur ou à un rôle avec attente de réponse éventuelle, une fonction événement ou bien un sous-processus qui lui-même est un sous réseau d'activités.

Le concept de flux synchrone et asynchrone est essentiel à la modélisation de processus métiers, c'est pour quoi Oracle workflow offre la possibilité de le définir en utilisant des objets qui lui sont spécifiques.

- ✿ **Le flux synchrone** : Le flux synchrone est un flux non interrompu du début à la fin, le résultat de ce type de flux est récupéré dans des variables (item attributes) ou directement dans la base de données. Cette solution est effective si le temps d'exécution du processus workflow n'est pas long.
- ✿ **Le flux asynchrone** : Le flux asynchrone est un flux dont le résultat ne peut pas être récupéré immédiatement, les activités incluses dans ce flux sont en attente d'autres résultats initiés ou modifiés par un autre flux. Le "Workflow Engine" interromp le flux pour éviter l'attente infinie et met à jour les tables appropriées.

➤ Les composants du processus de workflow

Selon la nature du processus à créer, on définit quelques uns ou tous les composants suivants lors de la construction du processus :

- ✿ **Item Types (type d'objet)** : Pour chaque processus de workflow est associé un item type. Il représente une classification des composants qui construisent le processus de workflow. Pour chaque item type, oracle workflow offre la possibilité de lui rattacher des attributs. Ces attributs sont équivalents à des variables globales. Le processus workflow peut les référencer ou les mettre à jour en utilisant les API¹¹(s) du "Workflow Engine". Il existe dix types d'attributs (texte, nombre, date, valeur de retour, forme, URL, document, rôle, attribut, événement). Le type détermine la valeur de retour tolérée pour cet attribut et la façon dont il sera utilisé.
- ✿ **Type de retour** : C'est une liste statique de valeurs, elle peut être référencée par les attributs des activités, messages, item types pour spécifier une valeur de retour éventuelle.
- ✿ **Message** : Représente l'objet envoyé au rôle par une activité de notification dans un processus workflow. Il est possible de rattacher au message des attributs locaux propres au message ou des attributs qui référencent l'item type. Un message peut être de type attente de réponse spécifique du rôle récepteur du message, cette réponse est nécessaire pour déterminer la prochaine activité à exécuter.
- ✿ **Activité** : Est une unité composant le processus de workflow, elle peut être : une notification, un événement, un sous-processus ou une fonction. La fonction peut être locale et écrite en PL/SQL ou externe écrite en Java.

➤ Les API(s) du "Workflow Engine"

¹¹ Application Programmer Interface.

Oracle workflow met à disposition des développeurs un ensemble de fonctions et de procédures prédéfinies pour les aider à réaliser le workflow. Il est possible d'appeler ces API(s) à n'importe quel moment durant le développement ou bien au moment de l'exécution. Ces fonctions servent à communiquer avec le "Workflow Engine" ou pour changer le statut des activités. Il est possible de les appeler à partir de n'importe quel programme java comme méthodes externes. Par exemple, "Create process" permet de créer une nouvelle instance de processus alors que "Lanchprocess" est pour lancer l'exécution de l'instance du processus.

➤ **Les API(s) du "Workflow Function"**

Les API(s) fonctions sont des classes abstraites incluses dans toute application JAVA externe, elles contiennent les méthodes pour accéder aux attributs des activités ou de l'item type.

➤ **Les API(s) du "Workflow Attributes"**

Elles contiennent la description des items attributs : le nom interne, le nom public, le type de données, ...etc, en plus des méthodes d'accès à ces mêmes informations.

➤ **Les API(s) "Notification"**

Les API(s) notifications sont appelées par l'agent de notification. La méthode "send" envoie un message spécifique au rôle et retourne un ID (identifiant) notification si l'opération a réussi.

I.6 Conclusion

Ce chapitre nous a permis de passer en revue les notions et principes relatifs aux systèmes d'information coopératifs SIC(s). Nous nous sommes particulièrement intéressés au domaine du travail collaboratif assisté par ordinateur TCAO dont le workflow est l'un des outils.

Le modèle de référence des SGWF(s) établi par la WfMC a pour objectif d'unifier les efforts fournis pour la construction de système workflows conformes à certaines caractéristiques comme l'interopérabilité par exemple. Ce modèle est un schéma spécifiant un cadre référentiel pour les SGWF(s) en définissant leurs caractéristiques communes, leurs interfaces et leurs fonctions.

La mise au point d'un SGWF nécessite impérativement celle d'un moteur de Workflow, composant principal de tout SGWF, qui fournit le "run-time", c'est-à-dire, l'environnement d'exécution des instances de processus Workflow. Le moteur de Workflow se doit d'être interopérable pour pouvoir communiquer – au cas échéant - avec d'autres moteurs de Workflow si cela s'avère nécessaire.

Les notions relatives à la modélisation de processus ont aussi été passées en revue notamment les processus de nature à être modélisés par les workflows. Ont été aussi relatés les aspects concernant la flexibilité des modèles de workflows et des systèmes workflows. A cet effet, l'une des premières conceptualisations du workflow peut se retrouver dans le méta-modèle proposé par la Wfmc qui présente les principaux éléments de l'ontologie des processus. Ce modèle peut constituer un bon cadre de départ

pour tout développement d'un système workflow prenant en compte l'adaptabilité et l'adaptativité.

La technologie des web services constitue aussi un autre aspect de l'évolution la technologie des workflows notamment en matière d'interopérabilité, d'orchestration et de chorégraphie. Aujourd'hui, on parle plus de composition de web services où un web service peut représenter lui-même un workflow ou bien un workflow peut lui-même être constitué à partir d'une collection de web services. C'est dire toute l'extensibilité que la technologie web service peut apporter au domaine du workflow et vis versa.

Comme exemple d'outils dédiés au workflow, nous présentons "W4 Suite" et "Oracle Workflow", qui sont des systèmes de Workflow orientés processus. "W4 Suite" est un système dédié complètement au domaine du workflow alors que "Oracle Workflow" est un des outils intégrés dans la large suite des produits Oracle destinés au monde du business et couvrant des besoins aussi divers que les bases de données, la business intelligence, le développement, ...etc.

Cette étude nous a permis de constater que l'usage des technologies workflows peut être étendu à toute organisation exerçant des métiers non forcément orientés vers le monde du business, pourvu que les processus visés répondent aux règles et critères de processus workflow.

Dans le prochain chapitre, nous allons présenter les systèmes multi-agents et leur application au domaine de l'enseignement.

Chapitre II

**Les Systèmes Multi-Agents et leur
Application dans l'Enseignement à Distance**



Chapitre II

Le Systèmes Multi-Agents et leur Application au Domaine de l'Enseignement

Dans ce chapitre, nous présentons le second contexte sur lequel se base notre travail à savoir la notion d'agent, ses caractéristiques, ainsi que les systèmes multi-agents, leurs architectures, la communication des agents au sein des SMA et enfin la coopération des agents dans un système multi-agents. Nous présenterons aussi, un domaine particulier auquel s'applique les agents à savoir, celui de l'enseignement. Mais avant tout, il nous a semblé important d'introduire le domaine de l'intelligence artificielle distribuée (IAD), branche relativement récente de l'intelligence artificielle (IA) à laquelle appartiennent les systèmes multi-agents.

II.1 Le paradigme Agent (dans le sens origine)

II.1.1 Domaines de l'intelligence Artificielle (IA) et de l'intelligence Artificielle Distribuée (IAD)

L'expression de « IA » a été employée pour la première fois (1995-1970) par John McCarthy, qui fonde l'IA sur le postulat mécaniste qui veut que toute activité intelligente soit modélisable et reproductible par une machine.

L'IA peut donc être vu comme une tentative de modéliser certains aspects de la pensée humaine essayant ainsi de résoudre par ordinateur les problèmes que les humains résolvent plus rapidement : « *L'IA a pour but d'exécuter par ordinateur des tâches pour lesquelles l'homme dans un contexte donné est aujourd'hui meilleur que la machine* » (Alliot & al, 1994).

« *L'IA regroupe les disciplines visant la conception d'une entité reproduisant les activités intellectuelles humaines...* » Autrement dit, l'IA modélise le comportement intelligent d'une seule entité (aspect individuel).

L'approche classique de l'IA qui s'appuie sur une centralisation de l'expertise au sein d'un système unique ayant montré ses limites dans les domaines de l'aide à la décision ou la conduite des processus industriels par exemple, a permis l'apparition de travaux menés sur la concurrence et la distribution. Durant les années 70, ces travaux ont été à l'origine de la nouvelle discipline appelée Intelligence Artificielle Distribuée IAD qui propose la distribution de l'expertise sur un groupe d'agents devant être capables de travailler et d'agir dans un environnement commun et résoudre des éventuels conflits. C'est cette étape qui a introduit dans l'IAD des concepts empruntés aux sciences humaines tels que la coopération, la coordination d'actions, la négociation et la notion d'émergence.

« ... l'IAD regroupe les disciplines visant la conception des entités reproduisant les activités intellectuelles humaines.... », en d'autres termes l'IAD s'intéresse à des comportements intelligents qui sont le produit de **l'activité coopérative** de plusieurs entités (aspect collectif). Ces entités utilisent des technologies de l'IA et sont dites « agents ».

En général, les types de problèmes et les systèmes qui ont été construits pour les résoudre, concernent les tâches complexes qui impliquent la présence de modules spécialisés dans un domaine donné. Ainsi on construit autant d'entités (d'agents) qu'il y a de spécialités à convoquer pour la résolution d'un problème.

Pour résumer, l'IAD étudie la manière de répartir un problème sur un certain nombre d'entités coopératives (agents). Elle s'intéresse donc à la manière de coordonner le comportement intelligent d'un ensemble d'entités selon des lois sociales. Ces entités ou agents sont autonomes et interagissent dans un environnement pour la résolution de problèmes. La figure ci-dessous montre bien les intersections de plusieurs domaines qui ont abouti au paradigme d'agent :

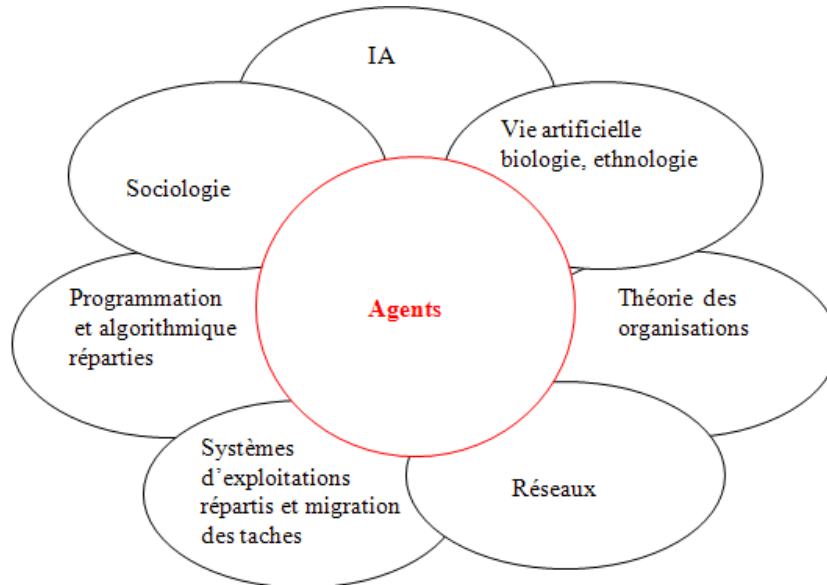


Figure II.1 : Intersection de plusieurs domaines pour le paradigme « Agent »

II.1.2 Programmation Orientée Agent (POA)

Depuis quelques années, il y a un effet de convergence qui se précise dans le sens d'une conception comportementaliste des objets informatiques. Cette tendance répond à des concepts d'agents, dits autonomes et dotés de buts propres par opposition à l'objet. Cette orientation se traduit plus globalement d'une nouvelle conception de la programmation plus modulaire, dite Programmation Orientée Agent. Il en ressort des architectures modulaires, communicantes, dynamiques, souvent complexes et à tendance émergentiste. Cette approche fait l'objet d'une forte diversification qui touche aussi bien le monde scientifique, le multimédia, l'informatique bien sûr, l'industrie, le commerce, ... etc (Gruselle, 2004).

Il convient de ne pas confondre "agent" et "objet". Tout d'abord, tout comme les agents, les objets encapsulent leur état interne (leurs données). Ils peuvent également poser des actions sur cet état par le biais de leurs méthodes et ils communiquent en s'envoyant des messages.

À ce niveau, ils diffèrent des agents par leur degré d'autonomie. En effet, une méthode doit être invoquée par un autre objet pour pouvoir accomplir ses effets. Un agent, quant à lui, recevra une requête et décidera de son propre gré s'il doit poser ou non une action.

Une seconde différence provient du caractère flexible (réactif, pro-actif et social)

du comportement d'un agent. Bien que certains disent qu'il est possible de bâtir un programme orienté objet qui intègre ces caractéristiques, on doit également voir que le modèle standard d'un objet ne dit rien à propos de ces types de comportements.

La troisième et dernière différence provient du fait que l'on considère un agent comme étant lui-même une source de contrôle au sein du système tandis que dans un système orienté-objet, on n'a qu'une seule source de contrôle.

Le modèle dit de "POA" est obtenu à partir de modèles de programmation antérieurs en rajoutant de plus en plus d'ouverture dans les architectures.

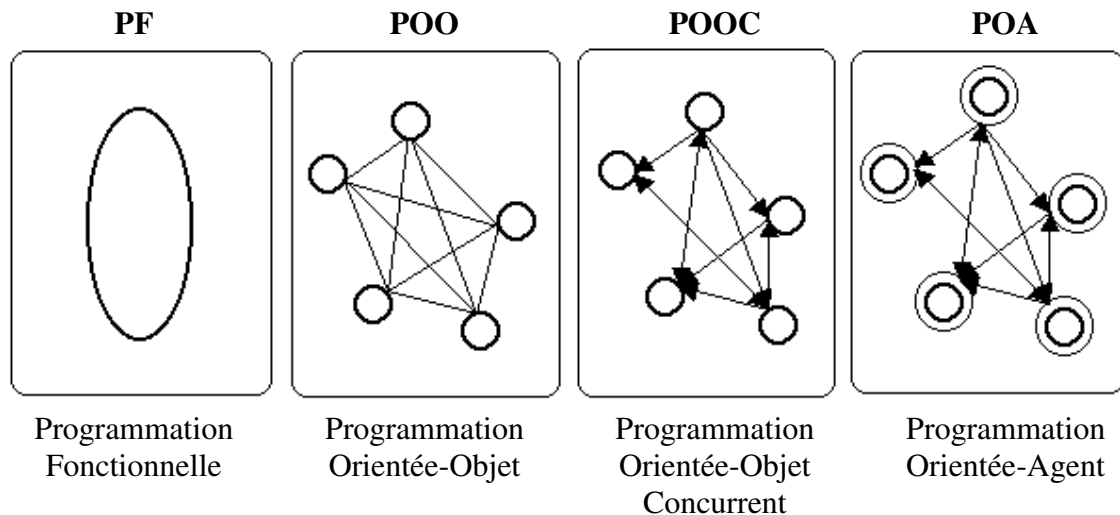


Figure II.2 : Evolution vers une programmation « Agent »

Afin de mieux comprendre ce modèle, nous présentons le modèle "Acteur" qui aboutit à une conception de systèmes distribués, en l'occurrence les systèmes ouverts. C'est un des premiers modèles proposés à l'époque de l'IAD. Selon Hewitt, les systèmes ouverts sont définis comme concurrents et asynchrones, ils dépeignent un contrôle décentralisé et sont extensibles. Les acteurs sont vus comme étant des objets actifs, autonomes et communicants par envoi de messages asynchrones avec les autres acteurs, ils peuvent être concurrents d'où la POOC.

Chaque acteur possède des accointances¹ et obéit à un script². Ils sont composés d'un ensemble de primitives, parmi lesquelles on retrouve :

- ❖ Create : créer un acteur à partir d'un ensemble de paramètres décrivant son comportement ;
- ❖ Send : envoyer un message à un autre acteur ;
- ❖ Become : changer l'état local d'un acteur.

Les acteurs s'avèrent être un modèle assez naturel pour le calcul parallèle. Cependant, les divers modèles d'acteurs, comme bien d'autres modèles de l'intelligence artificielle distribuée font face à un problème de cohérence. Leur granularité³ fine pose des problèmes de comportement dans des systèmes qui renferment plusieurs acteurs. Ils éprouvent également des difficultés à atteindre des buts globaux avec seulement des connaissances locales.

¹ Accointance : les acteurs connus d'un acteur sont dits accointance.

² Script : le comportement d'un acteur est appelé script.

³ La granularité d'un agent est le degré de détail des connaissances de l'agent. Elle exprime la complexité des fonctionnalités d'un agent.

II.2 L'Agent

II.2.1 Définitions

Le terme Agent (du latin « agere », (agir)) désignait déjà et entre autre, tout phénomène physique qui a une action déterminée (langue scientifique, ex : les agents atmosphériques, les agents infectieux) et plus généralement désigne celui qui est chargé d'une mission par une société, un gouvernement, un particulier, par exemple un agent de change, un agent immobilier.

Le concept d'agent a été l'objet d'études dans différentes disciplines. Il a été non seulement utilisé dans les systèmes à base de connaissances, la robotique, le langage naturel et d'autres domaines de l'intelligence artificielle, mais aussi dans des disciplines comme la philosophie et la psychologie. Aujourd'hui, avec l'avènement de nouvelles technologies et l'expansion de l'Internet, ce concept est encore associé à plusieurs nouvelles applications comme « agent ressource », « agent courtier », « assistant personnel », « agent interface », « agent ontologique », ... etc.

Dans la littérature, on trouve une multitude de définitions d'agents. Elles se ressemblent toutes, mais diffèrent selon le type d'application pour laquelle est conçu l'agent. Nous trouvons différentes définitions attribuées aux agents ainsi que la différence entre un agent et un programme classique.

D'après J.Ferber (Ferber, 1995) :

« un agent est une entité physique ou virtuelle agissant dans un environnement, communiquant directement avec d'autres agents, possédant des ressources propres, capable de percevoir partiellement son environnement, disposant d'une représentation partielle de l'environnement. Son comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont il dispose, et en fonction de sa perception, de ses représentations et des communications qu'il reçoit. »

Il ressort de cette définition des propriétés clés comme l'autonomie, l'action, la perception et la communication. D'autres propriétés peuvent être attribuées aux agents. Nous citons en particulier la réactivité, la rationalité, l'engagement et l'intention.

Jennings, Sycara et Wooldridge, dans (Jennings & al, 1998), ont proposé la définition suivante pour un agent :

« Un agent est un programme informatique situé dans un environnement doté de comportements autonomes (actions) lui permettant d'atteindre dans cet environnement les objectifs qui lui ont été fixés à sa conception ».

Nous pouvons aussi envisager de définir un agent par ses propriétés :

- *Est capable d'agir dans son environnement.*
- *Est un composant logiciel réutilisable.*
- *Peut communiquer avec d'autres agents.*
- *Il agit de manière réactive et proactive⁴.*

⁴ Pro-active : se dit d'une chose qui a effet sur une autre qui vient après.

- *Est doué d'autonomie et est mû par un ensemble de tendances (objectifs individuels).*
- *Possède des ressources propres.*
- *Est capable de percevoir son environnement et de s'adapter à ses modifications (c'est donc une entité adaptative).*
- *Ne dispose que d'une représentation partielle de son environnement.*
- *Possède des compétences et offre des services.*
- *Peut éventuellement se reproduire.*

Les agents suivent deux tendances : une tendance sociale tournée vers la collectivité (les mécanismes et connaissances associés concernant les activités du groupe) et une tendance individuelle avec mécanismes et connaissances associés concernant les règles de fonctionnement interne de l'agent. On peut caractériser un agent par son rôle, sa spécialité, ses objectifs et ses fonctionnalités, ses croyances et ses capacités d'apprentissage.

II.2.2 Structure générale d'un agent

La figure suivante décrit la structure générale d'un agent :

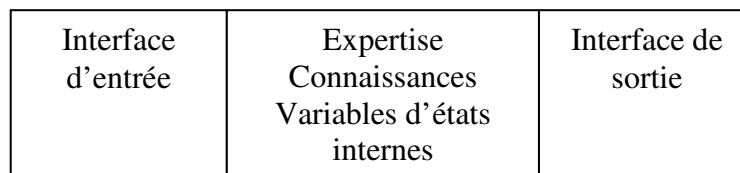


Figure II.3 : Structure générale d'un agent

- ✓ Interface d'entrée /sortie : elle assure la communication avec le monde extérieur à travers un protocole d'échange. Les variables d'états sont visibles de l'extérieur et sont gérées par les interfaces.
- ✓ Variable d'état : elle caractérise l'état d'un agent.
- ✓ Partie expertise : elle contient les traitements et les actions qu'un agent peut réaliser, c'est donc un ensemble de règles.

II.2.3 Déterminant d'un agent

Le déterminant d'un agent est l'ensemble nécessaire et suffisant de ses caractéristiques structurelles, environnementales et comportementales qui permet d'expliquer sa façon d'agir.

Les caractéristiques environnementales d'un agent sont liées à la représentation que se fait l'agent de son environnement et de lui-même. Les caractéristiques structurelles déterminent l'ensemble de ses composants, alors que ses caractéristiques comportementales contraignent l'ensemble de ses comportements, en accord avec les caractéristiques environnementales.

Citons ci-après quelques caractéristiques multidimensionnelles des agents qui en font des programmes différents des programmes conventionnels :

- ✿ **La nature** : agent physique ou virtuel.

- ✿ **L'autonomie** : un agent est plus ou moins indépendant de l'utilisateur, des autres agents et des besoins en ressources 'énergétique' (CPU, mémoire, ... etc).
- ✿ **L'environnement** : c'est l'espace dans lequel va agir l'agent ; celui-ci peut se réduire au réseau constitué par l'ensemble des agents.
- ✿ **La capacité représentationnelle** : l'agent peut avoir une vision très locale de son environnement mais il peut aussi avoir une représentation plus large de cet environnement et notamment des agents qui l'entourent.
- ✿ **L'objectif (ou dimension téléonomique)** : l'agent peut poursuivre le but global du système, peut satisfaire ses objectifs propres ou même se comporter dans la perspective de s'absoudre à une fonction de survie.
- ✿ **La perception** : de l'environnement par l'agent.
- ✿ **La communication** : l'agent aura plus ou moins la capacité de communiquer avec les autres agents.
- ✿ **Le raisonnement** : l'agent peut être lié à un système expert ou à d'autres mécanismes de raisonnements plus ou moins experts.
- ✿ **La quantité de ses congénères** : le système peut contenir de quelques-uns à plusieurs milliers d'agents.
- ✿ **Le contrôle** : il peut être totalement distribué entre les agents mais il peut être voué à une certaine classe d'agents comme les agents facilitateurs.
- ✿ **L'anticipation** : l'agent peut plus ou moins avoir les capacités d'anticiper les événements futurs.
- ✿ **La granularité ou la complexité** : l'agent peut être très simple comme un neurone mais aussi plus complexe.
- ✿ **La reproduction** : les agents peuvent se reproduire.
- ✿ **L'apprentissage** : un agent aura plus ou moins tendance à retirer, stocker et réutiliser des informations externes ou reçues de son environnement.
- ✿ **La contribution** : l'agent participe plus ou moins à la résolution du problème ou à l'activité globale du système.
- ✿ **L'efficacité** : l'agent et sa rapidité d'exécution, d'intervention.
- ✿ **La bienveillance** : l'agent a plus ou moins le devoir d'aider ses congénères plutôt que de s'opposer à eux.

II.2.4 Typologies d'agents

Les agents sont classés selon les degrés des caractéristiques (considérées très importantes en IAD) suivantes : autonomie, coopération et adaptation. Le plus haut degré d'autonomie permet à l'agent de planifier ses actions, le plus haut degré de coopération lui accorde des capacités de négociation, et finalement le plus haut niveau d'adaptation lui permet d'acquérir et d'adapter ses connaissances.

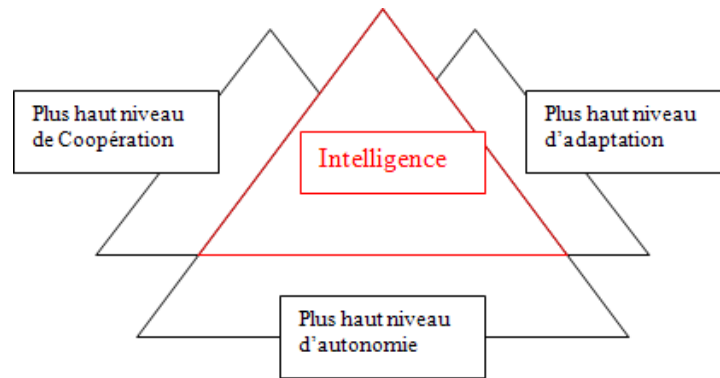


Figure II.4 : La relation qui existe entre les trois principales caractéristiques d'agent

Nous pouvons distinguer quatre types d'agents :

Agent naturel : il s'agit d'un agent doté de possibilités d'auto-apprentissage, utilisé en intelligence artificielle. Il a la faculté d'augmenter ses connaissances, par ajout de règles en fonction de l'évolution de son environnement et des événements qui peuvent se produire.

Agent réactif : capable de réagir à des événements externes pour assurer soit la conduite ou le pilotage d'un équipement matériel dont il est la partie commande, soit la supervision et la coordination des activités d'un ensemble d'agents dont il est l'agent de contrôle (partie de contrôle d'une application multi-agents (AMA)).

Agent de traitement : son rôle est d'effectuer des transformations ou des calculs sur des flots de données en entrée pour produire des flots de données en sortie, il assure l'exécution d'un algorithme de complexité quelconque (partie opérative d'une AMA).

Agent cognitif (ou délibératif) : il ne contiendra pas uniquement des données et des méthodes procédurales mais aussi une base de connaissances et qui en outre initie des inférences sur sa base de connaissances propre et sur celle des autres de même type (peut agir sur une base de données).

Dans le cadre de ce travail, nous nous intéressons particulièrement aux agents réactifs et cognitifs à propos de qui nous allons donner plus de détails.

II.2.4.1 Les agents réactifs

Leur comportement est basé sur les stimulus-réponse. Ils ne disposent ni de représentation du monde ni de représentation de leurs actions qui, prises individuellement, consistent en une réaction rapide face à un stimulus. Les agents réactifs possèdent une faible capacité de coopération qui est généralement limitée à la communication par messages.

La figure ci-dessous montre clairement le fonctionnement d'un agent réactif :

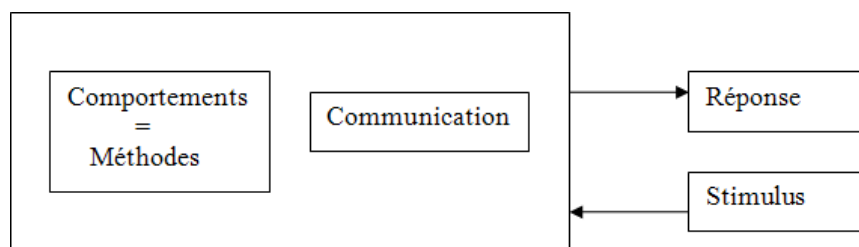


Figure II.5 : Fonctionnement d'un agent réactif

II.2.4.2 Les agents cognitifs ou délibératifs

Leur comportement est plus « réfléchi », c'est-à-dire, qu'il résulte d'un choix effectué parmi un ensemble d'actions possibles. Ces agents possèdent une représentation symbolique de l'environnement sur lequel ils peuvent raisonner.

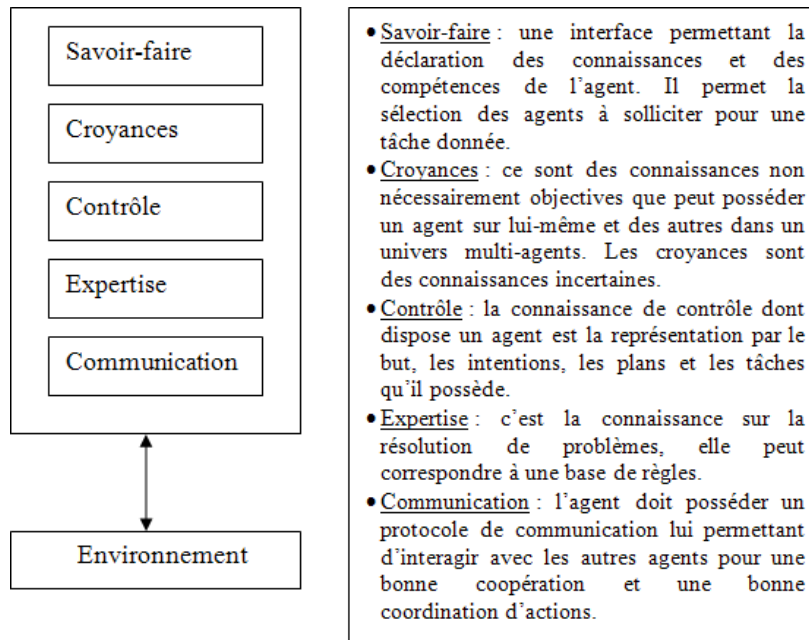


Figure II.6 : Représentation de la structure interne d'agent cognitif

Les agents cognitifs regroupent plusieurs sous-types d'agents qui sont les suivants :

1. Agents « intelligents »

Combinent les trois caractéristiques citées précédemment à leur plus haut niveau. Ils planifient leurs actions, sont capables de négocier avec d'autres agents, d'acquérir et de modifier leurs connaissances. Ces agents sont moins 'réactifs' aux modifications brutales de leur environnement.

2. Agents « collaborateurs »

Ce sont des agents cognitifs non apprenants, ils sont fortement autonomes et coopérants mais leur adaptabilité est très limitée. La collaboration entre de tels agents est plus élaborée que dans le cas d'agents réactifs car les agents collaborateurs sont capables de négocier entre eux ou avec l'utilisateur. Ces derniers sont généralement utilisés dans les maintenances de réseaux ou pour simuler le comportement d'organisations humaines ou animales.

3. Agents « interfaces »

Leur fonction consiste à capturer les actions de l'utilisateur (le plus souvent actions sur clavier ou souris mais il est également possible de capturer la voix ou l'expression du visage) ils sont le plus souvent utilisés dans certains logiciels bureautiques. Leur capacité de coopération est utilisée à l'échange d'information concernant les actions de l'utilisateur.

4. Agents « informations »

Ces agents sont dédiés à la recherche d'informations sur le net. Ces agents possèdent une grande autonomie ; ils agissent seuls, soit en fonction d'un calendrier,

soit en fonction d'un manque d'informations, soit en fonction d'une nouvelle disposition d'informations. Ils sont capables d'adapter leur fonctionnement en fonction du besoin de l'utilisateur ou de la quantité de l'information (exemple : si un nouveau site propose des informations plus pertinentes, il sera alors privilégié pour des recherches futures).

Le problème avec ce genre d'agents est qu'ils agissent le plus souvent de manière isolée, ce qui peut entraîner des problèmes de redondance de l'information. Il est à noter que l'on trouve également la notion d'« *agent mobile* ». Il s'agit d'une caractéristique supplémentaire à un agent information lui permettant de parcourir le réseau à la recherche d'informations.

Nous pouvons résumer les différents types d'agents cognitifs ainsi que leurs degrés d'autonomie, d'apprentissage et de coopération comme suit : Les *agents collaborateurs* sont autonomes et coopèrent, leur capacité de négociation implique une faculté d'adaptation. Les *agents interfaces* réagissent aux sollicitations de l'utilisateur, ils sont donc peu autonome. Les *agents informations* peuvent agir de façon très autonome, ils sont capables d'adapter leur stratégie de recherche en fonction des informations trouvées ou non.

A. Caractéristiques d'un agent cognitif

L'intentionnalité : un agent intentionnel est un agent guidé par ses buts. Une intention (Searle, 1990), (Tuomela & al, 1988) est la déclaration explicite des buts et des moyens d'y parvenir. Elle exprime donc la volonté d'un agent d'atteindre un but ou d'effectuer une action.

La rationalité : un agent rationnel est un agent qui suit le principe dit « principe de rationalité » suivant (Newell, 1982) : « *si un agent sait qu'une de ses actions lui permet d'atteindre un de ses buts, il la sélectionne* ». Les agents cognitifs disposent de critères d'évaluation de leurs actions, et sélectionnent selon ces critères les meilleures actions qui leur permettent d'atteindre le but. De tels agents sont capables de justifier leurs décisions. La rationalité se rapporte au comportement cognitif de l'agent. Ce terme qualifie l'utilisation efficace des ressources par l'agent.

L'engagement : la notion d'engagement (Becker, 1960), (Bond, 1990) est l'une des caractéristiques essentielles des agents coopératifs. Un agent coopératif planifie ses actions par coordination et négociation avec les autres agents. En construisant un plan pour atteindre ses buts, l'agent se donne les moyens d'y parvenir et donc, s'engage à accomplir les actions qui satisfont ce but : l'agent croit qu'il est en mesure d'exécuter tout le plan qu'il a élaboré, ce qui le conduit (ainsi que les autres agents) à agir en conséquence.

B. Fonctionnement d'un agent cognitif

Lorsqu'un agent perçoit une situation dans l'environnement, il essaie de le reconnaître. Si la situation lui est familière, il peut enclencher un processus de planification afin de résoudre le problème. Il peut aussi reconnaître la situation en terme d'action et donc, passer à l'exécution de la tâche (Reconnaissance - Exécution). Lorsque l'agent perçoit des situations qu'il connaît très bien, il peut faire intervenir son comportement réactif en passant directement à l'action (Perception - Exécution). S'il ne peut pas résoudre un problème (Situation non familière), il engage un processus de

coopération pour demander de l'aide aux autres agents (Reconnaissance – Prise de décisions). Ceci est illustré sur le schéma ci-dessous :

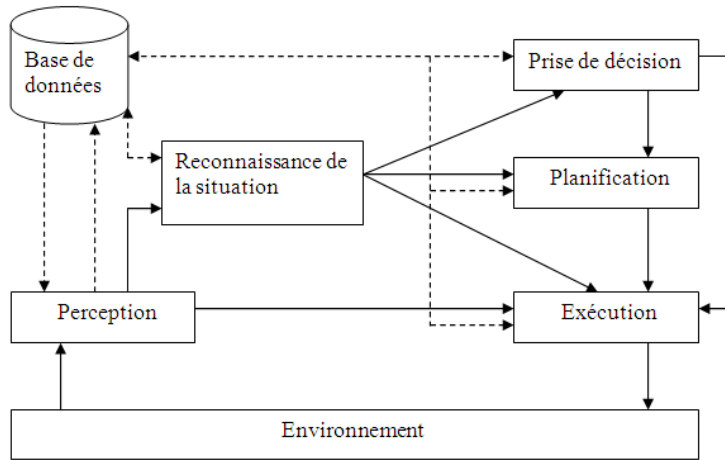


Figure II.7 : Fonctionnement d'un agent cognitif (a)

Le fonctionnement d'un agent peut être plus affiné dans ce qui suit :

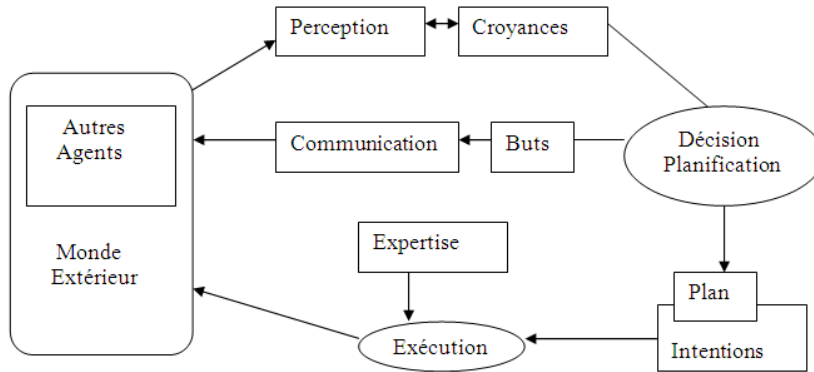


Figure II.8 : Fonctionnement d'un agent cognitif (b)

Perception : La perception fournit les capacités d'entrée nécessaire à l'agent. La principale fonction de la perception est d'extraire les informations sur les entités, évènements et états du monde extérieur gardant ainsi le modèle de ce dernier et faisant des mises à jour si nécessaire. Un agent peut puiser ses connaissances depuis plusieurs origines :

- ✿ Le savoir initial de l'agent.
- ✿ La communication avec les autres agents.
- ✿ La perception de soi et des autres.

Les informations parvenant à l'agent de sa perception et de son savoir-faire initial sont des informations certaines, cependant les informations provenant des autres agents sont considérées comme incertaines.

Prise de décision : Durant son exécution, un agent se fixe un certain nombre de buts suite à ses observations et à ses interactions avec le monde. Il se trouve donc confronté au problème de la sélection du but à satisfaire en premier et pour chaque but, l'action qui permet de l'atteindre.

Planification : La planification est le processus qui permet la construction de la séquence d'actions (plan d'actions) à réaliser pour atteindre un certain but. Le plan est constitué d'intentions, d'engagements et de croyances sur l'univers qui l'entoure.

Nous avons cité précédemment qu'un agent est intelligent s'il a l'ensemble des caractéristiques suivantes : une capacité sociale (coopération), une capacité à apprendre (adaptation) et un haut degré d'autonomie. Pour doter l'agent de cette intelligence, il doit intégrer les modules suivants :

- ✿ Un module permettant de mémoriser le savoir de l'agent qui sera le plus souvent fait par la réalisation d'un système de base de données (SGBD). Ce module sera nommé « base de connaissances ».
- ✿ Un module permettant de connaître l'ontologie de la « base de connaissances ». Il sera appelé « le contrôleur d'accès » qui reposera sur un SGBD, dit « intelligent ».
- ✿ Un module permettant à l'agent d'avoir une vision de son environnement, « l'interface de l'agent ».
- ✿ Un module permettant la génération de nouvelles connaissances, il sera nommé « moteur d'apprentissage ».
- ✿ Un module permettant le raisonnement, il sera nommé « moteur de raisonnement ».

Ces modules devront être agencés de telle façon que le flux d'information qu'ils génèrent, permet de caractériser le concept d'intelligence. Un des modèles les plus souvent rencontrés est représenté par le schéma suivant :

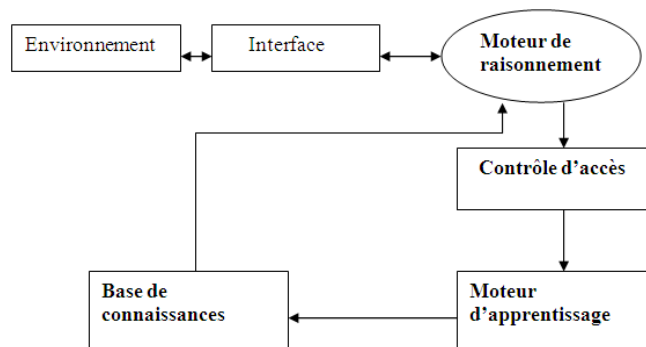


Figure II.9 : Modèle d'agencement des modules d'un agent permettant de caractériser son intelligence

II.2.4.3 Les agents cognitifs par opposition aux agents réactifs

La granularité des agents impliqués dans une application varie selon deux écoles coexistantes aujourd'hui : l'école cognitive et l'école réactive. Suivant les types d'agent utilisé, on parlera de systèmes cognitifs ou de systèmes réactifs.

Les systèmes cognitifs sont fondés sur la coopération d'agents capables à eux seuls d'effectuer des opérations complexes. Un tel système comprend un petit nombre d'agents.

Les systèmes d'agents réactifs sont fondés sur le fait que dans un SMA, il n'est pas nécessaire que chaque agent soit individuellement «intelligent» pour arriver à un comportement global intelligent. Un tel système comprend un nombre beaucoup plus grand d'agents.

Il est possible de doter les agents cognitifs de capacités de réaction aux événements : on parlera alors d'agents hybrides (Chaib-Draa, 1993). Le tableau ci-dessous montre les différences entre les systèmes d'agents cognitifs et les systèmes d'agents réactifs

Système d'agents cognitifs	Système d'agents réactifs
Représentation explicite de l'environnement	Pas de représentation explicite
Peut tenir compte de son passé	Pas de mémoire de son historique
Agent complexe	Fonctionnement stimulus/action
Petit nombre d'agents	Grand nombre d'agents

Table II.1 : Comparaison Agent cognitif / Agent Réactif.

Il existe une nouvelle typologie d'agents beaucoup plus complète dont certains sont construits sur des architectures constituées de plusieurs couches. Certaines incluent **un niveau réactif** pour des comportements de type stimulus-réponse, un second **niveau de type cognitif délibératif** pour des comportements survenant à la suite d'agencement de primitives comportementales et un troisième niveau tendant à être un **méta-niveau** afin de doter les agents d'une forme de conscience.

(Frasson & al, 1996) ont proposés le modèle d'agent "ACTOR" qui implémente une architecture construite selon trois paradigmes : *déterministe*, *rationaliste* et *émergentiste* et implémentée avec trois niveaux de description (réactif, adaptatif et cognitif) afin de supporter les comportements individuels et collectifs de chacun des acteurs.

- ✱ Le premier niveau réactif (déterministe) n'a pas de capacité d'apprentissage et permet des comportements réflexifs sur une reconnaissance plus ou moins précise de patterns ou modèles prédéfinis. Restreint à ce genre d'action, un "actor" est un agent réactif.
- ✱ Au deuxième niveau, l'apprentissage planifié (rationalité) autorise la modélisation de la manière dont les agents à stratégie (niveau tactique) se comportent relativement à leurs intentions envers chacun des autres agents, un "actor" étant considéré en tant qu'agent adaptatif.
- ✱ Le niveau cognitif permettant l'apprentissage (émergence) autorise également la modélisation des modes de raisonnement employés par l'actor apprenant ou/et enseignant qui explore les options alternatives pour la satisfaction des buts ; un actor est considéré ici en tant qu'agent cognitif.

Dans la littérature, on parle d'agents « affect » ou « à émotion ». Les modèles basés sur les émotions ou les affects ou encore sur l'esprit humain, sont imminents. Le choix des types d'agents à utiliser dépend du système multi-agents le plus pertinent pour le problème à résoudre.

II.3 Les systèmes multi-agents

Le thème des systèmes multi-agents (SMA), s'il n'est pas récent, est un champ de recherche très actif. Cette discipline est à la connexion de plusieurs domaines en particulier celui de l'intelligence artificielle, des systèmes informatiques distribués et du génie logiciel (Jarras & al, 2002). On peut dire que les SMA ont vu le jour avec l'avènement de l'IAD. À ses débuts toutefois, l'IAD ne s'intéressait qu'à la coopération entre solveurs de problèmes afin de contribuer à résoudre un but commun. Pour y parvenir, on divisait en général, un problème en sous-problèmes et on allouait ces sous-problèmes à différents solveurs qui sont appelés à coopérer pour élaborer des solutions partielles. Celles-ci sont finalement synthétisées en une réponse globale au problème de départ. Ainsi donc, l'IAD au départ privilégiait le "problème à résoudre"

tout en mettant l'accent sur la résolution d'un tel problème par de multiples entités intelligentes.

Dans les SMA(s) d'aujourd'hui, les agents sont (entre autres) autonomes, possiblement préexistants et généralement hétérogènes. Dans ce cas, l'accent est plutôt mis sur le fait de savoir comment les agents vont coordonner leurs connaissances, buts et plans pour agir et résoudre des problèmes.

II.3.1 Définition

Un SMA est un système qui se compose d'un ensemble d'entités spécialisées coopérantes pour le compte d'une application globale (fonction globale ou but commun). Les SMA(s) ont des applications dans le domaine de l'intelligence artificielle où ils permettent de réduire la complexité de la résolution d'un problème en divisant le savoir nécessaire en sous-ensembles et en coordonnant l'activité de ses agents (Mieux vaut plusieurs agents ou entités spécialisés qu'un seul omniscient) (Ferber, 1995).

Un SMA est composé d'un ensemble d'objets situés dans un environnement (ou un espace). Parmi ces objets certains sont des agents, d'autres sont des objets passifs qui peuvent être utilisés, détruits, modifiés, créés par des agents. Un ensemble de relations entre les objets de l'environnement est défini comprenant des relations entre agents, entre objets passifs et entre objets passifs et agents.

Les SMA(s) mettent en œuvre des agents homogènes et hétérogènes ayant des buts communs ou distincts. Un système multi-agents est un système distribué composé d'un ensemble d'agents qui interagissent le plus souvent, selon un mode de coopération, de concurrence ou de coexistence.

➤ **Caractéristiques d'un système multi-agents (Chaib-draa, 1993)**

1. Chaque agent a des informations ou des capacités de résolution de problèmes limités, ainsi chaque agent a un point de vue partiel.
2. Il n'y a aucun control global du système multi-agents.
3. Les données sont décentralisées.
4. Le calcul est asynchrone.

II.3.2 Environnement & Interaction

Le monde pour un agent = SMA + l'environnement :

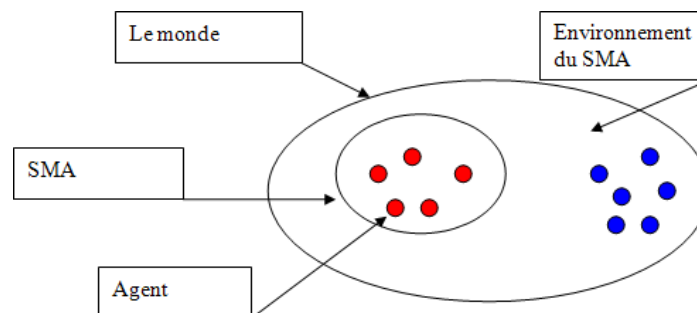


Figure II.10 : Positionnement d'un agent dans le monde (Adam, 2000)

- ✿ L'environnement est la représentation du monde dans lequel les agents se situent. Il est modifiable soit de façon globale soit en faisant une distinction entre les objets passifs (soumis aux actions des agents) et les entités actives (agents).

- ✿ L'interaction est la mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Elle s'effectue par la communication, les actes de langages et les protocoles d'interaction.

II.3.3 Société d'agents

II.3.3.1 L'organisation sociale

Comme pour les organisations humaines, à la société d'agents peut être associée une organisation sociale qui est la façon dont le groupe est constitué, à un instant donné, pour pouvoir fonctionner. Elle décrit la liste de composants fonctionnels du système, leurs natures, leurs responsabilités et leurs besoins en ressources ainsi que les liens de communication entre les agents. Selon (Gasser, 1990), une société d'agents est constituée de trois éléments :

- ✿ un ensemble d'agents ;
- ✿ un ensemble de tâches à réaliser ;
- ✿ un ensemble d'objets associés à l'environnement ;

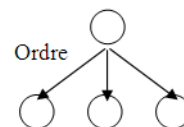
Un agent peut être responsable d'effectuer une tâche s'il en a la capacité par rapport à sa qualification et ses compétences. Il prend alors un rôle dans le groupe. La réalisation d'une tâche suppose la manipulation d'objets de l'environnement. Le groupe se réorganise à chaque fois en fonction de la tâche à accomplir (organisation dynamique). Les structures organisationnelles en IAD se différencient selon trois facteurs : le type de décentralisation, le type d'agents et le mode de coopération.

Dans le cas d'agents réactifs il n'existe pas de réelle organisation, elle émerge des comportements de ses membres et non d'une volonté organisationnelle (Steele, 1990). Par contre, dans le cas des agents cognitifs, l'organisation est requise puisque les agents communiquent selon un protocole et peuvent avoir des rôles durant la résolution de problèmes en fonction de leur nombre et localisation. Selon son niveau décisionnel, un agent peut être opérateur, superviseur ou autre.

Les chercheurs en IAD essayent d'adapter les modèles sociologiques aux systèmes multi – agents (et on parlera alors d'organisation multi-agents OMA⁵). (Mintzberg, 1986) distingue cinq types de configurations organisationnelles qui dépendent du type de décentralisation, du type principal de coordination et des éléments principaux de l'organisation (individus ou autres). Cela va de la structure simple (hiérarchie pyramidale totalement centralisée) à une bureaucratie professionnelle (fonctionnement par ajustement vertical ou horizontal) en passant par l'adhocratie (fonctionnement par ajustement mutuel entre les petits groupes de l'organisation). Ces structures sont schématisées comme suit :

❖ Structure hiérarchique

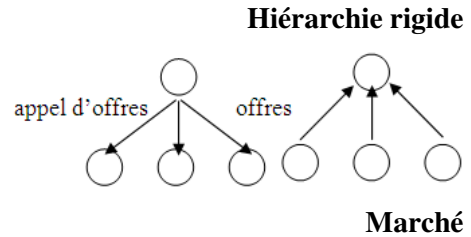
OMA à structure hiérarchique rigide ayant un seul objectif opérationnel, le système y est centralisé.



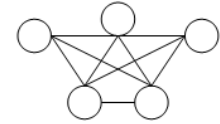
⁵ OMA : tout SMA peut être qualifié d'OMA lorsque les agents qui le composent possèdent une connaissance des autres et qu'ils suivent les règles fixées par un système global tout en gardant leur autonomie

❖ **Structure hétérarchique**

- OMA type marché, constitué d'agents coordinateurs et d'agents exécutants. Cette structure est un peu plus décentralisée que la première.

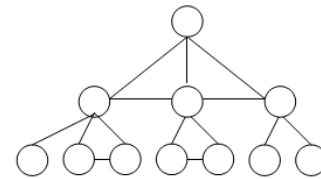


- OMA de type communauté, décrivant un contrôle fortement décentralisé, les membres possèdent les mêmes capacités et poursuivent des objectifs multiples



Communauté

- OMA de type société, le contrôle est décentralisé autour d'unités de même ou de différents types poursuivant des objectifs opérationnels différents et qui s'ajustent grâce aux principes de négociation.



Société

Figure II.11 : Structures organisationnelles d'agents

Le choix d'une structure organisationnelle dépend à la fois des caractéristiques du groupe et de la tâche à accomplir, une tâche étant caractérisée par la complexité du raisonnement, la quantité des connaissances et la capacité d'action.

Les solutions proposées par les systèmes multi-agents pour résoudre le problème de la distribution de l'information sont nombreuses, et ceci dépend du type d'organisations des entreprises. En fait, pour que l'organisation multi-agent soit proche de l'organisation humaine, un modèle organisationnel "holonique" a été proposé dans (Adam, 2000) comme modèle unificateur dédié aux systèmes administratifs complexes, dans lequel on parle d'agents holoniques.

II.4 La communication des agents

Dans les SMA(s), les agents ne disposent d'aucune mémoire commune. Le mécanisme de communication entre les agents repose explicitement sur les mécanismes d'envoi de messages, de réception et de synchronisation.

II.4.1 Protocole de communication

Dans un système distribué, il faut que les entités disposent d'un langage commun pour pouvoir échanger des informations et coopérer pour la résolution de problèmes. Ce protocole de communication, connu par chaque entité, repose sur les primitives de bases suivantes :

- ✱ Établissement d'une connexion entre les deux entités.
- ✱ Identification du nœud destinataire dans un réseau de communication.
- ✱ Envoi et réception de données.
- ✱ Définition d'un type de communication : synchrone ou asynchrone.

Ces protocoles de communications sont des règles nécessaires à la communication, ils permettent de structurer et d'uniformiser les interactions inter-agents.

II.4.2 Langage de communication entre agents

Il y a deux principales approches pour concevoir un langage de communication entre agents. La première approche est une approche procédurale où la communication est basée sur le contenu exécutable. Ceci pourrait être réalisé en utilisant les langages de programmation tels que Java par exemple. La seconde approche déclarative, où la communication est basée sur des déclarations telles que des définitions et des hypothèses. Un des langages déclaratifs d'agent le plus utilisé est KQML⁶.

II.4.3 Architecture de communication

La communication entre agents peut être organisée suivant trois schémas différents :

- ✱ **Réseaux en anneau ou en bus** : où un message destiné à un agent doit passer par n-1 agents.
- ✱ **Réseaux en étoile** : où le message doit transiter par un agent superviseur avant d'arriver à l'agent destinataire. La nature bidirectionnelle des connexions rend complexe la gestion des interactions inter-agents.
- ✱ **Réseaux hybrides** : ce type d'organisation est une solution hybride traitant deux types d'organisation : un réseau en bus et un en étoile.

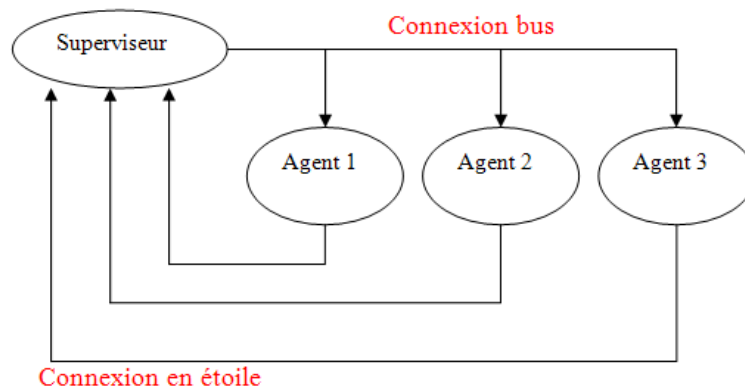


Figure II.12 : Architecture d'un réseau hybride

Dans (Cammarata & al, 1983), il y a une distinction entre les types de communication :

- ✱ **communication sélective ou diffusée** : par opposition à la diffusion, une communication sélective s'adresse à un nombre restreint d'agents, elle suppose donc un critère de sélection.
- ✱ **communication sollicitée ou non sollicitée** : une communication peut être demandée par un autre agent.
- ✱ **communication avec ou sans accusé de réception** : dans le cas de la communication avec accusé de réception, l'émetteur attend une confirmation de la bonne réception du message
- ✱ **communication à transmission simple ou répétée** : le message peut être répété sur plusieurs envois.

⁶ Knowledge Query and Manipulation Language.

II.4.4 Modes de communication

Il existe deux principaux modes de communication :

➤ **La communication par envoi de messages**

Les agents sont en liaison directe et envoient leurs messages directement et explicitement au destinataire (distribution totale à la fois de reconnaissance, des résultats et des méthodes utilisées pour la résolution de problème).

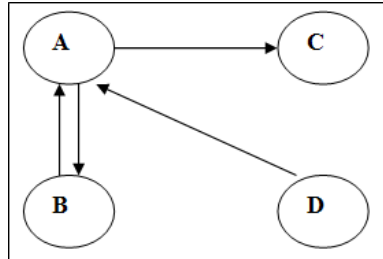


Figure II.13 : Communication par envoi de messages

➤ **La communication par partage d'information :**

Les agents ne sont pas en liaison directe mais communiquent via une structure de données partagée, où on trouve les connaissances relatives à la résolution. Cette manière est la plus utilisée dans les systèmes multi-experts. Par exemple, l'architecture "Blackboard" où l'on parle plus de source de connaissances que d'agents. Ce mode de communication n'existe pas dans les SMA où l'on ne dispose que d'une vision partielle du système, alors que la communication par partage d'information, suppose l'existence d'une base partagée sur laquelle les composants viennent lire et écrire.

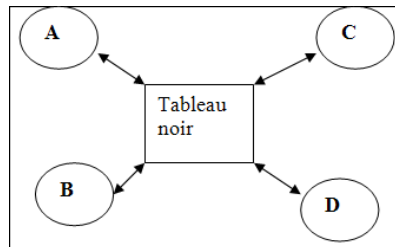


Figure II.14 : Communication par partage d'information

II.5 Contrôle et prise de décision au niveau des SMA(s)

Dans les SMA(s), le contrôle peut être exclusivement le résultat de mécanismes de contrôles locaux aux agents alors que dans les sociétés hiérarchisées, certains agents (superviseurs) peuvent prendre en main le contrôle de quelques éléments de la société (rapport maître-esclave).

Les mécanismes de la décision font référence à tout ce qui est allocation de tâches ou de ressources, en relation avec la résolution de conflit. Ils regroupent, entre autres, toute la partie décisionnelle des protocoles de négociation.

II.6 La coopération au niveau des SMA(s)

La coopération est une caractéristique très importante dans les SMA (Cammarata & al, 1983), (Dufree & al, 1989). En effet, une résolution distribuée d'un problème est le résultat de l'interaction coopérative entre les différents agents.

Dans une telle dynamique collective, un agent doit disposer en plus de la connaissance reflétant son degré d'implication (croyances, buts, intentions, engagements, modèle de soi et d'autrui), d'un certain nombre de compétences nécessaires pour la coopération. Il doit pouvoir mettre à jour le modèle du monde environnant, intégrer les informations venant d'autres agents et déléguer la tâche qu'il ne sait pas résoudre à un agent dont il connaît les compétences. Ces caractéristiques constituent les qualités essentielles d'un agent coopératif.

Selon (Bond, 1990), il existe deux architectures d'organisation pour les sociétés d'agents :

Structure horizontale : où les agents sont au même niveau, c'est le cas, par exemple, d'un groupe d'agents ayant des spécialités différentes et travaillant pour la résolution d'un même problème.

Structure verticale : où les agents sont structurés par niveaux (traitant différents niveaux d'abstraction de problème). Dans un même niveau on trouve localement une structure horizontale.

Quelle que soit l'organisation d'une société d'agents, un agent peut coopérer suivant différents modes comme on le montre dans ce qui suit.

II.6.1 Modes de Coopération

➤ **Coopération par partage de tâches**

Le partage de tâches est une forme de coopération entre les agents du système, adaptée surtout aux problèmes facilement décomposables en sous problèmes. Le problème est distribué entre les différents agents travaillant indépendamment les uns des autres. Chaque agent dispose des ressources et des compétences nécessaires pour accomplir la tâche qui lui a été assignée. Le contrôle est dirigé par les buts et les agents sont représentés par les tâches qu'ils se sont engagés à exécuter.

➤ **Coopération par partage de résultats**

Les agents ne peuvent accomplir leurs tâches de manière indépendante. Ils sont appelés à se transmettre mutuellement des résultats partiels.

➤ **Coopération par commande (maître/esclave)**

Dans ce mode, un agent superviseur décompose la tâche à effectuer en sous-tâches qu'il répartit entre les autres agents qui lui paraissent les plus appropriés pour la résolution. Ceux-ci résolvent les sous-tâches et renvoient les solutions partielles à l'agent superviseur. Ainsi il n'existe donc que deux types de messages entre les agents : Les messages de commandes et les messages de résultats partiels des sous-tâches comme le montre le schéma suivant :

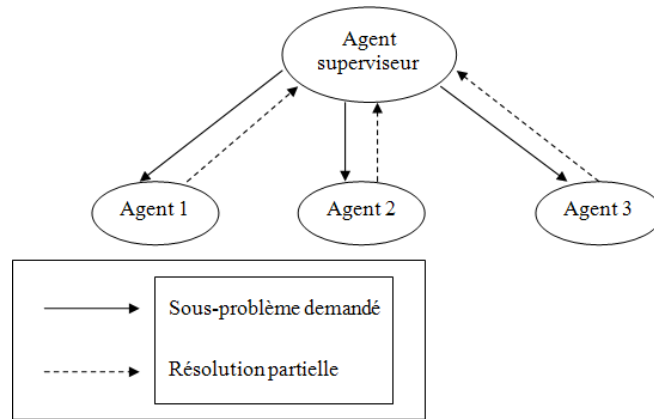


Figure II.15 : Mode de coopération par commande.

➤ **Coopération par compétition**

Un agent A décompose et diffuse la liste de sous problèmes comme le mode d'appel d'offre. Chaque agent X_i résout un ou plusieurs sous problèmes et envoie les résultats correspondants à A qui à son tour fait le tri.

➤ **Coopération par appel d'offre**

Dans le cas du mode d'attribution par appel d'offres, un agent « A » décompose un problème en sous-problèmes dont il diffuse la liste des sous-tâches à réaliser aux différents agents. Chaque agent « X_i » parmi les agents ayant reçu l'annonce fait une offre de service à l'agent superviseur « A » qui choisit parmi ces agents celui qui lui paraît le plus approprié et lui confie la sous tâche. Le système fonctionne ensuite en mode « Commande ». Dans ce cas, il y a quatre types de messages :

- ✿ Les messages d'annonce de tâches.
- ✿ Les messages d'offre de services.
- ✿ Les messages d'attribution de tâches.
- ✿ Les messages des résultats d'exécution des sous tâches.

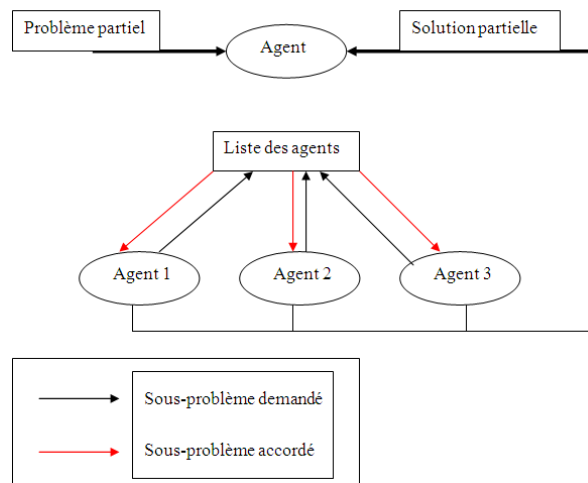


Figure II.16 : Mode de coopération par appel d'offre

II.7 Architectures d'Agents

A un niveau conceptuel, il est possible de définir et de classer la structure des agents selon des architectures représentatives et des caractéristiques pour des agents cognitifs et réactifs.

II.7.1 Structures des agents

II.7.1.1 La rationalité des agents

La vision la plus simple que l'on peut utiliser pour décrire un agent est celle d'une entité qui perçoit son environnement par des détecteurs, et agit sur l'environnement à travers des effecteurs. On aimerait qu'un agent intelligent exécute des tâches pour nous et qu'en même temps, il ait un comportement rationnel. Un agent rationnel est un agent qui agit d'une manière lui permettant d'obtenir le plus de succès possible dans la réalisation des tâches que l'on lui a assignées (Russel & al, 2002). Les facteurs qui influencent la rationalité d'un agent sont :

- ✱ la mesure de performance pour définir le succès de l'agent ;
- ✱ la séquence de perception ;
- ✱ les connaissances sur l'environnement ;
- ✱ et les actions que l'agent peut exécuter.

II.7.1.2 Structure conceptuelle des agents

Un agent est situé dans un environnement. Pour modéliser la structure de l'agent il faut avoir un modèle de l'environnement. L'environnement peut être vu comme étant dans un état e parmi un ensemble d'états $E = \{e_1, \dots, e, \dots\}$. L'environnement peut changer son état soit d'une manière spontanée soit comme résultat des actions de l'agent. L'évolution de l'environnement se modélise différemment selon les caractéristiques que l'on prend en compte, et les simplifications que l'on s'autorise. Les principales distinctions à faire sur les types d'environnements sont :

- a) Environnement accessible ou inaccessible : Un environnement est accessible à l'agent si l'agent peut percevoir entièrement l'état de l'environnement ou, au moins, tous les traits de l'environnement qui sont significatifs du point de vue des actions de l'agent. Sinon, l'environnement est inaccessible.
- b) Environnement déterministe ou non déterministe : Si l'état suivant de l'environnement est déterminé d'une manière unique par l'état courant et l'action de l'agent, alors l'environnement est déterministe. Si le résultat est incertain, notamment si, par suite d'une action de l'agent, l'environnement peut évoluer de différentes manières, alors on est dans le cas non déterministe.
- c) Environnement statique ou dynamique : Si l'environnement ne peut pas changer d'état sans l'intervention de l'agent, on est dans le cas statique. L'environnement est dynamique si son état peut se modifier sans l'action de l'agent dans l'intervalle de temps entre deux perceptions de l'agent.
- d) Environnement discret ou continu : Si tout passage d'un état de l'environnement à un autre nécessite le passage par une séquence d'états

intermédiaires, alors on a un environnement continu ; sinon, l'environnement est discret.

Les caractéristiques de l'environnement influencent la façon dont on conçoit un agent car il faut tenir compte de l'évolution de l'environnement, de la capacité de l'agent de saisir cette évolution et de sa capacité à décider en conséquence. Par exemple, si on a plusieurs agents qui agissent dans un même environnement, chaque agent va percevoir l'environnement comme dynamique et non déterministe, car l'état de l'environnement changera en raison des actions des autres agents, et une même action exécutée dans un certain état aura des résultats différents en fonction des actions de ces autres agents.

II.7.1.3 Modélisation des agents

Nous commençons par définir une structure simple (Bucharest, 2002), puis celle-ci sera développée vers des structures agents plus élaborées. Au niveau abstrait, on peut identifier quatre types d'agents : agents réactifs, agents avec états, agents avec buts et agents avec utilité.

➤ *Agents réactifs*

La structure d'un agent réactif, avec ses composantes et fonctions principales est représentée dans la figure II.17. Il faut noter qu'au niveau très général, cette structure correspond aussi aux autres types d'agents intelligents, sauf que chaque composante est plus développée et a sa propre structure.

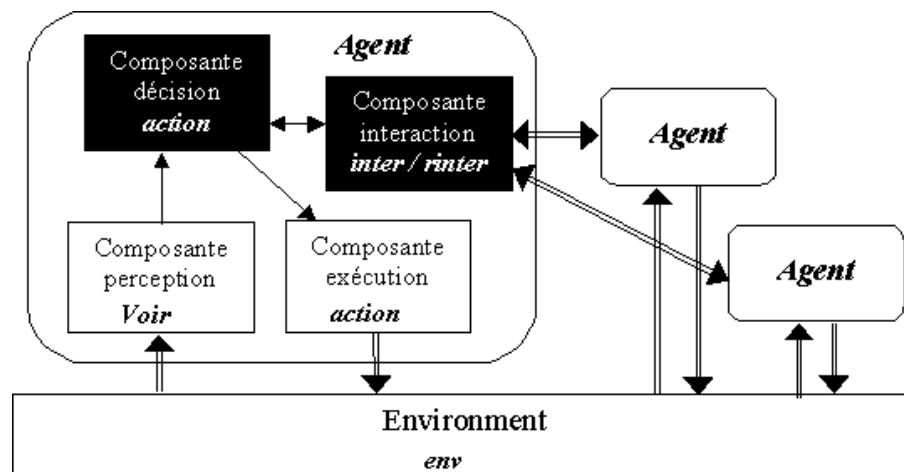


Figure II.17 : Structure générale d'un agent (Bucharest, 2002).

➤ *Agents avec états*

Les agents avec états, comme leur nom l'indique, maintiennent un état interne qui mémorise la séquence des perceptions de l'agent et, dans certains cas, les actions effectuées par l'agent. L'agent a besoin de maintenir cet état interne pour faire la distinction entre des états de l'environnement qui lui semblent identiques selon la perception qu'il en a, mais qui sont néanmoins différents.

➤ *Agents avec buts*

Connaître l'état de l'environnement et la séquence des perceptions mémorisées dans l'état interne, n'est pas toujours suffisant pour décider quelle est la meilleure action à effectuer à un moment donné. D'où, il faut associer une mesure de performance à l'agent. Pour fonctionner d'une manière efficace, l'agent doit essayer d'obtenir le maximum de performance et il doit donc choisir ses actions en conséquence.

➤ *Agents avec utilité*

La mesure de performance d'un agent peut être définie d'une manière plus fine en associant à chaque état de l'ensemble E une valeur réelle « d'utilité » qui mesure la désirabilité de cet état pour l'agent.

II.7.2 Type d'architectures des agents intelligents

II.7.2.1 Architecture réactive

Les architectures réactives représentent le fonctionnement de l'agent au moyen de composantes avec une structure de contrôle simple, et sans représentation évoluée des connaissances de l'agent. L'intelligence de l'agent est vue comme étant le résultat des interactions entre ces composantes et l'environnement. Cela veut dire qu'une telle architecture peut résoudre des problèmes complexes, qui normalement demandent un comportement intelligent, sans traiter l'intelligence du point de vue classique de l'intelligence artificielle. On dit que l'intelligence émerge de l'interaction entre des composantes simples, et entre les agents réactifs et l'environnement (Florea, 2001).

L'architecture réactive la plus connue et la plus influente est celle proposée par Rodney Brooks : elle s'appelle architecture de subsumption, en anglais "subsumption architecture". Une architecture de subsumption comporte plusieurs modules, chaque module étant responsable de la réalisation d'une tâche simple (figure II.18).

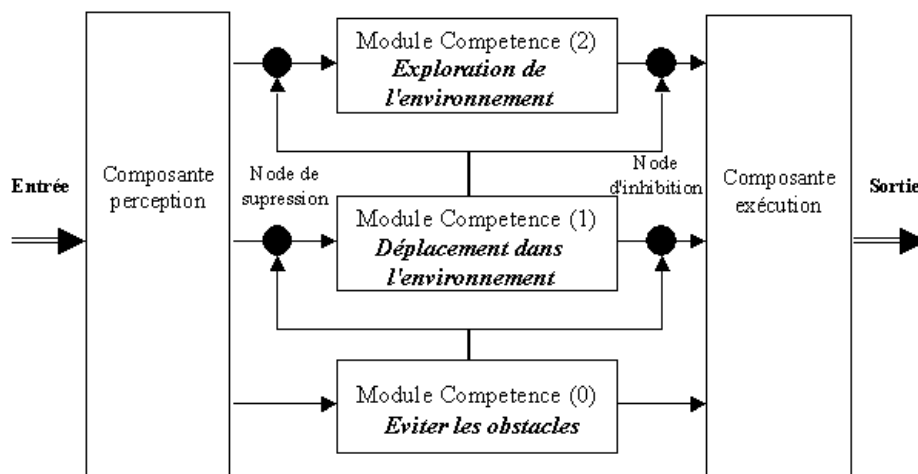


Figure II.18 : Composants d'une architecture réactive (Florea, 2001).

II.7.2.2 Architecture hybride

Une architecture hybride d'un agent intelligent est une architecture composée d'un ensemble de modules organisés dans une hiérarchie, chaque module étant soit une composante cognitive avec représentation symbolique des connaissances et capacités de raisonnement, soit une composante réactive. De cette manière, on combine le comportement proactif de l'agent, dirigé par les buts, avec un comportement réactif aux changements de l'environnement. En plus, on espère obtenir simultanément les avantages des architectures cognitives et réactives, tout en éliminant leurs limitations (Florea, 2001).

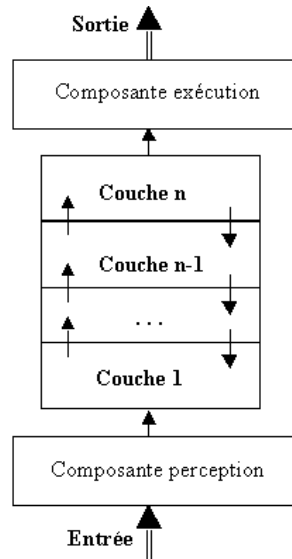


Figure II.19 : Architecture hybride en couches verticales

II.7.2.3 Architecture BDI

Une architecture BDI est conçue en partant du modèle "**Croyance-Désir-Intention**", en anglais "Belief-Desire-Intention", de la rationalité d'un agent intelligent. Dans ce qui suit, présentons d'abord d'une manière informelle et intuitive, la signification de ces trois éléments dans un modèle BDI. Le modèle repose sur une théorie de logique formelle (logique du 1^{er} ordre).

➤ **Le B = Belief = Croyance**

Les croyances d'un agent sont les informations que l'agent possède sur l'environnement et sur d'autres agents qui existent dans le même environnement. Les croyances peuvent être incorrectes, incomplètes ou incertaines et, à cause de cela, elles sont différentes des connaissances de l'agent, qui sont des informations toujours vraies. Les croyances peuvent changer au fur et à mesure que l'agent, par sa capacité de perception ou par l'interaction avec d'autres agents, recueille plus d'information.

➤ **Le D = Desire = Désir**

Les désirs d'un agent représentent les états de l'environnement, et parfois de lui-même, que l'agent aimerait voir réalisés. Un agent peut avoir des désirs contradictoires ; dans ce cas, il doit choisir parmi ses désirs un sous-ensemble qui soit consistant. Ce sous-ensemble consistant de ses désirs est parfois identifié avec les buts de l'agent.

➤ **Le I = Intention = Intention**

Les intentions d'un agent sont les désirs que l'agent a décidé d'accomplir ou les actions qu'il a décidé de faire pour accomplir ses désirs. Même si tous les désirs d'un agent sont consistants, l'agent peut ne pas être capable d'accomplir tous ses désirs à la fois.

L'exemple suivant va apporter plus de clarification à ce modèle :

L'agent Pierre a la croyance que, si quelqu'un passe son temps à étudier, cette personne peut faire une thèse de doctorat. En plus, Pierre a le désir de voyager beaucoup, de faire une thèse de doctorat et d'obtenir un poste d'assistant à l'université. Le désir de voyager beaucoup n'est pas consistant avec les deux

autres et Pierre, après réflexion, décide de choisir, parmi ces désirs inconsistants, les deux derniers. Comme il se rend compte qu'il ne peut pas réaliser ses deux désirs à la fois, il décide de faire d'abord une thèse de doctorat. En ce moment Pierre a l'intention de faire une thèse et, normalement, il va utiliser tous ses moyens pour y parvenir. Il serait irrationnel de la part de Pierre, une fois sa décision prise, d'utiliser son temps et son énergie, notamment ses moyens, pour voyager autour du monde. En fixant ces intentions, Pierre a moins de choix à considérer car il a renoncé à faire le tour des agences de voyage pour trouver l'offre de voyage qui le satisferait au mieux.

C'est cette idée même qui est au cœur de la théorie BDI de l'action rationnelle, proposée pour la première fois par Michael Bratman. C'est une théorie du raisonnement pratique qui essaie de simuler comment les gens raisonnent dans la vie de tous les jours, en décidant, à chaque moment, ce qu'ils ont à faire. En développant sa théorie, Bratman montre que les intentions jouent un rôle fondamental dans le raisonnement pratique, car elles limitent les choix possibles qu'un humain (ou un agent artificiel) peut faire à un certain moment.

➤ **Les composantes BDI**

Une architecture BDI est alors un bon candidat pour modéliser le comportement d'un agent intelligent car :

- * elle s'appuie sur une théorie connue et appréciée de l'action rationnelle des humains ;
- * la théorie a été formalisée dans une logique symbolique formelle et rigoureuse ;
- * elle a été implémentée et utilisée avec succès dans beaucoup d'applications.

La figure suivante présente les composantes principales d'une architecture BDI.

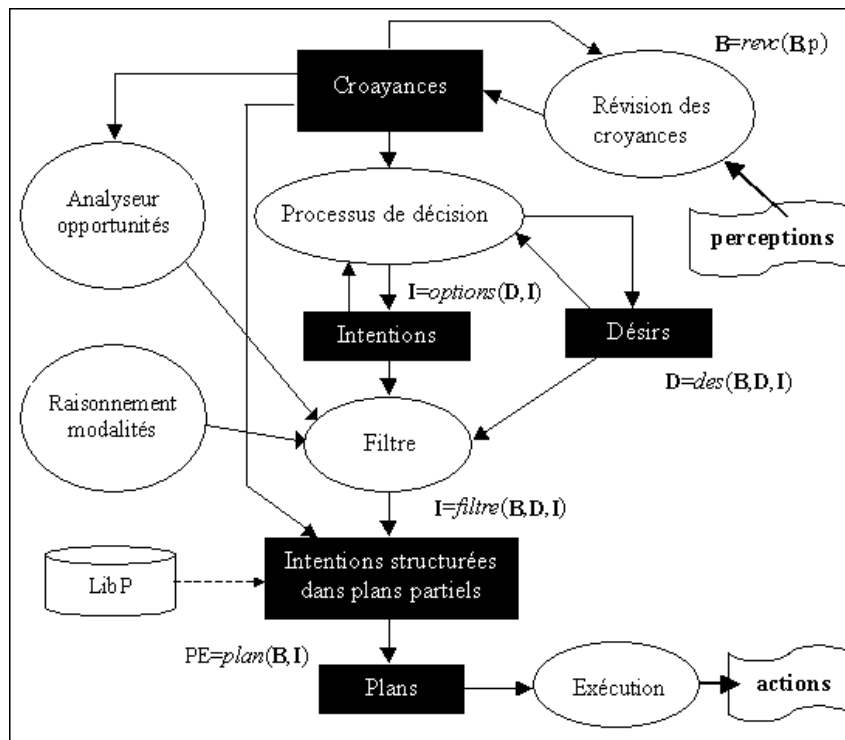


Figure II.20 : Architecture BDI d'un agent

Dans la figure II.20, les carrés représentent des structures de connaissances ou de données alors que les ovales représentent des composantes de contrôle et d'exécution. L'agent a une représentation explicite de ses croyances (B), désirs (D) et intentions (I).

Les ensembles B, D et I peuvent être représentés au moyen de divers modèles de représentation de connaissances :

- la logique des prédicats du premier ordre
- une logique d'ordre supérieur
- le modèle des règles de production
- ou bien comme de simples structures de données.

Les composantes de contrôle et d'exécution sont les suivantes :

A. Révision des Croyances : C'est ce module qui est responsable de la révision des croyances de l'agent. Il calcule ses nouvelles croyances à partir de celles courantes et de ses nouvelles perceptions de son environnement.

B. Processus de décision : C'est le module qui s'occupe de la représentation du processus de décision de l'agent en prenant compte de ses désirs et ses intentions pour générer de nouvelles options pertinentes (vis-à-vis des désirs et intentions courantes). Il peut aussi changer les désirs d'un agent si ses croyances ou intentions changent, pour maintenir la consistance des désirs de l'agent, il faudra supposer que l'agent a toujours des désirs consistants.

C. Analyseur opportunités : En conséquence de ce qu'il perçoit de son environnement et de sa révision des croyances, l'agent peut détecter des nouvelles opportunités qui favorisent la réalisation de ses intentions ou qui peuvent même empêcher cette réalisation.

D. Raisonnement modalités : Cette composante a la responsabilité d'effectuer le raisonnement orienté action et la modalité de réalisation des plans.

E. Filtre : C'est le module le plus important car il décide des intentions à poursuivre, il a la responsabilité de bâtir des plans partiels pour réaliser les intentions de l'agent, tout en tenant compte des nouvelles opportunités. C'est un processus continu, il doit éliminer (rendre inactives) les intentions devenues irréalistes ou incohérentes (entre elles).

F. Lib : C'est une bibliothèque de plans. Cette composante peut être présente ou non dans l'architecture. Si elle existe, le processus de planification de l'agent est simplifié car il peut retrouver des plans adéquats à une certaine situation en parcourant cette bibliothèque de plans.

G. Exécution : Une fois que l'on dispose d'un plan exécutable, ce module va exécuter les actions du plan, l'une après l'autre, dans l'environnement. La représentation des actions consiste aussi à représenter les effets de ces actions sur l'environnement.

Par exemple, si dans un état de l'environnement, l'agent déplace un chariot de la pièce A à la pièce B, la représentation de cette action sera "déplacer-chariot" de A à B et l'effet sur l'environnement contiendra notamment le fait que le chariot n'est plus dans la pièce A et qu'il est dans la pièce B.

Le résultat d'une action sera celui envisagé si l'environnement est déterministe. Les plans conçus par l'agent sont toujours des structures de connaissances ou structures de

données. *Un plan* est une séquence d'actions à exécuter dans le temps. *Un plan partiel* est un plan dans lequel tous les détails de planification n'ont pas été spécifiés.

Par exemple, on sait que pour partir en voyage, il faut faire ses valises et téléphoner à un ami pour dire au revoir, mais on n'indique ni dans quel ordre il faut faire ces deux actions, ni le nombre de valises à emporter. Ceci est un plan partiel. Une fois décidé que l'on préparerait d'abord deux valises et que l'on appellerait ensuite son ami, on a raffiné le plan partiel et obtenu ainsi un plan complètement spécifié et exécutable.

➤ **Le contrôle BDI selon les stratégies d'obligation**

Pour apporter plus de précision sur les mécanismes de l'architecture BDI, nous allons décrire le processus de contrôle d'un agent BDI par l'algorithme proposé par Michael Wooldridge.

Soient B_0 , D_0 et I_0 les croyances, désirs et intentions initiales de l'agent.

```
1 B = B0
2 D = D0
3 I = I0
4 Répéter
  4.1 obtenir nouvelles perceptions p.
  4.2 B = revc (B, p) //Révision des croyances
  4.3 I = options (D, I)
  4.4 D = des (B, D, I)
  4.5 I = filtre (B, D, I)
  4.6 PE = plan (B, I)
  4.7 exécuter (PE)
  Jusqu'à ce que l'agent soit arrêté
Fin
```

Listing II.1 : Algorithme de contrôle d'un agent BDI

Le problème qui se pose dans cet algorithme est que : si une intention a été choisie par la fonction *filtre*, on dit que l'agent s'est obligé à réaliser cette intention. L'obligation d'un agent envers une intention, en anglais appelé "commitment", a une grande importance dans le modèle BDI d'un agent situé dans un environnement non déterministe. On peut donc supposer que l'agent va maintenir, pour un certain temps, ses intentions choisies au pas 4.5.

Mais si, au moment où l'agent en arrive à exécuter son intention (pas 4.7), les conditions de l'environnement ont changé par rapport aux perceptions effectuées au pas 4.1 et qu'en conséquence, d'autres intentions deviennent plus opportunes à réaliser, on se pose la question suivante "Combien de temps l'agent restera-t-il obligé par une intention ?". Donnons un exemple :

Si Pierre a décidé que son intention est de faire une thèse de doctorat, il s'est alors obligé à cette intention. Pierre commence à travailler et faire sa recherche. Entre temps, il découvre qu'il n'aime pas trop travailler et qu'il préfère passer son temps avec ses amis dans les bistros. Pierre se rend compte que son intention est impossible à accomplir, ce qui le conduit à abandonner son intention de faire une thèse. Dans un autre cas, comme Pierre veut aussi gagner un peu d'argent, il prend un job dans une compagnie. Comme il est très éloquent et persuasif, on lui offre un poste dans la compagnie qui lui demande de voyager beaucoup pour

représenter les intérêts de la compagnie. Pierre constate alors qu'une autre de ses intentions devient opportune, il abandonne son engagement de faire une thèse et accepte le poste qu'on lui a offert.

La réponse à cette question est donnée par la stratégie d'obligation envers les intentions choisies par le concepteur pour les agents BDI. On peut identifier trois principales stratégies d'obligation :

- a. **Obligation aveugle (ou obligation fanatique) « blind commitment »** : Un agent suivant cette stratégie va maintenir ses intentions jusqu'à ce qu'elles soient réalisées, plus précisément jusqu'à ce qu'il croie qu'elles sont réalisées. Tenant compte de ce qui a déjà été dit, cette stratégie n'est pas la meilleure si l'environnement change entre le moment où l'agent a choisi (filtré) ses intentions, et le moment où ces intentions doivent être accomplies.
- b. **Obligation limitée « single minded commitment »** : Cette stratégie dit que l'agent va maintenir ses intentions, ou bien jusqu'à ce qu'elles soient réalisées, ou bien jusqu'à ce qu'il croie qu'elles ne sont plus réalisables.
- c. **Obligation ouverte « open minded commitment »** : Un agent ayant une stratégie d'obligation ouverte maintient ses intentions tant que ces intentions sont aussi ses désirs. Cela implique aussi que, une fois que l'agent a conclu que ses intentions ne sont plus réalisables, il ne les considère plus parmi ses désirs.

L'algorithme de contrôle d'un agent BDI décrit plus haut correspond à un agent avec une stratégie d'obligation aveugle envers ses intentions. Dans les deux autres cas, l'agent doit vérifier ses intentions après l'exécution de chaque action du plan qu'il a conçu au pas 4.6.

II.7.2.4 Architecture JAM (Java Agent Model)

Le modèle JAM a été élaboré par Huber et ses collègues (Huber, 2001) (Ricordel, 2001) (Mbala, 2003). JAM est une architecture hybride qui se fonde sur les idées et les théories du système de raisonnement procédural (PRS pour *Procedural Reasoning System*), de la sémantique des circuits structurés (SCS pour *Structured Circuit Semantic*) et du "Act plan interlingua". JAM offre des représentations procédurales riches et étendues, un raisonnement rationnel et de niveau méta prenant en compte plusieurs objectifs en même temps, un comportement basé à la fois sur les objectifs poursuivis et la réaction aux événements. L'architecture JAM permet aussi d'implémenter des capacités de mobilité ; elle est construite autour d'un noyau BDI et se base sur les implémentations PRS de l'Université de Michigan et de l'entreprise "SRI International", connues respectivement sous les sigles "UMPRS" et "PRS-CL".

Chaque agent JAM est composé de cinq composants primaires représentés à la figure II.21 : un modèle du monde, une bibliothèque de plans, un interpréteur, une structure intentionnelle et un observateur. Le modèle du monde est une base de données représentant les croyances de l'agent. La bibliothèque de plans est un ensemble de plans que l'agent utilise pour atteindre ses objectifs. La structure intentionnelle est un modèle interne des objectifs actuels de l'agent ; elle garde les traces de l'engagement et de la progression de l'agent dans la réalisation de ses objectifs. L'observateur est une procédure déclarative spécifiée par l'utilisateur qui permet à l'agent de relier les étapes d'un plan afin de remplir des fonctionnalités simples et périodiques (par exemple, transférer les messages entrants au modèle du monde). L'interpréteur est le cerveau de l'agent, il permet à l'agent de raisonner sur ce qu'il doit faire, quand et comment.

L'agent JAM fonctionne en utilisant les sémantiques d'exécution et de comportement, c'est une combinaison de UMPRS et de SCS : les changements dans le modèle du monde et l'apparition de nouveaux objectifs déclenchent un processus de raisonnement afin de rechercher les plans à appliquer à la nouvelle situation ; l'interpréteur sélectionne un plan de la liste des plans applicables en faisant un raisonnement de niveau méta ou d'utilité maximale.

➤ L'interpréteur

L'interpréteur sélectionne et exécute les plans sur la base des intentions, plans, objectifs et croyances de l'agent. Il est associé à la structure intentionnelle ; c'est une pile d'exécution des objectifs associés ou non à des plans instanciés. L'agent contrôle les plans applicables à un objectif. Une variable d'utilité est associée à chaque plan. S'il n'est pas possible de sélectionner un plan unique à partir du critère d'utilité, l'interpréteur met en œuvre un raisonnement de niveau méta pour sélectionner le plan à exécuter. L'interpréteur est une procédure écrite en Java, comprenant une boucle qui s'exécute jusqu'à ce que l'agent réalise tous ses objectifs.

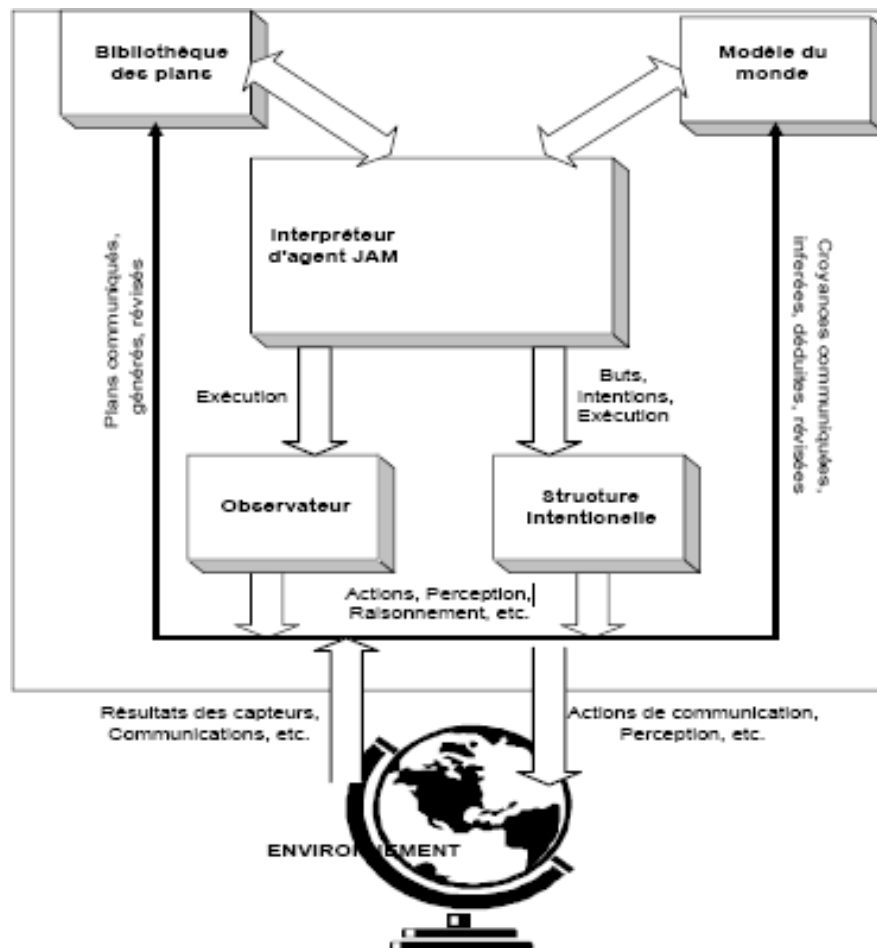


Figure II.21 : L'architecture JAM selon (Mbala, 2003)

➤ Les buts

Le comportement d'un agent JAM est motivé par les buts. On a trois types de buts, chacun ayant une sémantique différente : ACHIEVE (*réaliser*), PERFORM (*se comporter*), MAINTAIN (*maintenir*). Le but ACHIEVE spécifie que l'agent désire atteindre un état, il s'agit typiquement du but associé aux architectures BDI et aux

systèmes générant des plans. Le but PERFORM spécifie le comportement que l'agent désire avoir ; il ne s'agit pas de réaliser un but mais d'exhiber un comportement particulier. Le but MAINTAIN indique qu'un but particulier doit être réalisé de nouveau à chaque fois qu'il viendrait à ne plus l'être, c'est un but homéostatique qui ne doit pas être retiré de la liste des buts, à moins que ce soit du fait exprès de l'agent. Les buts sont assignés aux agents, en précisant le type du but, son nom, des paramètres, un critère d'utilité et une expression (optionnels). L'utilité du but est associée à l'utilité du plan pour déterminer l'utilité totale de l'intention. L'agent essaye toujours de réaliser le but pour lequel le critère d'utilité est le plus élevé, ceci permet d'obtenir des agents rationnels. Un ou plusieurs buts initiaux sont assignés à l'agent au moment de son invocation.

La liste des objectifs peut être augmentée durant son exécution à travers les interactions avec d'autres agents, ou des raisonnements internes de l'agent ou encore par d'autres moyens. Les buts initiaux de l'agent sont dits persistants en ce sens qu'ils sont poursuivis jusqu'à ce qu'ils soient réalisés grâce à des plans bien appliqués ou par d'autres agents ou jusqu'à ce qu'ils soient retirés du plan par exemple quand l'agent estime qu'ils ne peuvent plus être atteints. Si le plan associé à la réalisation des buts initiaux échoue, l'agent renonce à son engagement à réaliser ce but en retirant son intention ; le but reste toutefois dans la structure intentionnelle pour des tentatives ultérieures de réalisation de l'objectif. Par contre si un plan associé à un sous-objectif échoue, l'interpréteur considère que l'action associée à ce sous-but a échoué (en supposant qu'il existe un autre plan pour réaliser ce sous-but).

➤ Les plans

Un plan JAM spécifie une procédure (écrite en Java) pour réaliser un but, réagir à un événement ou avoir un comportement donné. Les agents JAM peuvent dès lors être guidés par des buts (rationnels) ou simplement réagir à des événements (réactifs). Un ou plusieurs plans sont affectés à l'agent au moment de son invocation ; des plans peuvent être ajoutés durant l'exécution soit à travers les interactions avec les autres agents, soit générés à partir d'un processus de raisonnement interne de l'agent, soit par d'autres moyens. L'applicabilité d'un plan est limitée soit à un objectif, soit à la réaction à une combinaison donnée d'événements.

Chaque plan est contraint par : des pré-conditions, conditions devant apparaître avant le début de son exécution ; et un contexte, conditions que l'on doit avoir avant et pendant l'exécution du plan. Chaque plan doit comporter une procédure de calcul des critères d'utilité implicite et explicite afin de permettre la sélection d'un plan à partir d'un ensemble donné de plans. JAM offre plusieurs structures itératives comme DO...WHILE, WHILE ; des structures de branchement conditionnel comme OR, AND, DO_ALL, WHEN ; des structures d'initialisation comme ASSIGN. Les croyances d'un agent JAM peuvent être modifiées ou contrôlées en utilisant les primitives ASSERT, FACT, RETRACT, RETRIEVE, UPDATE.

JAM permet des exécutions parallèles avec PARALLEL et une primitive de synchronisation : WAIT.

Le concepteur peut augmenter les fonctionnalités offertes par JAM en écrivant des fonctions en Java, plusieurs méthodes d'accès au code Java sont offertes par JAM. C'est à travers ces mécanismes que JAM permet de construire des applications spécifiques (par exemple l'interfaçage avec des bases de données) et d'implémenter des

caractéristiques de sociabilité (comme la communication ou la collaboration entre agents).

➤ **Le modèle du monde**

Le modèle du monde de JAM contient les données représentant l'état courant du monde tel que perçu par l'agent. Les informations contenues dans cette base de données concernent : les variables d'état, les informations sensorielles, les conclusions tirées des déductions et des inférences, la modélisation des autres agents, etc. Une grande flexibilité est laissée au programmeur pour définir l'ordre, le type et la sémantique des arguments utilisés.

➤ **L'observateur**

L'observateur est une procédure déclarative optionnelle que l'interpréteur JAM exécute après chaque action contenue dans le plan. L'observateur est une architecture-crochet que le programmeur peut utiliser pour implémenter plus facilement des caractéristiques qui ne sont pas prises en compte dans les plans et les buts normaux de JAM. Le programmeur peut utiliser l'observateur pour vérifier par exemple le tampon des messages entrants. Cette procédure est utilisée pour observer les événements asynchrones.

L'architecture des systèmes multi-agents est un aspect très important dans l'étude des SMA, il permet de prévoir les composants essentiels de chaque agent d'après le type de l'architecture choisie. Cependant les architectures BDI sont l'exemple le plus représentatif d'architecture permettant de bâtir un agent cognitif. D'autres architectures que BDI pour les agents cognitifs ont été conçues et utilisées dans diverses applications mais, jusqu'à présent, l'architecture BDI est celle qui est préférée dans la communauté de recherche agents intelligents et systèmes multi-agents. Pour les raisons si dessus et après l'étude des différentes architectures, notre choix s'est porté sur l'architecture BDI pour implémenter nos agents.

II.8 Méthodologies de développement orienté agents

La littérature relative au domaine des agents abonde de démarches et de méthodes permettant la conception et le développement de systèmes multi-agents. Selon la nature et les spécificités du problème à résoudre par le SMA, l'approche méthodologique choisie permettra de couvrir les besoins conceptuels et même de s'étaler pour certaines méthodes jusqu'à la phase de développement.

Beaucoup de méthodes sont indépendantes des outils de développement, cependant certaines d'entre elles sont supportées par des outils permettant la génération de code. Une des phases de notre travail de recherche a été de passer en revue les méthodologies les plus intéressantes afin d'en choisir une qui convienne à notre objectif. Un intérêt particulier a été porté aux méthodologies basées sur l'orientation objet étant donné que notre travail de conception repose sur l'UML et afin de rester cohérent avec la démarche globale adoptée pour l'infrastructure qui sera proposée dans les prochains chapitres.

II.8.1 Ingénierie logicielle orienté agent

Construire un logiciel de qualité supérieure pour les applications du monde réel est une tâche difficile à cause du nombre important et de la flexibilité nécessaire des

composantes mais aussi à cause de la complexité des interconnexions requises. Cette affirmation est vraie quels que soient le modèle et les techniques appliquées.

Le rôle de l'ingénierie logicielle est justement celui de fournir des modèles et des techniques qui puissent faciliter le contrôle de cette complexité. A cette fin, plusieurs paradigmes d'ingénierie logicielle ont été proposés, par exemple l'orientation objet, les modèles de conception et diverses architectures de logiciel.

Chaque paradigme proposé a essayé soit de faire en sorte que le processus de conception soit plus facile, soit d'étendre la complexité des applications qui peuvent être construites en utilisant le paradigme.

Même si ces paradigmes représentent une étape nouvelle dans la construction du logiciel, quand il s'agit du développement des systèmes distribués complexes ils échouent principalement pour deux raisons :

- ▼ Les interactions entre les diverses entités sont définies d'une manière trop rigide
- ▼ Il n'y a pas de mécanismes assez complexes pour représenter la structure organisationnelle du système.

Le paradigme des agents et systèmes multi-agents peut être une bonne réponse à ces problèmes, parce que les approches orientées agent accroissent significativement notre capacité de modéliser, concevoir et construire des systèmes distribués complexes (Wooldridge & al, 1995). En plus, les approches orientées agent sont une étape naturelle dans l'évolution logique d'une gamme d'approches contemporaines en ingénierie logicielle.

II.8.1.1 L'approche orientée agent en ingénierie logicielle

Dans son ouvrage "*Analyse et conception orientées objet et applications*", Booch énonce les caractéristiques nécessaires d'un paradigme d'ingénierie logicielle permettant de construire des systèmes complexes :

- a) **la décomposition** : la capacité de diviser le problème en sous-problèmes plus simples, les sous-problèmes étant plus maniables et pouvant être traités d'une façon relativement autonome ;
- b) **L'abstraction** : la capacité de définir un modèle simplifié du système qui met en évidence une partie de ses détails et propriétés, tout en éliminant les autres; donc l'attention peut se concentrer sur les aspects saillants du problème et se dispenser des détails moins importants;
- c) **L'organisation** : le procédé consistant à identifier et à gérer les rapports entre les divers sous-problèmes (composantes) à résoudre. La possibilité de spécifier et d'établir des relations organisationnelles entre composantes aide les concepteurs à faire face à la complexité parce qu'en premier lieu on peut grouper diverses composantes dans une composante plus complexe qui peut être alors traitée comme une unité unique d'analyse et, deuxièmement, parce qu'on a la possibilité de décrire des relations complexes entre ces unités, par exemple coopération, fonctionnalités conjointes, etc. Ces trois caractéristiques peuvent être bien retrouvées dans le modèle des systèmes multi-agents.

II.8.1.2 Méthodologies orientées agent

Il y a deux grands groupes de méthodologies pour le développement du logiciel orienté agent. Le premier groupe étend ou adapte les méthodologies orientées objet pour prendre en compte les caractéristiques des agents, alors que le deuxième groupe part des méthodologies de l'ingénierie de connaissances, en outre, il y a ce qu'on appelle les méthodes qui ont été conçues pour un contexte particulier qui utilisent les deux approches.

1. Méthodologies orientées agent basées sur l'ingénierie des connaissances

Plusieurs méthodologies pour le développement des systèmes multi-agents ont été proposées en partant des celles qui ont été dédiées à la modélisation des systèmes à bases de connaissances. Un bon nombre de ces méthodologies repose sur la méthodologie "CommonKADS" qui est une norme européenne pour la modélisation de connaissances (Sabas, 2001).

2. Les méthodes qui ont été conçues pour un contexte particulier

Conçues pour des contextes particuliers, ce type de méthodes met l'accent sur certains aspects plus que d'autres dans le processus de conception/développement afin de répondre aux besoins fixés en priorité et en fonction du domaine abordé. Par exemple, la méthodologie "Cassiopea" a une vision par le bas, contrairement à la méthodologie "Gaia" qui fait partie des méthodes basées sur l'orienté objet. Autrement dit, dans "Cassiopea" on commence par spécifier les comportements nécessaires pour réaliser les tâches. Ceci correspond à une vision comportementaliste des agents mis en avant par Brooks (1999) (Gauthier, 2004).

Une autre méthode, "CIAD" (Cooperative Information Agent Design) est destinée au domaine du commerce (Sabas, 2001), elle se base sur l'usage de trois modèles : le modèle d'autorisation, le modèle de communication et le modèle de l'univers de discours.

3. Méthodologies orientées agent basées sur l'orientation objet

Beaucoup de méthodologies orientées objet sont utilisées dans l'industrie avec succès, tel que par exemple (Muller, 1997) :

- ✱ La Technique de Modélisation à base d'Objets (OMT- Object Modelling Technique - Rumbaugh, 1991).
- ✱ L'Ingénierie du Logiciel Orientée Objet (OOSE- Object Oriented Software Engineering – Jacobson qui développa les uses cases, 1992).
- ✱ La Conception Orientée Objet avec les méthodes de Booch, 1994).
- ✱ UML (Unified Modelling Language - Rational Software Corporation, 1997).

Les extensions et les modifications des méthodologies orientées objet doivent prendre en compte les différences qui existent entre les objets et les agents. Dans le cadre de nos travaux, nous nous sommes penchés sur un certain nombre de méthodes dont nous avons utilisé et expérimenté la méthode MaSE – qui présente l'avantage d'être supportée par un outil, AgentTool (Deloach, 2001) - et la notation AUML⁷ (Odell & al, 2000), (Bauer, 2001). Dans ce Nous nous pencherons particulièrement sur les enrichissements apportés à UML pour supporter le paradigme d'agent car la partie

⁷ <http://www.auml.org>

concernant les acteurs automatiques de l'infrastructure proposée a été modélisée avec AUML.

a) Multi-agent Systems Engineering (MaSE)

L'auteur de cette méthodologie (Deloach, 1999) (Deloach, 2003) définit un agent comme un objet actif ayant un objectif et un langage de communication. Cette méthodologie utilise les techniques de OMT⁸ ou de UML avec quelques caractéristiques de plus et quelques modifications de la sémantique du paradigme objet pour pouvoir capter les concepts d'agent et les comportements coopératifs des agents.

Elle utilise deux langages pour décrire les agents et les SMA : AgML⁹ et AgDL¹⁰. Les différentes phases de MaSE sont (Figure II.22):

- ✱ Conception du domaine (Domain Level Design) ;
- ✱ Conception d'agents (Agent Level Design) ;
- ✱ Conception de composantes (Component Design) ;
- ✱ Conception du Système (System Design).

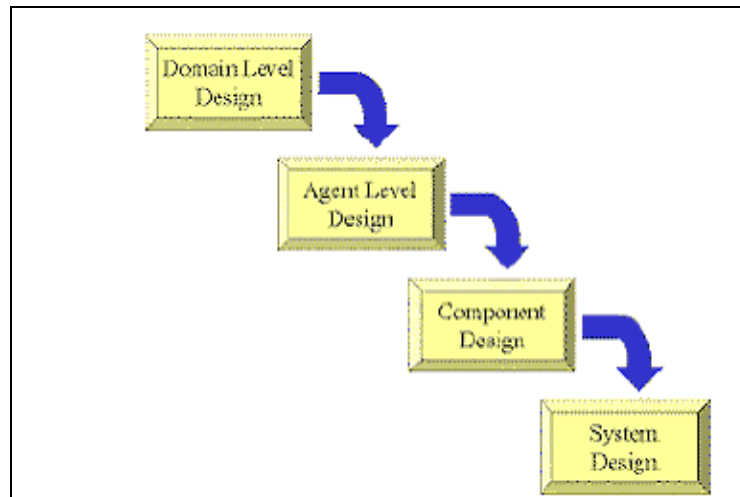


Figure II.22 : les différentes phases de MaSE (Deloach, 1999).

Cette méthodologie met l'accent sur la façon de coordonner dans les SMA le comportement de chaque agent afin d'obtenir un comportement du système tout entier. Pour cela, elle utilise un certain nombre de modèles graphiques pour décrire le type d'agents, les interfaces inter- agents, une architecture indépendante de la structure interne des agents.

⁸ : Object Modeling Technic

⁹ : Agent Modeling Language

¹⁰ : Agent Definition Language

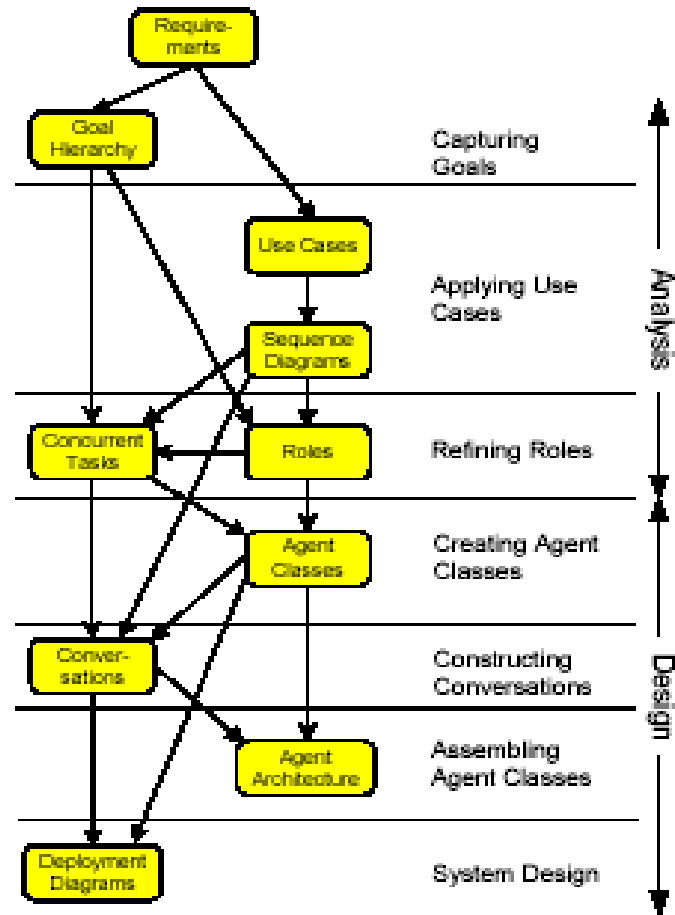


Figure II.23 : Méthodologie MaSE (Deloach, 1999)

Dans MaSE, un agent est considéré comme une abstraction de plus haut niveau qu'un objet. Alors qu'un objet peut voir ses méthodes invoquées par n'importe quel autre objet, les agents interagissent entre eux à travers des interactions (*conversations*) et prennent des initiatives pour accomplir leurs buts personnels et les objectifs du système tout entier.

b) Le langage AUML (Agent Unified Modelling Language)

Le langage de modélisation unifiée (Unified Modelling Language) est un essai d'unifier les divers paradigmes d'analyse et de conception orientée objet du logiciel et d'offrir une notation unique pour la modélisation des systèmes orientés objet (Muller, 1997).

Il faut noter qu'UML n'est pas une méthodologie, mais, comme son nom le dit, un langage pour élaborer la documentation des modèles de systèmes. Il y a aussi une méthodologie associée à UML qui s'appelle « Rational Unified Process » (Jacobson & al, 2000).

Dans ce contexte, il a été proposé d'adapter les notations d'UML pour décrire la modélisation orientée agent. Les modifications proposées à UML sont :

- ✱ Soutien pour la représentation des *threads* simultanés d'interaction (par exemple transmission de messages à plusieurs agents) permettant ainsi à UML de modéliser les protocoles d'interactions entre agents, par exemple le réseau contractuel ;

- ✿ La notion de rôle qui étend celle fournie par UML et permet de modéliser un agent qui joue plusieurs rôles.

L'initiative la plus connue pour étendre UML avec des facilités permettant de décrire les agents s'appelle AUML (Agent Unified Modelling Language) et a comme but de :

- ✿ Recommander une technologie pour l'adoption d'une sémantique, un méta-modèle et une syntaxe abstraite communes pour l'analyse et la conception des méthodologies basées sur les agents ;
- ✿ Recommander une technologie qui permet l'interopérabilité à travers le cycle de vie des produits et d'outils de travail AUML ;
- ✿ Etre en accord avec les spécifications existantes de la FIPA¹¹ et de l'OMG¹².

Il faut noter qu'actuellement, AUML est plutôt un but qu'un langage de modélisation existant. L'OMG et la FIPA appuient le développement des notations UML pour la modélisation des systèmes multi-agents et prévoient des efforts considérables dans cette direction.

Dans ce qui suit, nous tenterons de présenter uniquement les extensions apportées à UML et qui sont nécessaires à notre travail. Ceci, dans le but de mettre au clair les différents concepts qui seront utilisés plus loin dans cette thèse.

➤ **Le diagramme de Classe Agent**

Des extensions ont été apportées au schéma d'une classe UML qui classiquement comporte l'ensemble des compartiments suivants :

- **Nom** : Chaque classe doit avoir un nom différent de celui des autres classes.
- **Attribut** : C'est une propriété nommée d'une classe qui décrit un ensemble de valeurs que les instances de cette propriété peuvent prendre. Cette propriété est commune à l'ensemble des objets de cette classe.
- **Opérations (méthodes)** : Une opération est l'implémentation d'un service qui peut être demandé à tous les objets d'une même classe. Une classe peut avoir plusieurs opérations ou ne pas en avoir du tout.
- **Compartiment supplémentaire** : C'est un compartiment facultatif utilisé dans le traitement des exceptions.

Comme décrit dans les sections précédentes de ce chapitre, un agent se caractérise par son autonomie, pro et réactivité, sa capacité de communiquer avec le monde extérieur et son état interne qui est plus qu'un champ avec des types de données bien spécifiques. Tous ces concepts doivent être supportés par le diagramme de classe agents. Ceci a conduit aux modifications suivantes présentées en figure II.24 :

¹¹ : Foundation for Intelligent Physical Agents

¹² : Object Management Group

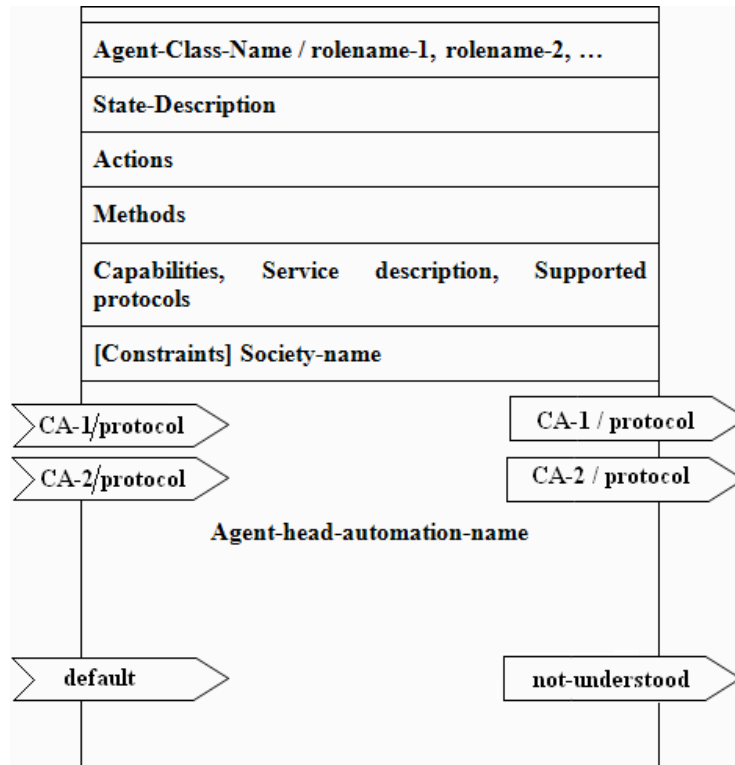
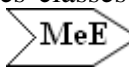
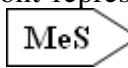


Figure II.24 : Modèle de classe Agent dans AUML¹³ (Bauer, 2001)

- **Nom de la classe / rôle1, rôle2** : Un agent d'une classe donnée peut avoir différents rôles durant son exécution.
- **Description des états** : Cette extension donne la possibilité de bien définir les agents basés sur l'architecture BDI. Ainsi, par exemple nous pouvons définir quatre instances de variables nommées : Croyance, Désir, Intention et But.
- **Actions** : Deux types d'actions peuvent être spécifiés : action proactive (avec le stéréotype "proactive") exécutée par l'agent lui-même si une pré-condition devient vraie et réactive (avec le stéréotype "réactive") résultant d'un message reçu d'un autre agent. En d'autres termes, les actions sont les plans qu'a un agent.
- **Méthodes** : Elles sont définies comme dans UML avec éventuellement des pré-conditions, des post-conditions et/ou des invariants.
- **Capacités** : Elles peuvent être définies en utilisant des spécificités informelles ou des diagrammes de classes. Par exemple, la description des services FIPA.
- **Envoi et réception de messages** : La principale interface entre les agents et leurs environnements est la réception et l'envoi de messages. Ces messages peuvent être des classes ou des objets. Les messages entrants sont représentés par le symbole  et les messages sortants par le symbole .

De plus, pour différencier les types de classes agents et les instances, les notations illustrées dans la figure II.25 ont été introduites :

¹³ CA signifie "Communication Act".

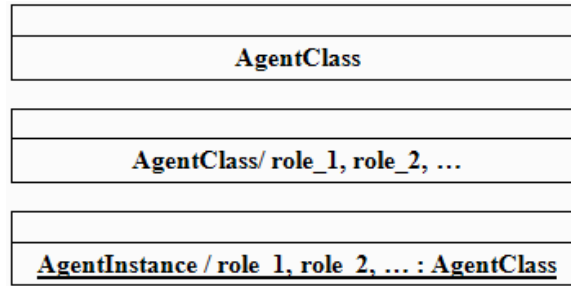


Figure II.25 : Classes et Instances d'Agents

La première notation désigne une classe d'agents ordinaires, la seconde une classe d'agents satisfaisant un rôle bien déterminé et la dernière une instance de classe agents qui satisfait un rôle qui est aussi bien déterminé.

➤ **Le diagramme de Séquence ou diagramme de protocoles (Huguet, 2002)**

La communication élémentaire entre agents se fait de la même manière que dans UML comme le montre la figure II.26 :

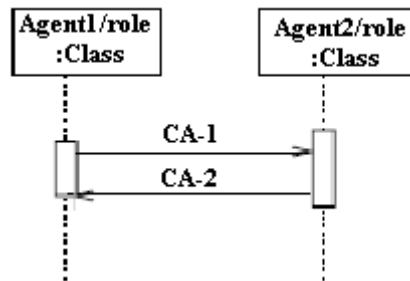


Figure II.26 : Communication élémentaire entre deux agents

Une autre extension UML a été ajoutée pour supporter les threads d'interactions concurrents ou connecteurs AUML, comme le montre la figure II.27 :

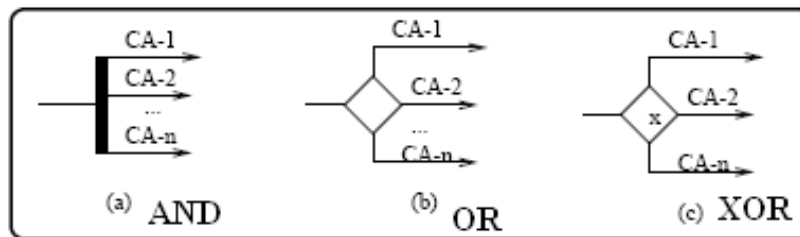


Figure II.27 : Connecteurs AUML.

- (a) Indique que tous les threads de CA-1 à CA-n sont envoyés concurremment. Elle inclut aussi (b).
- (b) Inclut une boîte de décision (decision box) qui décidera quel est le thread (un ou plus) à envoyer.
- (c) Un carré de "ou exclusif" où sera envoyé un seul thread CA.

La figure II.28 illustre les manières d'utiliser les threads d'interaction concurrents décrits en figure II.27.

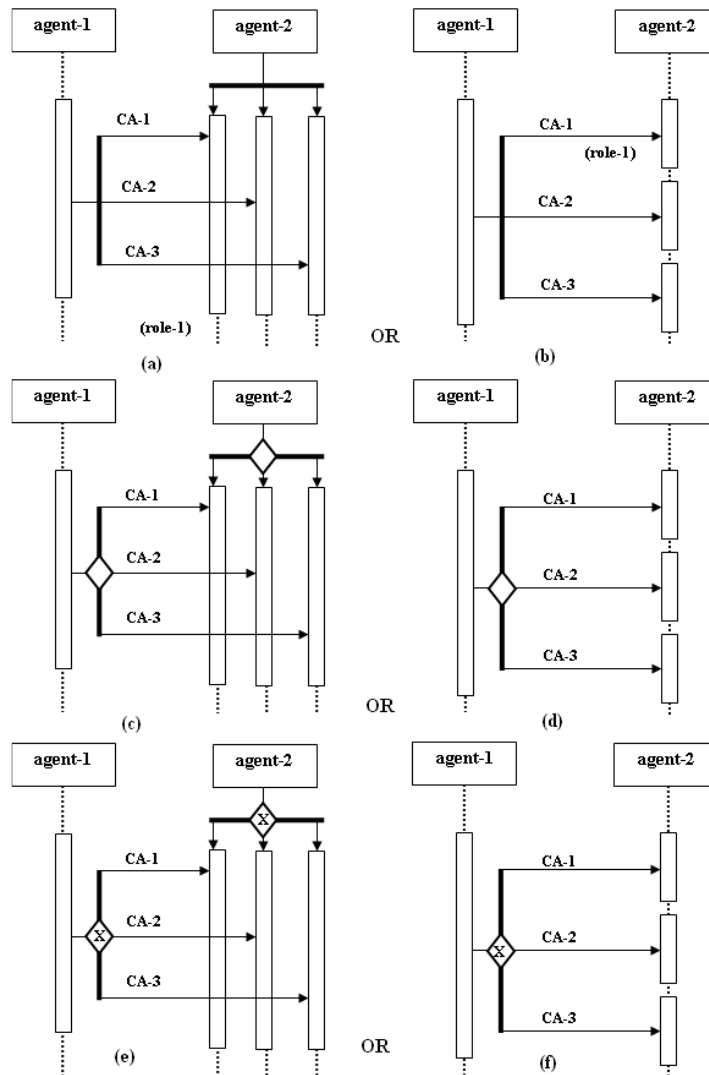


Figure II.28 : Quelques techniques pour exprimer la communication avec un agent jouant plusieurs rôles ou répondant à différents cas (Bauer, 2001).

La figure II.27 montre deux méthodes de représentation des threads concurrents envoyés de "agent-1" vers "agent-2". Les barres d'activation verticales multiples indiquent que l'agent récepteur est entrain d'exécuter les divers threads de communication de manière concurrente.

Pour chaque variante de thread utilisée, la sémantique est équivalente. Le choix est basé sur la facilité et la clarté de l'apparence visuelle.

Chaque barre d'activation peut indiquer si l'agent utilise un rôle différent ou s'il emploie simplement un processeur de threads différent pour supporter l'acte de communication "CA". Si l'agent utilise un rôle différent, la barre d'activation peut être annotée de manière particulière. Par exemple, pour les figures (II.27(a)) et (II.27(b)), "CA-n" est traité par l'agent sous le role "role-1".

Ces figures indiquent qu'un seul agent exécute les multiples CA(s) de manière concurrente. Cependant, les CA(s) concurrents peuvent chacun être envoyé vers un agent différent. Par exemple, CA-1 vers agent-2, CA-2 vers agent-3, ...etc. Un tel comportement de protocole est déjà supporté par UML, par ailleurs, les notations présentées en figure II.27 sont une extension recommandée d'UML.

➤ **Le diagramme de Collaboration ou Communication**

Le diagramme de collaboration représente une vue unique du système, autrement dit, c'est un ensemble de rôles joués par des objets dans un contexte particulier. Les liens et les interactions entre ces objets sont représentés à travers l'échange de messages.

La différence essentielle entre le diagramme de collaboration UML et celui d'AUML réside dans le fait que pour le premier (relatif à UML), un objet ne peut avoir qu'un seul rôle tout au long de la collaboration alors que pour le second, chaque agent peut jouer plusieurs rôles, i-e, porter plusieurs casquettes dans la même collaboration.

La figure II.29 illustre un exemple de diagramme de collaboration représentant des interactions entre des agents qui peuvent changer de rôle selon les besoins. Par exemple l'agent "B" sous le rôle de contractant (contractor) émettra une requête vers l'agent A qui joue le rôle de client. Ensuite l'agent B doit changer de rôle pour devenir analyseur de concurrents (Competitor Analyser) pour communiquer avec l'agent C qui a le rôle de concurrent (Competitor).

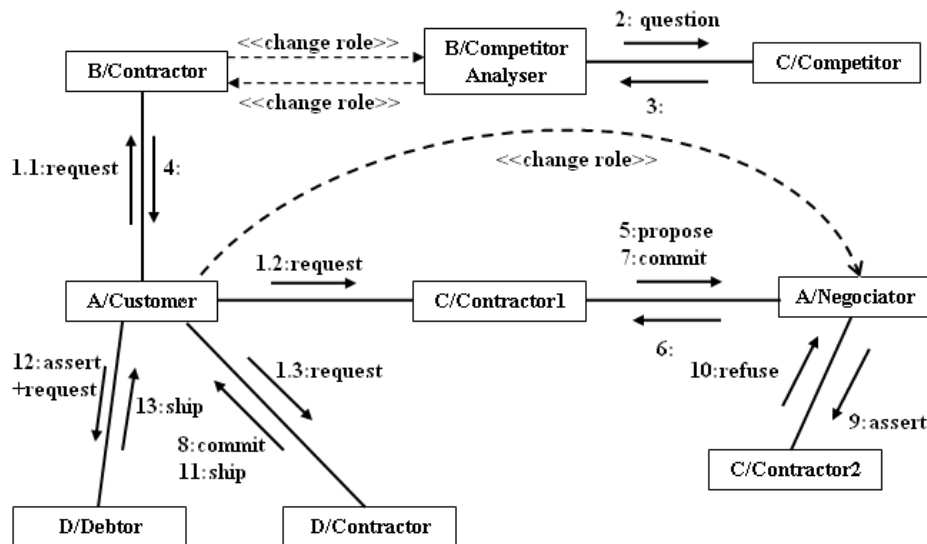


Figure II.29 : Diagramme de collaboration entre agents jouant plusieurs rôles.

L'approche par agents constitue un paradigme de programmation adéquat, puissant et efficace pour les applications d'un SIC et la recherche d'informations. Ceci, en facilitant aux acteurs de ce système d'information, une meilleure exploitation du flux d'informations circulant dans une entreprise ou organisation et matérialisant ainsi la coopération.

L'enseignement, constitue l'un des domaines favorisés pour l'usage des agents. Nous nous intéresserons particulièrement à deux types d'application des agents pour l'enseignement : les Systèmes Tuteurs Intelligents (STI) et les Environnements Interactifs pour l'Apprentissage Humain (EIAH) que nous allons présenter dans ce qui suit.

II.9 Les Systèmes Tutoriels Intelligents (STI)

Depuis le milieu des années 80, les avancées de la recherche dans le domaine de l'enseignement assisté par ordinateur ont donné lieu à l'émergence d'un nouveau champ d'investigation connu sous le nom de "Intelligent Tutoring Systems" (ITS) qui utilise l'intelligence artificielle, la psychologie cognitive et la vidéo interactive.

Le tutorat en intelligence artificielle consiste à mettre un tuteur artificiel à disposition de l'apprenant. Ce tuteur doit être capable de résoudre les mêmes problèmes que ceux posés à l'apprenant. Il pourra ainsi suivre le raisonnement de l'élève et "comprendre" ses erreurs.

II.9.1 Enseignement par ordinateur traditionnel versus systèmes tutoriels intelligents (Mahdaoui, 2002)

Un programme d'enseignement assisté par ordinateur traditionnel peut être comparé à un livre : il contient deux types de connaissances : celles relatives au contenu enseigné et d'autres, pédagogiques, qui ont trait à l'organisation ou à la structure des connaissances. Ces deux types de connaissances sont inséparablement liés dans un livre, la mise en forme du contenu détermine le type de pédagogie utilisée.

L'accès au contenu peut être varié, en lecture linéaire, par un index, par la table des matières, ...etc, mais reste néanmoins rigide. Le concepteur d'un programme sur ce modèle doit prévoir à l'avance toutes les interactions possibles. Il peut mettre une aide et un glossaire à disposition de l'utilisateur mais son programme sera incapable de répondre à des questions inattendues et de saisir qualitativement les progrès de l'utilisateur.

Un système de tutorat intelligent est construit différemment. Il contient avant tout les règles nécessaires à l'expert pour produire une interaction pédagogique à partir d'un corpus de connaissances. Ici, les connaissances pédagogiques et celles relatives au contenu sont séparées. Les connaissances relatives au contenu enseigné doivent être disponibles sous une forme "brute" qui permette leur utilisation indépendamment d'une interaction particulière. Il s'agit de reconstituer l'aspect relationnel de la communication pédagogique sur un ordinateur.

De cela découle la question suivante : « *Quand faut-il de l'intelligence artificielle dans l'enseignement assisté par ordinateur ?* ». Lorsque la tâche que l'on veut enseigner ne se réduit pas à un apprentissage par mémorisation, c'est à dire qu'une stratégie d'apprentissage par répétition du contenu n'est pas suffisante ou lorsqu'il s'agit d'acquérir une compétence qui nécessite une réflexion en plusieurs étapes, l'implication d'un agent extérieur peut être d'une aide précieuse.

II.9.2 Structure d'un STI

Dans son article « From Cognitive Modelling to the Design of Pedagogical Tools », Kurt Reusser (Reusser, 1993) décrit les Intelligent Tutoring Systems (ITS) comme étant composés de 4 modules de base comme le montre le schéma suivant :

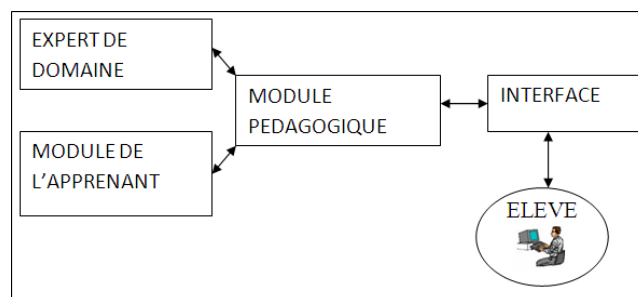


Figure II.30 : Architecture de base d'un STI

- a) **Un module expert du domaine de connaissance** : il contient une représentation des connaissances intrinsèques au domaine enseigné tel que le détiennent normalement les experts : faits, concepts et stratégies. Il contient également un programme expert qui résout les problèmes du domaine, qui utilise donc les représentations précédemment évoquées en les appliquant à un cas particulier.
- b) **Un module de l'apprenant qui permet une simulation cognitive de l'apprenant** : Il doit permettre un diagnostic et une évaluation de faible granularité de la compréhension de l'apprenant, de son niveau de connaissance et des processus de raisonnement qu'il utilise. Par ailleurs il sert à identifier des connaissances et des compétences émergentes chez l'apprenant.
- c) **Un module tuteur ou module pédagogique** : qui sert à orchestrer l'intervention pédagogique du système. Il doit pour cela être capable :
 - ✓ de modéliser l'interaction maître-élève ;
 - ✓ de sélectionner de manière adaptative les tâches présentées à l'apprenant ;
 - ✓ de donner des feed-back et de l'aide ciblés ;
 - ✓ d'exercer des activités comme donner des explications ou des conseils et faire passer des tests ;
- d) **Un module d'interface utilisateur** : qui s'occupe de l'ergonomie de l'interaction, c'est à dire de la communication entre système et utilisateur ainsi que de la manipulation directe, des graphiques et de la compréhension du langage naturel.

Un programme d'intelligence artificielle est fait de règles qui sont des énoncés du type : « *SI et ... ALORS faire x , activer la règle y etc...* ». L'expertise d'un système consiste à savoir choisir les bonnes règles à appliquer à un moment donné.

Les modules a), b) et c) sont constitués d'une base de règles. Le module tuteur nous intéresse plus particulièrement dans ce travail. Nombreux travaux de recherche tentent rendre le module tuteur indépendant du domaine de connaissance. Il contient alors des règles qui permettent de choisir un problème à présenter à l'apprenant en fonction de ce qu'il maîtrise déjà, ou encore de moduler le style d'interaction en le rendant plus autoritaire ou plus flexible.

➤ **Les principes de design d'un STI**

- ✱ Utiliser la technologie comme un outil didactique versus comme une fin en soi.
- ✱ Mettre à disposition de l'apprenant une aide procédurale et conceptuelle en implémentant un apprenant-expert (expert learner) ainsi qu'un expert du domaine de connaissance.
- ✱ Disposer d'outils permettant une représentation intelligible des données, de la communication et de la pensée.
- ✱ Respecter le principe de : variable control and minimal help, qui consiste à laisser un maximum d'initiative à l'apprenant tout en donnant de l'aide si elle est nécessaire ou demandée par l'apprenant.
- ✱ L'ordinateur devrait permettre à l'apprenant d'exprimer sa représentation du problème et à réfléchir à ses processus de pensée et aux produits de son apprentissage.
- ✱ Étendre l'usage des ordinateurs à un contexte de coopération. On a trop tendance à considérer l'apprenant comme un individualiste. La cognition est distribuée dans un environnement d'apprentissage (Lewis, 1998).

L'évolution des STI(s) a conduit à d'autres formes de systèmes d'enseignement utilisant l'informatique dont les EIAH(s) qui constituent aujourd'hui un domaine de recherche intense et dont nous allons donner quelques détails dans ce qui suit.

II.10 Environnement Interactif pour l'apprentissage Humain (EIAH) (Tchounikine, 2002), (Pesty & al, 2001)

Le terme EIAH présente l'intérêt de mettre au cœur de la problématique de l'apprentissage humain, la conception d'artefacts informatiques dédiés. Autrement dit, le terme EIAH désigne tout environnement informatique conçu pour favoriser un apprentissage humain où l'enjeu réside dans la mise en relation d'une intention didactique et d'un environnement informatique. Cette intention didactique peut se limiter à l'organisation du dispositif de formation dans lequel s'intègre l'artefact informatique ou se traduire également dans l'artefact lui-même.

La question du lien entre la recherche en EIAH et la recherche en informatique ne se pose donc pas en termes de type de système, mais en termes de problématique étudiée. Ainsi, la conception d'un EIAH de type « tuteur intelligent » ou STI pose des problèmes bien connus de recherche en informatique (résolution de problème, contrôle de l'interaction, etc.).

C'est également le cas de la conception d'une plateforme de diffusion de ressources pédagogiques sur le Web (« Learning Management System ») si elle est conçue pour proposer une gestion élaborée de scénarios pédagogiques ou aborder des problèmes tels que la malléabilité de l'environnement (i.e., si l'on se pose des problèmes qui sont spécifiquement liés au contexte d'apprentissage), alors que ce n'est pas le cas si elle est abordée comme un simple système d'information.

Au sens large, un EIAH est un environnement qui intègre des agents humains (e.g. élève ou enseignant) et artificiels (i.e., informatiques) et leur offre des conditions d'interactions, localement ou à travers les réseaux informatiques, ainsi que des conditions d'accès à des ressources formatives (humaines et/ou médiatisées) locales ou distribuées.

Les dernières années se caractérisent par le contexte créé par l'avènement des TIC et le « E-learning ». Ce terme qui ne se limite pas aux travaux de l'enseignement à distance fondés sur les TIC, mais dénote également les dimensions d'ouverture et de distribution introduites par les technologies qui conduisent à de nouveaux types de systèmes et renouvelle donc certaines des problématiques classiques du domaine.

Dans un EIAH, la machine peut donc avoir différents rôles non mutuellement exclusifs :

- ✱ Outil de présentation de l'information, par exemple, un hypermédia ou une plateforme Web spécialisée ;
- ✱ Outil de traitement de connaissances, comme par exemple, un système à base de connaissances résolvant les exercices avec l'élève ou un module pilotant l'interaction ;
- ✱ Outil de communication entre l'homme et la machine ou entre les hommes à travers les machines.

La problématique de l'interaction est cependant renouvelée dans les travaux où la machine est une machine « partenaire » des apprenants et des enseignants/tuteurs, voire

simplement un milieu dédié. Ces systèmes, s'ils dénotent une intention didactique par la nature des interactions qu'ils permettent et des activités qu'ils supportent, n'embarquent cependant pas nécessairement de connaissances du domaine de l'apprentissage (et l'apprenant n'est pas nécessairement sous le contrôle d'un tuteur pilotant l'interaction), sortant en cela du paradigme classique des EIAO¹⁴.

La prise de conscience de cette complexité suggère de chercher d'autres voies pour la conception d'un EIAH autre que celle qui consisterait à spécifier un module « pédagogue » prenant des décisions centralisées. Une issue alternative est, au contraire, de disposer d'agents spécialisés selon des types de compétences didactiques ou pédagogiques, capables de coopérer et de coordonner leurs actions pour fournir le meilleur appui à l'apprentissage.

Par ailleurs, la rapidité d'évolution des connaissances enjeux d'apprentissage, suggère que des agents humains soient intégrés d'emblée dans la conception d'un EIAH pour participer à une société qui éduque et est dynamique et flexible. D'après (Pesty & al, 2001), la fonction didactique d'un EIAH est une propriété émergente des interactions organisées entre ses composants et non une fonctionnalité déterminée et centralisée de l'un d'entre eux.

Ce point de vue trouve un appui naturel pour sa mise en œuvre dans les développements actuels des technologies « agent » en matière de développement logiciel ; technologies particulièrement pertinentes lorsque des critères comme la distance, la coopération entre entités distinctes ou encore l'intégration de logiciels existants sont à prendre en considération. En effet, poussée plus en avant, cette dernière approche suggère de repenser la globalité d'un EIAH comme un ensemble de composants élémentaires en interaction les uns avec les autres et de nombreux travaux vont dans cette direction (Pesty & al, 2001) (Mbala, 2003) (Gascueña & al, 2005) (Caballero & al, 2005).

Pour notre cas, nous adoptons l'approche qui considère au plan théorique, qu'un processus éducatif est le produit de la collaboration entre des agents didactiques et pédagogiques spécialisés et, au plan technologique, qui exploite les concepts et les méthodes de l'approche multi-agents de la conception de systèmes logiciels.

II.11 Conclusion

Dans ce chapitre nous avons d'abord présenté les principaux éléments relatifs à technologie agent aussi bien sur les plans des concepts que des technologies. Un intérêt particulier a été porté aux Systèmes Multi-Agents, technologie sur laquelle va reposer le système dédié à l'assistance dans le processus de tutorat par les workflows que nous prévoyons dans ce travail.

Enfin, nous avons terminé par une présentation des Systèmes Tuteurs Intelligents et Environnement Interactifs pour l'Apprentissage Humain d'abord pour l'intérêt qu'ils présentent par rapport à notre domaine d'application, i-e, le tutorat distant par les workflows et ensuite pour ses liens étroits avec l'usage de l'IAD et de la technologie agent.

La prochaine partie de cette thèse consiste à présenter la solution de tutorat distant envisagée en utilisant les technologies workflow et le multi-agent.

¹⁴ Enseignement Intelligemment Assisté par Ordinateur.

Partie II

**Une Infrastructure Statique et Dynamique
pour le Travail Coopératif appliquée
au tutorat en e-learning**

Chapitre III

**Une démarche à Base de Méta-Modèle
de Workflow pour le Tutorat en e-learning**



Chapitre III

Une démarche à base de méta-modèle de Workflow pour le tutorat en e-learning

L'état de l'art nous a permis de dégager les différents concepts et notions nécessaires pour l'accomplissement de ce travail. En effet, le choix d'une approche de modélisation à base de méta-modèle pour processus liés à l'enseignement découle de la nature même du processus à modéliser comme nous l'avons longuement expliqué en introduction.

Le e-learning étant issu à l'origine des milieux professionnels, son objectif initial était d'améliorer les performances des employés dans une entreprise, leur assurer une formation continue toujours dans le cadre de leur travail sans avoir à les détacher. Les objectifs du e-learning étaient fortement liés aux performances de l'entreprise et aux compétences exigées pour les atteindre.

Evidemment, les notions de profit et de valeur ajoutée, en termes de management sont placées en tête des objectifs stratégiques visés derrière un tel dispositif (de e-learning). Mais vu la richesse et la maniabilité des environnements dédiés au e-learning, ces derniers se sont vus propulsés vers d'autres niveaux nettement plus pédagogiques et éducatifs. Aujourd'hui donc, on parle de e-learning à tous les niveaux d'enseignement, notamment le niveau universitaire. Plusieurs disciplines interviennent dans la définition de ce qu'est un dispositif de e-learning allant des spécialistes en pédagogie, en psychologie et psychologie cognitive jusqu'aux spécialistes en TICE¹ (s).

Chacun essaye d'apporter sa pierre à l'édifice car l'enjeu n'est plus économique au premier plan mais plutôt « éducatif » et/ou de « formation professionnelle² » selon le niveau auquel on se retrouve.

Notre intérêt se porte particulièrement sur le e-learning dans un cadre d'enseignement supérieur où ce dernier pourrait être appliqué de deux façons distinctes :

1. Utiliser les dispositifs de e-learning comme un **complément** à ce qui est déjà dispensé en présentiel et dont nous expliquerons les motivations plus loin.
2. Permettre à l'université de dispenser ses formations à distance à un public n'ayant pas l'accès à des institutions universitaires pour des raisons diverses.

Nous sommes partisans de la première approche pour les raisons suivantes :

¹ Technologies de l'Information et de la Communication pour l'Enseignement (ou l'Education).

² Dans un contexte de formation professionnelle, l'enjeu économique peut très bien remonter en surface selon l'importance de la formation elle-même. C'est le cas des formations pointues et onéreuses.

- ✿ La formation présentielle est la principale mission de toute université disposant de locaux et d'effectifs prévus à cet effet. Il n'est donc pas question de changer cette vocation du moins pas d'une façon brutale et mal étudiée.
- ✿ En plus, il ne s'agit pas de troquer ce qui marche « plus ou moins bien » contre quelque chose de « non bien maîtrisé encore » ou dont on ne connaît pas l'issue. Il n'est pas non plus question d'éliminer l'enseignement classique mais plutôt de l'améliorer.
- ✿ Par ailleurs, l'évolution rapide des connaissances dans tous les domaines font que l'usage traditionnel du temps d'étude réparti en séances présentielles de « Cours », « TD », « TP » (par exemple) et monopolisant les étudiants à longueur de journée n'est peut être pas la meilleure manière de transmettre le maximum de savoir.
- ✿ Enfin, nous entendons par complément, la mise en place de dispositifs favorisant le travail individuel ou collaboratifs des étudiants tout en prenant compte des contraintes temporelles imposées. Cependant, l'efficacité d'une telle approche aurait tout à gagner par une meilleur distribution des temps d'étude en présentiel en faveur des cours à distance. Cela donnerait plus de flexibilité pour les étudiants et les enseignants.

La seconde approche nous semble être dans une perspective beaucoup plus lointaine même si certaines expériences à travers le monde ont montrés leur efficacité. En plus, la formation à distance en général a toujours été une mission assurée par des organismes spécialisés en la matière dont pourrait très bien faire parti le e-learning. Certains fervents défenseurs du passage vers un mode e-learning pur arguent des motifs comme le nombre d'étudiants sans cesse croissant, l'insuffisance des infrastructures ou encore l'évolution rapide des domaines de connaissance. Pour eux, ce serait plus efficace et nettement moins couteux, chose qui est largement discutable.

Rappelons enfin que notre objectif dans ce travail n'est pas de trouver une solution à la question de l'introduction du e-learning dans les institutions de formation car son étendue est beaucoup plus vaste que la problématique abordée et qui est celle du tutorat à distance.

Au sens système d'information classique, une institution d'enseignement est une organisation pourvue d'une mission d'éducation et de formation, d'un ensemble d'acteurs ayant des responsabilités et représentant ses compétences et son savoir-faire ainsi que d'un ensemble de moyens lui permettant d'accomplir ses tâches. Par analogie, un « dispositif de e-learning » peut être vu comme une institution virtuelle de formation faisant intervenir un ensemble d'acteurs qui coopèrent entre eux dans un but d'assurer des enseignements.

ASPI³ est une démarche de conception de dispositif d'enseignement dont le e-learning et utilisant les technologies modernes. Son principe est d'utiliser les TICE(s) autant que possible afin d'améliorer la qualité de l'apprentissage et de l'enseignement. Elle offre aussi un bon cadre pour l'évaluation des performances de chaque constituant d'un dispositif e-learning (Peraya & al, 2006).

³ Accompagner, Soutenir et Piloter un projet d'Innovation techno-pédagogique (ici e-learning).

D'après la démarche ASPI (Peraya & al, 2004) qui perçoit le dispositif de e-learning comme un projet d'innovation techno-pédagogique, il est important de spécifier le contexte d'insertion d'un dispositif de e-learning. On entend par contexte d'insertion, le cadre de mise en place du dispositif par rapport à l'institution créatrice et de l'environnement institutionnel et juridique qui l'entoure.

Ainsi par exemple, la mise en place d'un tel dispositif par la FEI⁴ de l'USTHB⁵ pourrait donner lieu aux contextes d'insertion suivant (Table III.1):

USTHB	1 (macro)
FEI	2 (méso)
Département INFO	3 (micro)
Filière Ingéniorat	1 (macro)
Module I.SY.I	2 (méso)
TP-SI-01	3 (micro)

Table III.1 : Contexte d'insertion d'un dispositif de e-learning

La définition du contexte est à même de nous guider sur les prérogatives du dispositif, ses ambitions ainsi que les moyens et les ressources à mettre en jeu. La démarche encourage aussi l'émergence de rôles assurés par ce qu'ils appellent le « Staff Techno-Pédagogique » qui n'est autre que l'ensemble des enseignants et managers destinés à assurer la mission de l'enseignement à distance⁶.

Des rôles peuvent aussi émerger du public cible à qui est dédié le dispositif, ce sont principalement les apprenants. Chaque rôle est investi d'un ensemble de fonctions qu'il doit assurer dans le dispositif.

La démarche préconise aussi que l'on définisse le ou les supports technologiques à même de rendre le dispositif « réalisable » : on doit préciser dans ce contexte, la plate forme ou le LMS⁷ à utiliser ainsi que les outils logiciels d'accompagnement. ASPI recommande aussi que l'on spécifie l'ensemble des scénarios pédagogiques, des contenus et des pré-requis nécessaires au bon déroulement des sessions d'apprentissage.

Les ressources pédagogiques constituent l'ensemble des moyens didactiques et des références bibliographiques utiles aux apprenants, plus tout document susceptible de les aider sur le fonctionnement de leurs cours (voir annexe B).

Comme on peut l'observer, ASPI couvre tous les aspects concernant la description d'un dispositif de formation à distance. La question première qui émerge de tout cela est : « *Un dispositif de e-learning tel que décrit par ASPI, constitue-t-il un Système d'Information Coopératif (SIC) ?* ». L'analyse des éléments décrits dans la démarche permet de répondre affirmativement grâce aux arguments suivants :

- ✓ Nous avons des acteurs qui accomplissent des rôles et des tâches bien définis.

⁴ Faculté d'Electronique et d'Informatique.

⁵ Université des Sciences et de la Technologie Houari Boumédiène.

⁶ Les détails complets sur la démarche sont présentés en annexe B. L'annexe C présente l'exemple d'un projet décrit par la méthode. Ce même projet a été utilisé dans la partie validation de cet ouvrage.

⁷ Learning Management System.

- ✓ Ces rôles, et par le biais des fonctions dont ils sont investis interviennent dans le cadre de procédures (ou processus) décrites dans la démarche d'enseignement lui-même.
- ✓ Ces acteurs coopèrent via une plate forme dotée d'outils technologiques.

Le schéma suivant résume le fonctionnement d'un environnement e-learning :

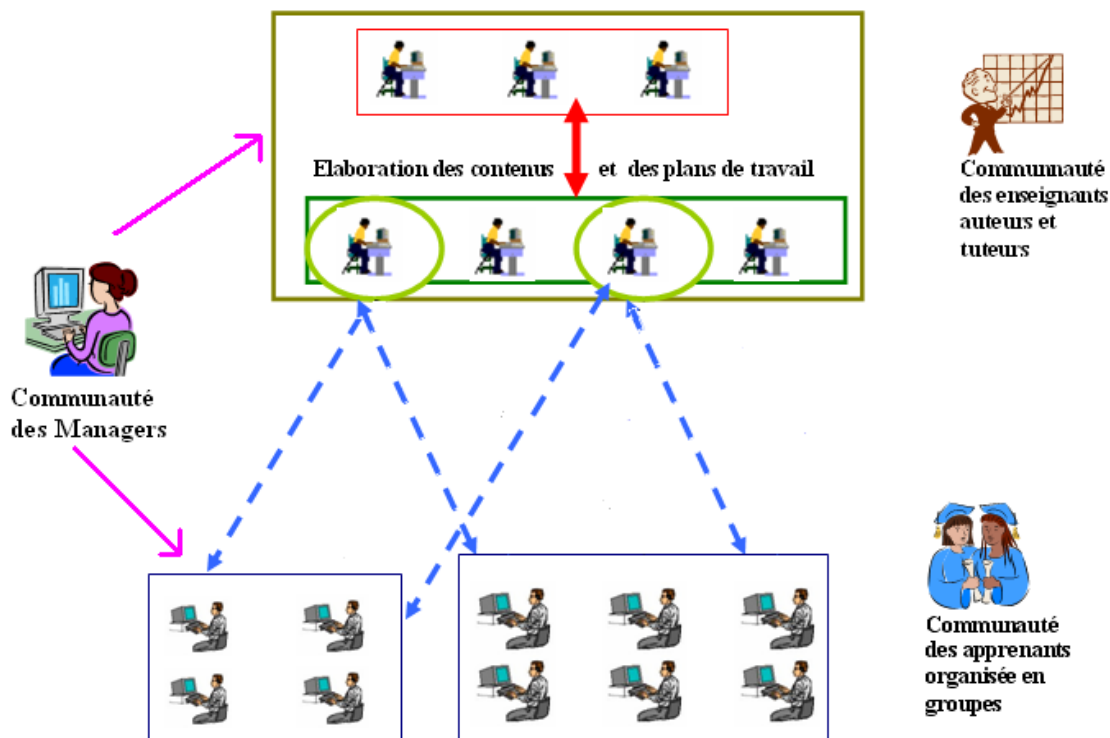


Figure III.1 : Communautés du e-learning

Comme on peut le constater, trois communautés d'acteurs interviennent dans un dispositif de e-learning que sont les : administrateurs, les enseignants et les étudiants. Les liens de coopération peuvent être retrouvés à différents niveaux et pour divers objectifs.

Vient alors la seconde question : « *Quels sont les processus de ce système d'information en terme de nature et de technologies applicables ?* ». Rappelons ici que dans la partie état de l'art, nous avons passé en revue différentes classifications des systèmes d'information coopératifs. Dans le schéma figure de la figure 3.1, nous retrouvons bien les « 3C » du Groupware que sont : Communication, la Collaboration et la Coordination. A notre sens, il existe au moins deux classes de processus dans le e-learning :

- ✱ Processus de management ou d'administration : ce sont des processus concernant la gestion des adhérents à la plate forme (enseignants, étudiants), la gestion des contenus et des supports technologiques.
- ✱ Processus d'enseignement : ce sont des processus liés au déroulement des enseignements eux-mêmes engageant des « Staff pédagogiques », des « Etudiants » et des « Managers ». Ces processus peuvent être liés à l'apprentissage lui-même, i-

e, concernant l'accomplissement des activités pédagogiques faisant intervenir l'apprenant et les enseignants devant l'encadrer ou le suivre.

Comme nous pouvons le constater, tous les éléments nécessaires à la modélisation en tant que système de travail coopératif assisté par ordinateur se retrouvent. Cependant, on se limitera à décrire uniquement les rôles impliqués dans le tutorat. Dans la section qui suit, nous allons présenter quelques concepts et définitions liés au domaine de l'enseignement afin d'en faciliter la compréhension dans le reste de cette partie.

III.1 Définitions & concepts de base

Afin de permettre une meilleure compréhension des différents concepts manipulés, nous présentons dans ce qui suit quelques définitions et notions de base.

III.1.1 Famille, Groupe & Equipe

Une famille est formée d'apprenants réunis autour d'un programme de formation donné. Par exemple, il est possible de former des familles selon des critères géographiques, la matière enseignée ou encore sur la base d'un système d'évaluation.

Un groupe est formé d'apprenants ayant le même niveau de connaissances ou les mêmes pré-requis. Un groupe est formé par des apprenants de la même famille.

Une équipe est un groupe restreint d'apprenants appartenant à la même famille mais pas nécessairement au même groupe. Les membres d'une même équipe doivent partager un sous ensemble de pré-requis en plus des pré-requis propres à chaque individu. Ils ont à coopérer, interagir et coordonner leur travail en vue d'atteindre un objectif commun.

III.1.2 Objectifs Pédagogiques

Dans la littérature, certains auteurs recommandent de définir trois niveaux d'objectifs pédagogiques que sont : les "Objectifs Pédagogiques Globaux" (OPG), les "Objectifs Pédagogiques d'Intégration" (OPI) et les "Objectifs Pédagogiques Opérationnels" (OPO) (Mager, 1977).

Un OPG représente un objectif pédagogique global à atteindre dans le cadre d'un enseignement donné. Il peut s'agir d'un thème, d'une matière ou d'un module par exemple. Un OPG peut être décomposé en sous-objectifs de type OPI.

Un OPI définit un objectif pédagogique plus affiné qu'un OPG et couvre généralement un thème particulier d'un sujet plus général. Un OPI est décomposable en plusieurs objectifs de type OPO.

Un OPO est un objectif pédagogique directement évaluable et concerne en général un concept ou une compétence particulière qu'un apprenant aura à acquérir.

L'exemple suivant explique le découpage d'un cours selon des objectifs pédagogiques :

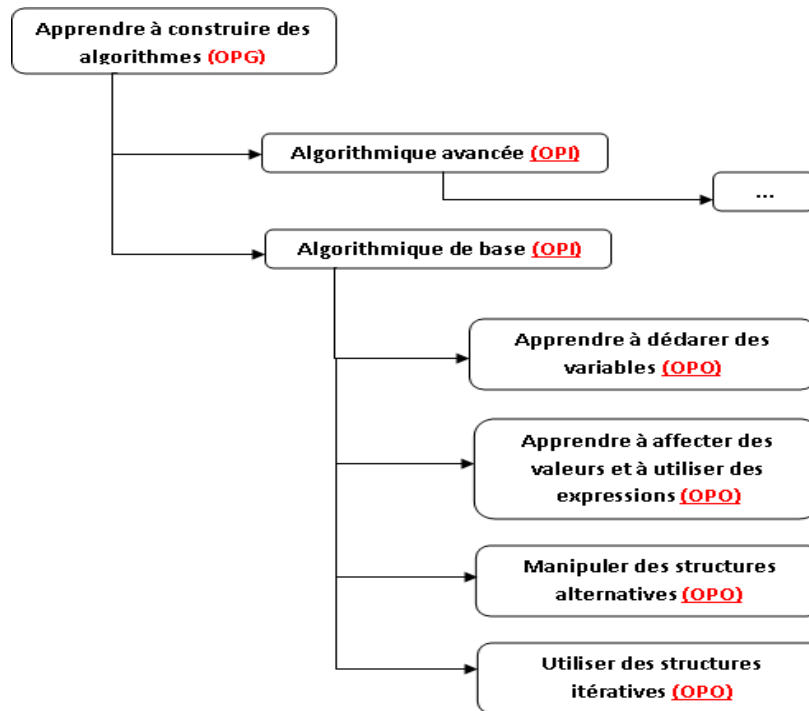


Figure III.2 : Exemple de découpage d'un cours selon des objectifs pédagogiques

III.1.3 Tâche, Activité Individuelle & Activité Collective

Une tâche est une unité élémentaire de travail. Une tâche d'apprentissage concerne l'apprenant et une tâche pédagogique concerne l'enseignant ou tuteur. Une tâche peut avoir au plus un objectif pédagogique opérationnel (OPO) comme par exemple "apprendre à utiliser un certain type d'instruction dans un programme". Dans le cas où une tâche n'a pas d'objectif pédagogique, cela signifie que ce dernier est relevé au niveau de l'activité qui contient la tâche elle-même.

Une activité individuelle est une séquence de tâches d'apprentissage à accomplir par un apprenant tout seul et indépendamment des autres. Notons que pour le tutorat, il s'agira d'activités pédagogiques accomplies par un seul tuteur.

Une activité collective est une séquence de tâches collectives dont l'accomplissement concerne tous les membres d'une même équipe. Celle-ci peut être exécutée de manière synchrone ou asynchrone comme nous le verrons dans l'étude de cas qui viendra dans la suite.

III.1.4 Plan d'Apprentissage Individuel (PAI) & Plan d'Apprentissage Collectif (PAC)⁸

Un Plan d'Apprentissage Individuel (PAI) est un ensemble d'activités d'apprentissage individuelles planifiées par le tuteur. Ces activités sont attachées entre elles par des liens de séquence, de choix et/ou de parallélisme. L'apprenant aura à exécuter ces activités tout seul en respectant l'ordre défini par le tuteur et sous son

⁸ En anglais, nous définissons : "Individual Learning Plan" (ILP) et "Collective Learning Plan" (CLP).

contrôle. En terme de workflow, un cas de workflow (ou cas de PAI) concerne un et un seul apprenant.

Un Plan d'Apprentissage Collectif (PAC) est un ensemble d'activités d'apprentissage planifiées par le tuteur et reliées entre elles par des opérateurs de séquence, de choix et/ou de parallélisme. Un PAC concerne tous les membres d'une même équipe qui devra accomplir les activités du PAC de manière synchrone ou asynchrone.

Dans un PAC, il est possible de trouver des parties de travail individuel et au moins une partie de travail collectif obligatoire car l'équipe a à atteindre un objectif commun. En effet, une équipe est concernée par un cas de workflow qui représente un cas de PAC et partagent donc la même "worklist" ou corbeille de travail.

Certains travaux relatant la pédagogie par projet comme (Sébastien & al., 2001) proposent d'organiser un projet d'apprentissage en deux phases : Individuelle et Collective. En se basant sur cette idée, nous considérons un PAC comme une alternation de phases individuelles et collectives. Chaque phase peut contenir une ou plusieurs activités, comme le montre le schéma de la figure III.3 :

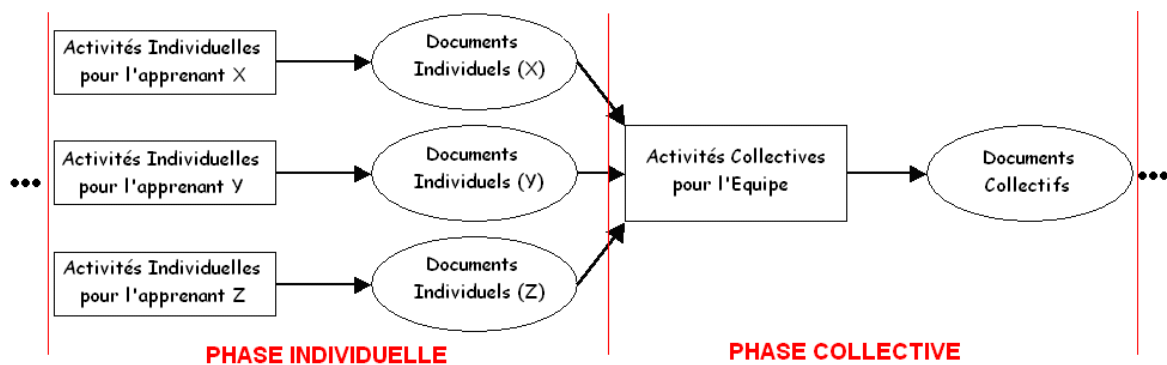


Figure III.3 : Phases d'un PAC

III.1.5 Tutorat Individuel & Tutorat Collectif

Le tutorat individuel est le suivi et le contrôle par le tuteur d'un plan d'apprentissage individuel (PAI) soumis à un apprenant particulier. Le tuteur doit assister et évaluer le travail de l'apprenant.

Le tutorat collectif concerne le suivi d'un plan d'apprentissage collectif soumis à une équipe d'apprenants. Le tuteur devra assister tous les membres de l'équipe et évaluer à la fin le travail collectif.

III.1.6 Processus e-Learning ou de Tutorat Distant

Un processus de e-learning ou de Tutorat Distant représente l'ensemble de tous les PAI(s) et/ou PAC(s) construit en vue d'atteindre un objectif pédagogique global (OPG) dans le cadre d'un enseignement ou d'une formation donnée. Cela peut concerner une matière, un module ou même un programme de formation.

En termes de workflow, on parlera de processus workflow pour le e-learning. Les PAI(s) et PAC(s) peuvent être des sous-processus du processus global ayant chacun son objectif pédagogique à atteindre. Mais rien n'empêche de considérer un processus e-learning comme un seul PAI ou PAC.

Au fait, le découpage et le niveau de granularité (des sous-processus, des activités et des tâches) sont laissés aux soins de l'équipe pédagogique en charge de décrire le processus de e-learning.

Pour tout le reste de ce travail on parlera indifféremment de "processus de e-learning" ou de "tutorat distant" pour le e-learning.

III.1.7 Adaptabilité et adaptativité

Nous parlerons d'adaptabilité lorsqu'une instance particulière (cas de workflow) du modèle de workflow est modifiée.

L'adaptativité est lorsque le modèle de workflow lui-même subit des modifications.

III.2 La démarche de modélisation à base de méta-modèle

La démarche de développement d'un processus de e-learning ou de tutorat à distance permet de couvrir les aspects statiques et dynamiques de la modélisation. Nous proposons une démarche composée des quatre phases suivantes :

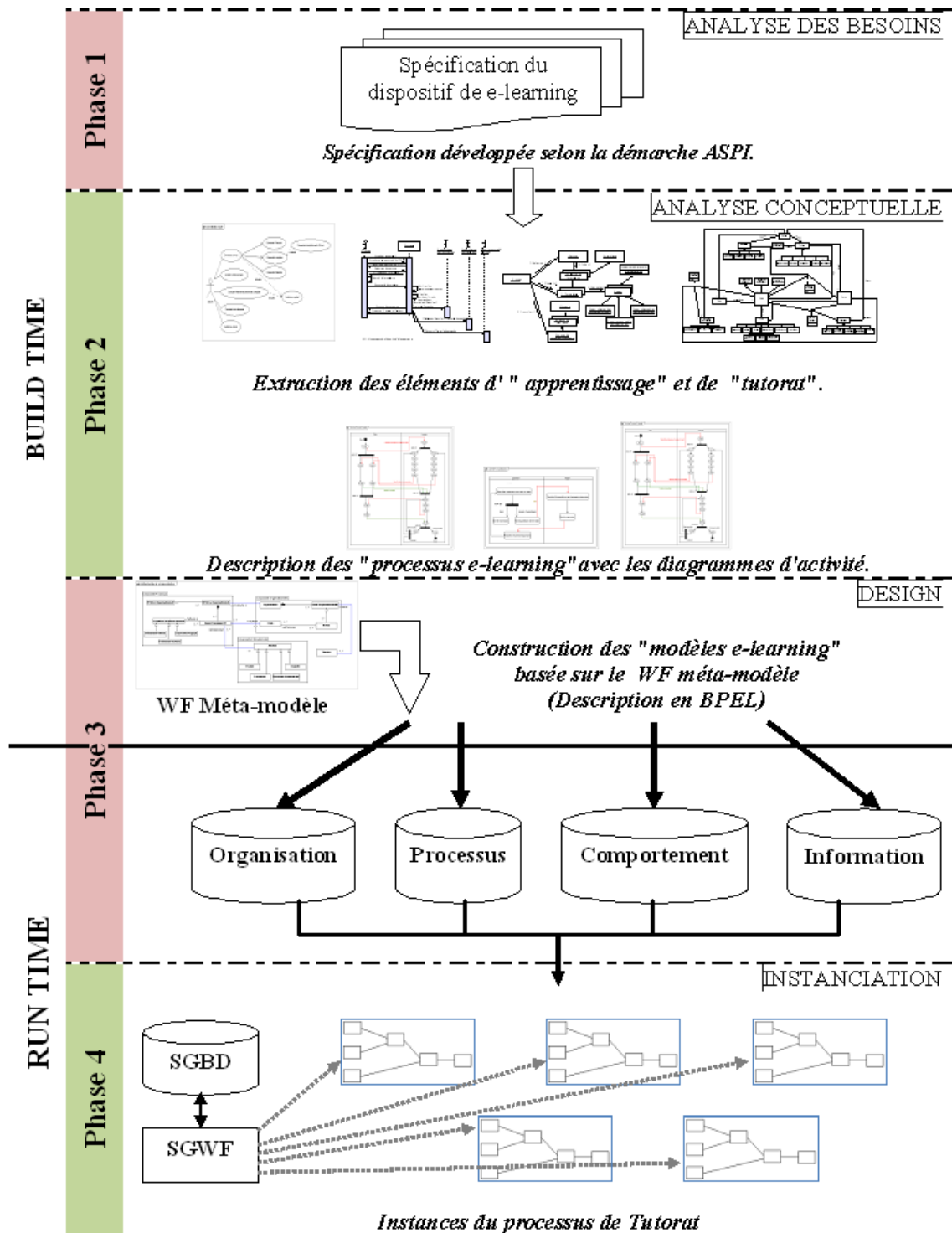


Figure III.4 : Etapes de développement d'un processus de tutorat distant

Le schéma ci-dessus récapitule les principales étapes de la démarche à utiliser pour modéliser et instancier les processus workflows pour le e-learning.

Les deux premières phases ("Analyse des besoins" et "Analyse Conceptuelle") sont préliminaires et spécifiques au domaine e-learning. La troisième phase ("Design") est pour la construction des modèles et leur implémentation (en BPEL) tandis que la quatrième phase est pour l'instanciation et l'exécution des modèles. Enfin, rappelons que

tout processus workflow passe par deux étapes (Van Der Aalst, 2000) (Levan, 2000) : la phase de construction (build-time) puis celle d'exécution (run-time). Les phases 1 et 2 sont relatives au build-time, la phase 4 au run-time, alors que la phase 3 est une phase commune entre les deux.

III.2.1 Analyse des besoins (Elaboration du cahier des charges)

Cette phase est cruciale pour tout le reste du processus de développement car elle doit définir de manière assez détaillée toutes les facettes du dispositif de e-learning à mettre en place. C'est à partir de ce dispositif que seront extraites les parties concernant l'apprentissage et le tutorat. Cependant, rien n'empêchera de démarrer à partir d'un cahier des charges déjà élaboré par exemple, dans le cadre d'une formation. Dans ce cas on procédera à l'analyse des parties que l'on voudrait automatiser par les workflows.

Nous avons expliqué précédemment que notre choix s'est porté sur cette méthode pour l'analyse des besoins car elle offre un cadre riche en concepts spécifiques pour l'enseignement et la pédagogie. Ceci aura pour effet de ne pas nous écarter du domaine d'application, vu sa spécificité et sa sensibilité. Il n'est pas difficile d'imaginer les conséquences néfastes qu'aurait le déploiement d'un processus mal étudié sur une communauté d'apprenants et/ou de tuteurs.

Les grandes lignes de la démarche ASPI sont décrites en annexe A. L'annexe B présente quant à elle le cahier des charges concernant l'étude de cas présentée dans cette thèse. La démarche est très flexible et permet de définir un dispositif de e-learning selon le degré de granularité souhaité. En effet, un dispositif peut concerner un cursus complet de formation, comme il peut concerner un module d'enseignement ou encore une activité traditionnelle d'enseignement comme les travaux dirigés (TD) ou les travaux pratiques (TP) par exemple.

III.2.2 Analyse Conceptuelle (Conception des processus de tutorat distant)

Cette étape devra se solder par le recueil des informations et l'analyse des besoins, i-e, recenser les acteurs, leurs rôles et les ressources nécessaires. Les fonctionnalités attendues du système peuvent éventuellement être décrites par les diagrammes de cas d'utilisation. L'étape de conception consiste donc à extraire les éléments concernant le tutorat à partir du cahier des charges et à les modéliser à l'aide des diagrammes d'UML. Elle permet de se focaliser uniquement sur les aspects couvrant l'apprentissage de l'apprenant et son suivi/encadrement par ses tuteurs. Les diagrammes que nous préconisons principalement sont :

- ✱ Le diagramme d'activités qui permet de décrire la dynamique des processus workflow. Il est même possible de n'utiliser que ce type de diagramme s'ils représentent suffisamment bien les aspects de synchronisation et de communication du processus e-learning.

Mais rien n'empêche d'utiliser d'autres diagrammes s'il s'avère nécessaire de considérer d'autres aspects spécifiques du processus comme :

- ✿ Mettre en évidence les aspects d'interaction et de synchronisation par les diagrammes de séquence.
- ✿ Identifier les acteurs et les objets en coopération par les diagrammes de collaboration.
- ✿ Construire le diagramme de classe global représentant toutes les entités du système, à partir des ébauches de diagrammes de classe associés aux diagrammes de collaboration.

III.2.3 Le Design (Construction et Implémentation des modèles de processus de tutorat distant)

Cette phase permet de construire les modèles de processus conformément au méta-modèle de workflow établi à cette effet et couvrant quatre aspects : organisationnel, fonctionnel, comportemental et informationnel. Les modèles sont alimentés par les éléments modélisés par les diagrammes d'UML de la phase précédente, notamment les diagrammes d'activités.

Le diagramme d'activité est ensuite translaté vers un langage de spécification de workflow. Notre choix s'est porté sur le langage BPEL qui permet la description de processus workflow en plus de ses capacités d'orchestration de web-services.

Donc, suite aux étapes décrites ci-dessus, les éléments clés relatifs au workflow (acteurs, rôles, activités, tâches, objectifs pédagogiques, processus, ...) seront dégagés, permettant ainsi l'instanciation du méta-modèle afin d'obtenir les modèles de processus de tutorat relatif au cas d'application. Une présentation détaillée du méta-modèle sera donnée en section (III.3).

III.2.4 Instanciation des modèles de processus e-learning

Après l'implémentation des différents aspects dans le modèle de processus, cette phase consistera à générer les instances de modèles et à suivre leur exécution par un système de gestion de workflow (SGWF) interagissant avec un système de gestion de base de données (SGBD).

III.3 Méta-modèle de workflow pour le tutorat en e-learning

Le méta-modèle fait ressortir les principaux concepts du workflow (WFMC) tels que : acteur, rôle, processus, sous-processus, document, ... etc.

Il est étendu aux concepts définis précédemment et reliés au monde de l'enseignement tels que : le tuteur, l'auteur de cours, le groupe, l'équipe et/ou la famille d'apprenants, l'apprentissage individuel, l'apprentissage collectif, les objectifs pédagogiques, ...etc.

Il couvre quatre aspects complémentaires : organisationnel, fonctionnel, comportemental et informationnel, que nous détaillons dans les sous-sections qui suivent.

III.3.1 Aspect organisationnel

Cet aspect gère deux grandes catégories d'acteurs que sont les acteurs humains et les acteurs automatiques ou artificiels. Les acteurs automatiques seront traités dans le chapitre suivant de cette seconde partie du document (chapitre IV).

Les rôles spécifiés pour les acteurs humains sont :

- **Auteur** : il s'agit d'enseignants dont la mission est de concevoir et préparer des contenus pédagogiques, principale ressource pour le travail des apprenants.
- **Tuteur** : c'est un enseignant dont le rôle est de suivre, assister et évaluer le travail des apprenants durant leur parcours d'apprentissage.
- **Apprenant** : c'est l'élève ou l'étudiant à qui reviendra la tâche d'accomplir les PAI et/ou les PAC.

Cependant, les rôles sont facilement extensibles à d'autres rôles pédagogiques comme celui d'examineur par exemple. La figure III.5 décrit la composante organisationnelle du méta-modèle :

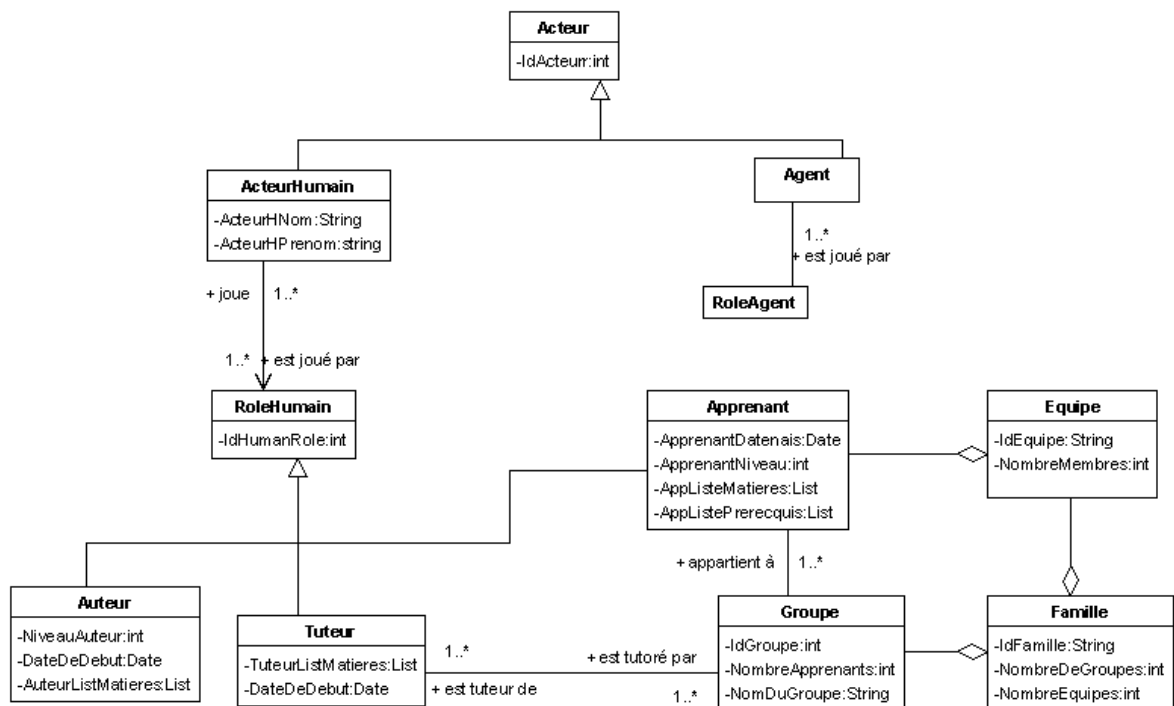
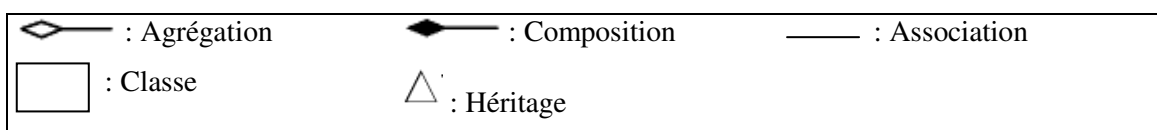


Figure III.5 : Détails de l'aspect organisationnel

Légende :



Chaque acteur a une identité unique "*IdActeur*" qui le distingue des autres dans le système. De plus chaque nouveau rôle défini dispose d'un certain nombre de propriétés spécifiques : pour le rôle d'auteur, il est nécessaire de préciser le profil de l'enseignant "*NiveauAuteur*" ainsi que la liste des matières sur lesquelles il intervient. Pour un tuteur,

il est nécessaire de préciser la liste des matières "*TuteurListeMatières*" à tutorer. Enfin, pour un apprenant on précisera entre autres son niveau "*ApprenantNiveau*", ses matières d'étude "*AppListeMatières*" et ses pré-requis "*AppListePrérequis*".

Les rôles d'enseignants peuvent être joués indifféremment par le même acteur ou par des acteurs différents.

Le concept de **groupe** est introduit pour supporter l'apprentissage individuel (PAI), alors que les concepts d'**équipe** et de **famille** sont introduits pour supporter l'apprentissage collaboratif (PAC). On forme des familles relativement à un cursus de formation donné.

Un groupe ou une équipe sont tutorés par un et un seul tuteur dans le cadre d'une matière ou d'un module donné.

Un groupe ou une équipe d'apprenants peut être formé par rapport au niveau de connaissance ou pré-requis ou bien n'importe quel critère jugé d'intérêt. Cependant, dans le cas de l'équipe, les notions d'adhésion et de cohésion sont importantes.

D'après (Saadoune & al., 1996), l'adhésion est la force du sentiment d'appartenance à un groupe. Alors que la cohésion est la force d'une équipe à coordonner ses activités. Bien que ces notions se retrouvent surtout dans le domaine du management d'équipes, il serait intéressant de développer cet aspect pour former des équipes d'apprenants selon des critères étudiés.

Dans le cadre de nos travaux de recherches, nous nous intéressons à développer cet aspect (Boukhedouma & al., 2006), mais il ne sera pas discuté ici car il dépasse le cadre de cette thèse.

Enfin, pour une question d'efficacité et de facilité de suivi (tutorat), (Lewis, 1998) et (Sebastien & al., 2001) recommandent que les équipes d'apprenants soient composées de deux à quatre étudiants au maximum.

III.3.2 Aspect fonctionnel

Cet aspect permet de décrire un processus de e-learning en termes de sous-processus (PAI et/ou PAC), d'activités (Individuelles ou Collectives) et de tâches, indépendamment des règles, des événements et des contraintes auxquelles ils sont soumis. En considérant donc le travail d'apprentissage des apprenants et celui de tutorat d'un tuteur comme un seul processus (de "e-learning" ou de "tutorat distant"), il est possible de décomposer ce dernier autant de fois que nécessaire jusqu'à arriver au niveau de détail souhaité en termes d'activités et de tâches.

Rappelons qu'un PAI ne peut être constitué que par des activités individuelles alors qu'un PAC doit contenir au moins une activité collective.

Un sous-processus peut être décomposé en d'autres sous-processus autant de fois que nécessaire. Chaque activité pourra alors être décomposée itérativement en d'autres activités ou bien en une séquence de tâches qui sont des unités de travail élémentaires, donc non décomposables.

Le choix du niveau de granularité pour la décomposition est laissé au tuteur seul ou mieux encore au "staff pédagogique"⁹ qui pourra le définir durant l'étape de l'élaboration du cahier des charges ou bien durant l'analyse de ce dernier. L'itération au niveau sous-processus et activité permettra par exemple, de bien présenter de larges cursus de formation. La figure suivante détaille les propriétés de cet aspect :

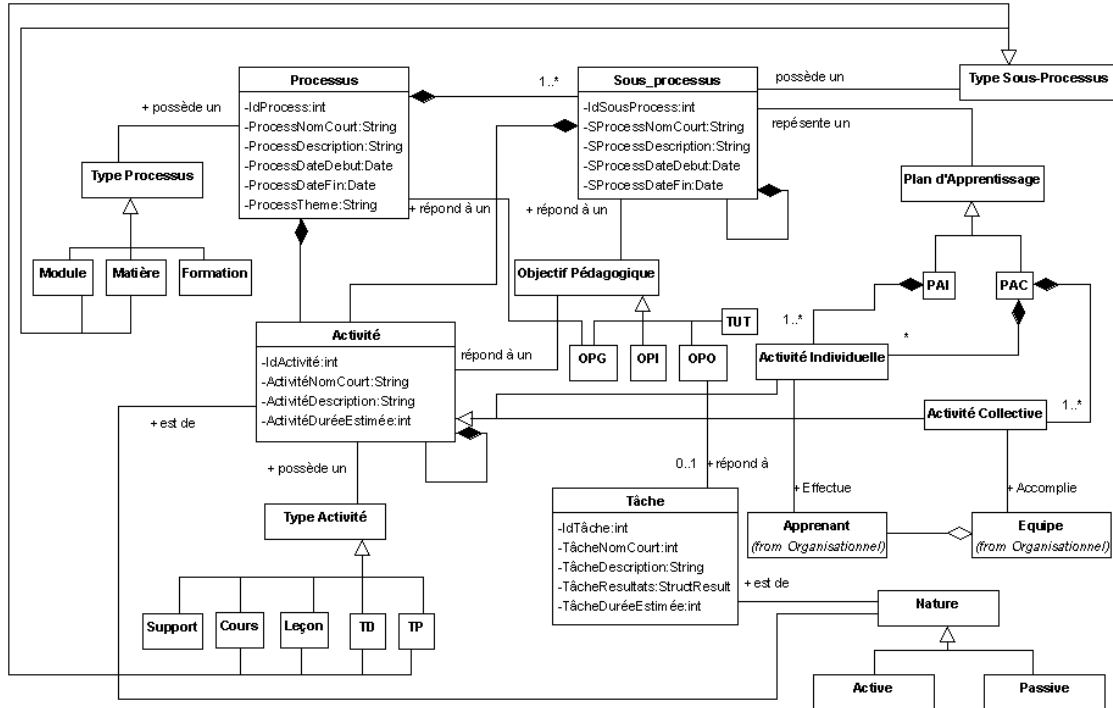


Figure III.6 : Détails de l'aspect fonctionnel

Naturellement, chaque niveau de granularité d'un processus (sous-processus, activité, tâche, ...) doit être identifiable de manière unique (propriétés commençant par "Id"). De plus, pour chaque part du processus, on doit préciser un "NomCourt" et une longue "Description" qui seront utilisées pour l'affichage au niveau des interfaces des apprenants et des tuteurs. Une attention particulière devra être apportée à la qualité du texte à écrire notamment pour les apprenants.

Un sous-processus représente un plan d'apprentissage pouvant être un PAI ou un PAC dont les définitions ont été données plus haut (cf. III.1).

Une activité peut être définie comme une *Activité Individuelle* à effectuer par l'apprenant tout seul ou comme une *Activité Collective* à accomplir par une équipe d'apprenants. Evidemment, toutes les activités du tuteur seront individuelles.

Certains de nos travaux se sont intéressés à la façon de réaliser et d'évaluer l'apprentissage collaboratif (Boukhedouma et al, 2006). Dans le cadre de cette thèse, nous relaterons uniquement l'aspect de mise en œuvre.

Nous proposons deux manières de réaliser la répartition des activités d'apprentissage dans un contexte collaboratif (PAC).

⁹ Equipe pédagogique composée au minimum d'un responsable de formation ou matière et d'un tuteur.

La première approche propose un partage équitable des activités et/ou tâches selon le nombre d'apprenants dans l'équipe. Les activités collectives doivent être obligatoirement accomplies par tous les membres de l'équipe alors que les activités individuelles doivent l'être une et une seule fois par un des membres de l'équipe¹⁰. En effet, pour les parts de travail d'apprentissage individuel, chaque apprenant devra exécuter la part qu'il aura choisi de faire. La même instance est diffusée à tous les membres de l'équipe partageant une corbeille de travail "WorkList" commune et où chaque activité est associée à un bon de travail "WorkItem". Pour les activités collectives, un compteur d'exécution est associé dont le rôle est de calculer le nombre de fois que ladite activité a été exécutée.

Cette façon de faire est relativement simple à implémenter et vise à promouvoir la coopération dans l'équipe.

Dans la seconde approche, le travail n'est pas partagé à l'avance de sorte à donner plus de liberté aux apprenants dans le choix et la distribution du travail entre eux. Similairement à la première approche, le PAC est diffusé à tous les membres de l'équipe et chaque activité est associée à un "WorkItem" placé dans une "WorkList" partagée. Chaque apprenant devra accomplir son travail selon son propre rythme sans délimitation de la quantité de travail. Les parties de travail collectif demeureront inévitables pour tous les membres de l'équipe et sont d'une grande importance pour le processus d'évaluation.

Par cette façon de faire, nous voudrions augmenter la compétitivité entre les apprenants et ne pas encourager les "bras-cassés" à ne rien faire. Cette approche semble plus élaborée et plus complexe, mais elle offre certains avantages comme celui de donner aux apprenants la responsabilité de trouver un consensus pour un dispatching équitable du travail. Au fait, cela dépendra du degré de compréhension et d'accord entre eux. De plus cela est sensé augmenter l'esprit de compétitivité dans l'équipe.

La table III.2 dresse une comparaison entre les deux approches :

Critère	Première approche	Seconde approche
Equité	Assurée	Dépend du rythme de travail des apprenants dans l'équipe.
Coopération	Favorisée	Dépend de la motivation des apprenants de l'équipe.
Compétitivité	N'apparaît pas car le travail est dispatché à l'avance.	Haute
Communication	Librement utilisée	
Coordination	Assurée par les points de synchronisation dans le processus d'apprentissage collectif.	

Table III.2 : Comparaison entre les deux approches d'implémentation de PAC.

Pour toutes les parties d'un processus, il est important de spécifier un objectif pédagogique. En effet, les objectifs pédagogiques permettent de suivre l'évolution du parcours d'apprentissage et de décider des évaluations appropriées à chaque niveau.

¹⁰ A ne pas confondre avec les activités individuelles d'un PAI.

Comme nous l'avions défini plus haut, nous avons choisi trois niveaux d'objectifs pédagogiques que sont : OPG, OPI et OPO (cf. III.1).

Cependant, le tuteur n'étant pas concerné par ces niveaux, nous avons rajouté l'objectif "TUT" à utiliser en cas d'activités de tutorat. L'attribution des objectifs pédagogiques se fera en fonction du découpage souhaité du processus, il revient au staff pédagogique de le décider. Nous proposons les associations suivantes (Table III.3):

Objectif Pédagogique	Eléments Impliqués	Explications
OPG	Processus	Le processus global doit répondre à un objectif global.
OPI	Sous-Processus	Il constitue un niveau intermédiaire entre un processus global et ses activités effectives.
	Activité	Même une activité pourrait avoir un objectif pédagogique intermédiaire. C'est le cas, lorsque les tâches doivent répondre à des objectifs opérationnels.
OPO	Activité	Lorsqu'une activité à un objectif opérationnel alors il n'est pas nécessaire d'associer un objectif pédagogique aux tâches. Cette propriété sera donc laissée à blanc pour les tâches.
	Tâche	Peut uniquement être de niveau opérationnel ou rien.
TUT	Activité	A utiliser uniquement en cas d'activités concernant le tuteur.

Table III.3 : Association des objectifs pédagogiques aux parties d'un processus de e-learning

Les informations comme les "*DateDebut*" et "*DateFin*" de toute partie d'un processus sont aussi importantes et contribueront à un meilleur déroulement des cas de workflow. Pour les activités et les tâches, nous avons ajouté l'attribut concernant la "*DuréeEstimée*" pour leur accomplissement. Ceci est utile pour l'apprenant que l'on lui définisse le temps moyen dont il a besoin pour accomplir une tâche ou une activité. Cependant, cette information doit être spécifiée uniquement lorsque l'objectif pédagogique est un OPO.

Il est possible aussi de spécifier le type d'un processus, d'un sous-processus ou d'une activité. Au fait, c'est cette propriété qui permet de choisir le niveau de granularité souhaité. La table III.4 propose quelques possibilités mais à titre non exhaustif :

Élément de processus	Types Possibles	Explications
Processus	Module, Matière, Formation.	Le processus global couvre généralement un cursus complet (Formation) ou une partie d'un cursus (Matière ou Module).

Sous-Processus	Module, Matière, Cours, Leçon, TD, TP.	Représente un niveau de détail inférieur au processus. Le choix de la valeur se fera en fonction de celle du processus global et/ou des sous-processus de niveau supérieur ¹¹ .
Activité	Cours, Leçon, TD, TP, Support.	Le même principe prévaut pour les activités. La valeur "Support" sera utilisée lorsqu'il s'agit d'activités du tuteur ou bien d'activités à faire par l'apprenant mais qui ne rentrent pas forcément dans un acte d'apprentissage. Comme exemple d'activités de ce genre : demander à un apprenant de choisir une matière parmi liste de modules affichés à l'écran.

Table III.4 : Types des éléments d'un processus de e-learning

Comme dit plus haut, cette table ne prétend pas présenter tous les cas possibles mais plutôt montre toute la latitude laissée aux concepteurs (staff) pour associer les types aux éléments de processus. Le cahier des charges du dispositif d'enseignement (selon la démarche ASPI) aidera beaucoup à régler ce problème.

Finalement, lorsqu'une activité ou une tâche impliquent un retour ou feedback vers le tuteur, on dira que celle-ci est de **Nature Active**, sinon on la considère de **Nature Passive**. Par exemple l'activité "faire devoir" impliquera forcément l'envoi du devoir vers le tuteur pour correction. Alors que l'activité "lire la leçon" n'impliquera aucun feedback direct.

Au fait, cette propriété est d'une grande importance car c'est à partir des activités d'apprentissage actives que l'on définira les activités que nécessite le tutorat. Une activité est considérée comme active lorsqu'au moins une des tâches la composant l'est.

III.3.3 Aspect comportemental

Cet aspect met en évidence le flux de contrôle intrinsèque à un processus et permet de définir les états des activités et des tâches, leurs conditions d'exécution et le flot d'événement les caractérisant, ainsi que les ressources pédagogiques nécessaires au bon travail de l'apprenant. Nous supposons à tout instant que les ressources pédagogiques nécessaires au travail de l'apprenant sont disponibles. Nous entendons par ressources pédagogiques l'ensemble des outils et supports documents utilisés lors de l'apprentissage¹².

¹¹ Ne perdons pas de vue qu'un sous-processus peut-être lui-même découpé itérativement en d'autres sous-processus.

¹² Les ressources sont représentées par l'aspect informationnel.

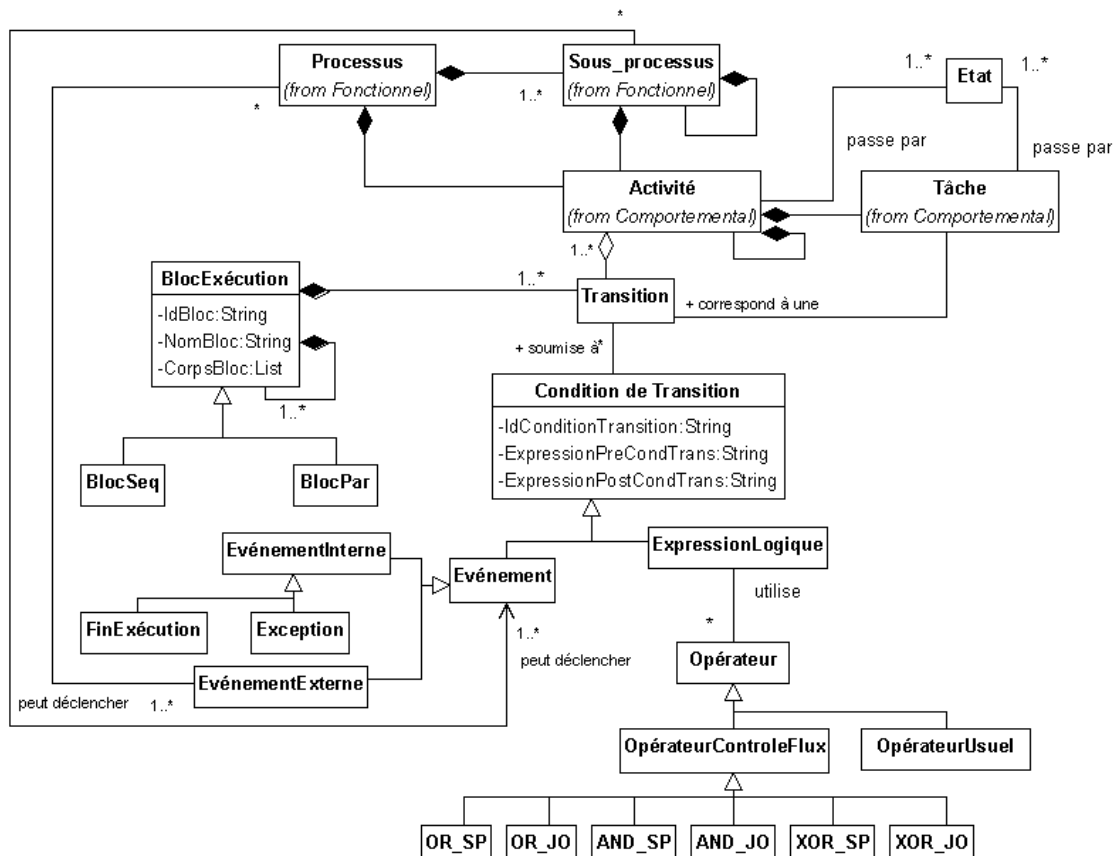


Figure III.7.a : Aspect Comportemental

Vu de près, le processus d'enseignement peut devenir très complexe à cause de la richesse des cours et des leçons aussi bien en termes de contenu qu'en termes d'activités et tâches à exécuter, sans oublier les activités de tutorat qui peuvent en découler. Partant de là, il est possible d'utiliser les réseaux de pétri (Van Der Aalst, 2002) qui offrent l'avantage de permettre de simuler l'exécution des modèles de workflow avant leur déploiement sur une plate forme, évitant ainsi des erreurs dont les conséquences peuvent être néfastes. Cette complexité risque de s'accroître davantage si les processus sont décrits pour un travail en équipe d'apprenants.

Donc, l'aspect comportemental peut être étendu par les éléments susceptibles de supporter une description de processus par les réseaux de Pétri « RdP » (Mahdaoui & al2, 2004). Le concepteur sera donc libre de décrire ses processus soit par les diagrammes d'activités soit par les réseaux de pétri. Dans notre cas, nous utiliserons les diagrammes d'activités étendus d'UML¹³ car l'objectif n'est pas de valider les processus de tutorat en e-learning. La figure III.7.b décrit l'aspect comportemental étendu aux concepts des RdP(s). Comme nous pouvons le constater, cet aspect peut supporter une modélisation des processus par les diagrammes d'activité que par les réseaux de Pétri (RdP(s)).

¹³ <http://www.omg.org>.

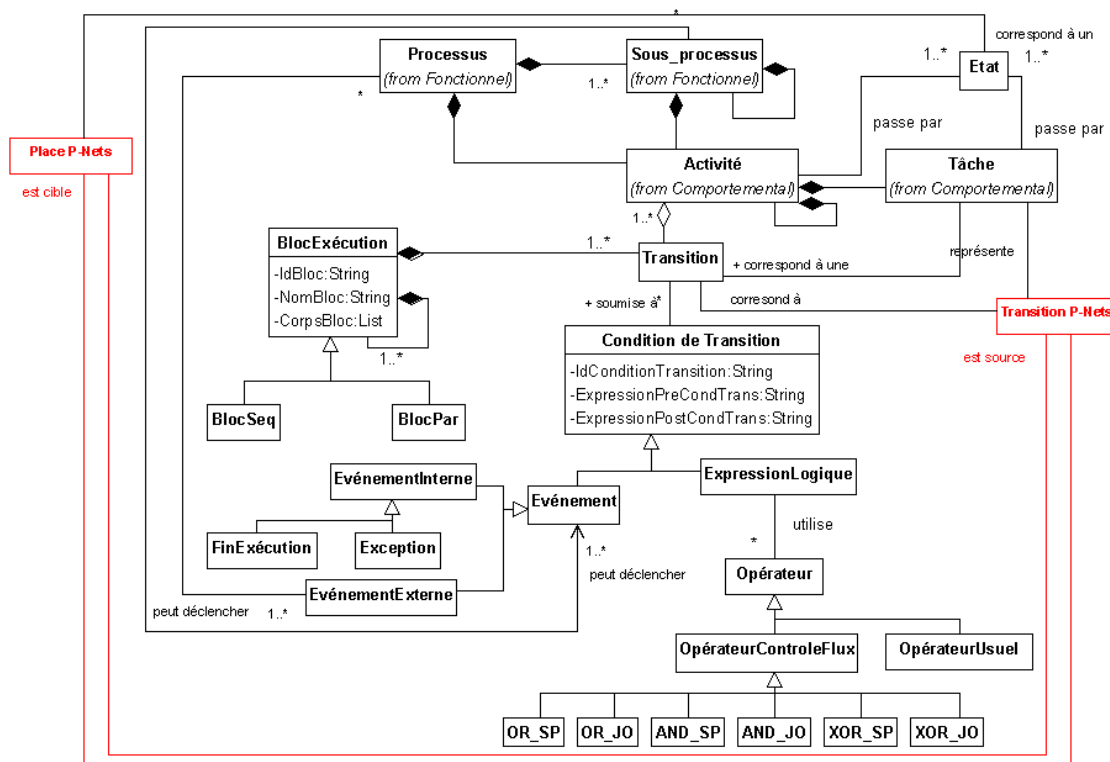


Figure III.7.b : Aspect Comportemental étendu aux Wf-Nets.

Van Der Aalst parle de modèles de Workflow Net « WF-Net » (Van der Aalst, 1997), (Verbeek & al., 2001). Rappelons qu'un WF-Net est un réseau de pétri ayant deux places spéciales « in » et « out ». L'exécution du processus est déclenchée par l'arrivée d'un jeton dans la place « in » et se termine lorsque celui-ci arrive à la place « out ». Une tâche représente une transition de type Wf-Nets "Transition P-Nets" et un état correspond à une place de type Wf-Nets "Place P-Nets". Dans un lien d'une transition P-Nets à une place P-Nets, la première constitue la source et la seconde la cible. En dehors de la capacité du formalisme des RdP(s) à représenter des modèles de processus, ils offrent l'avantage de permettre la vérification de propriétés sur les modèles comme la bornitude, la vivacité, ... etc (Van Der Aalst, 1999). Mais nous ne traiterons pas des problèmes de vérifications car ils sont en dehors du cadre de ce travail.

La description d'un processus commence toujours par un symbole de début noté **Start** et un symbole de fin noté **END**. Un diagramme d'activité spécifiera le comportement global d'un processus ou sous-processus ainsi que les parts de travail impliquant les acteurs "apprenants" et "tuteurs". Un processus sera décrit par un diagramme à couloir où chaque couloir représentera un rôle d'acteur (apprenant ou tuteur). Enfin, les événements et flux de contrôle y seront détaillés ainsi que les interactions "apprenants-tuteur".

III.3.3.1 Etats des activités et des tâches

Dans la figure III.7.a, les transitions expriment des activités et des tâches. Une activité est décrite par une succession de tâches exécutables, donc chaque activité peut-être décomposée en une suite de tâches qui s'exécuteront séquentiellement.

Une activité ou une tâche passe durant son exécution par un ou plusieurs états possibles. Une activité ou une tâche peut être dans l'un des états suivants : *Initiée*, *EnAttente*, *EnCours*, *Bloquée*, *Interrompue* ou *Terminée*. Le diagramme d'état transition suivant montre tous les états par lesquels peut transiter une activité éventuellement une tâche :

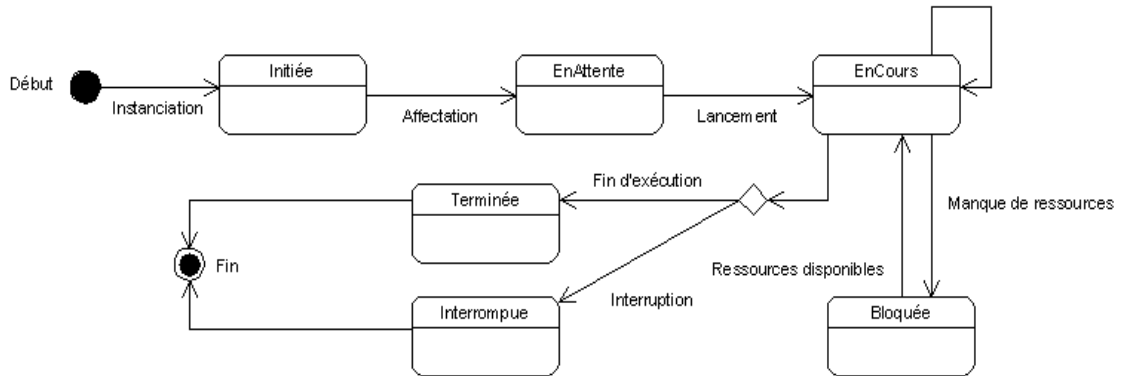


Figure III.8 : Diagramme d'état/transition relatif aux états possibles d'une activité.

- ✿ ***Initiée*** : caractérise une activité ou une tâche qui vient d'être instanciée, mais qui n'est pas encore affectée à un exécutant (apprenant ou tuteur).
- ✿ ***EnAttente*** : caractérise une activité ou une tâche en attente d'être exécutée après son affectation à un exécutant.
- ✿ ***EnCours*** : caractérise une activité ou une tâche en cours d'exécution. Cependant, une activité bouclera dans l'état "*EnCours*" tant que toutes les tâches qui la composent ne sont pas terminées.
- ✿ ***Bloquée*** : caractérise une activité ou une tâche qui a déjà démarré son exécution et se bloque par exemple suite au manque de ressources matérielles dont elle a besoin pour continuer son exécution ou encore parce que l'exécutant s'arrête à ce niveau pour reprendre prochainement (cela dépend de l'état de la tâche en cours d'exécution).
- ✿ ***Terminée*** : caractérise une activité ou une tâche qui a déjà terminée son exécution.
- ✿ ***Interrompue*** : cet état peut avoir lieu dans le cas de la survenue d'une exception par exemple.

Une activité sera considérée comme correctement finie (*Terminée*) si et seulement si toutes les tâches qui la composent le sont. Les tâches représentent les étapes de l'accomplissement d'une activité. Ainsi, une activité sera toujours "*EnCours*" tant que ses tâches ne seront pas toutes accomplies. Cependant, le fait qu'une activité soit

correctement finie ne doit pas s'interpréter par rapport à la qualité des résultats obtenus par le ou les apprenant(s) (cas d'activité individuelle ou collective).

III.3.3.2 Les conditions de transition et les blocs d'exécution

La dynamique d'un workflow se traduit par deux aspects : le déclenchement d'une ou plusieurs activités représenté par la classe "**ConditionDeTransition**" et le contrôle de flux représenté par la classe "**BlocExécution**".

Une condition de transition peut être un événement ou une expression logique. La classe "**Evénement**" représente tout événement émis pour le déclenchement ou lors de l'exécution d'un processus. Un événement peut être externe ou interne.

- ✱ La classe "**EvénementExterne**" : représente les contraintes externes servant au déclenchement des processus workflow. De ce fait, un processus comporte au moins un événement externe qui est l'événement initiateur de la première activité d'un workflow (qui permet de le démarrer). Eventuellement, un sous-processus peut aussi être démarré par un événement externe lorsque ce dernier représente une phase d'un certain enseignement que le tuteur humain initiera.
- ✱ La classe "**EvénementInterne**" : représente tous les événements émis au cours de l'exécution d'un processus. Un événement interne peut être soit une fin d'exécution "**FinExécution**" soit une "**Exception**". Une fin d'exécution signale que la transition (tâche) correspondante a fini son exécution. Une activité sera terminée lorsque toutes ses tâches le seront. Une exception peut arriver durant l'exécution d'une tâche ou d'une activité.

Dans la modélisation workflow, on distingue généralement trois types d'opérateurs booléens : AND, OR et XOR utilisés dans les expressions logiques. Ces opérateurs caractérisent deux grandes familles d'opérateurs : les opérateurs de type "JOIN" que nous terminons par "_JO" et les opérateurs de type "SPLIT" que nous terminons par "_SP". Donc, la classe "**ExpressionLogique**" permet d'exprimer les liens entre les transitions décrivant des relations de précedence (précédence, parallélisme, choix ou itération). Pour le contrôle de flux dans un processus, les opérateurs suivants sont proposés¹⁴ (Table III.5):

Opérateur OR	
<p>OR-SP</p>	<p>Disjonction non exclusive en sortie d'activité :</p> <p>L'exécution de A1 implique l'exécution de A2 ou A3 ou les 2 simultanément. Le choix entre A2 et A3 dépend des données de l'instance de processus.</p>
<p>OR-JO</p>	<p>Disjonction non exclusive en entrée d'activité :</p> <p>L'exécution de A1 ou A2 ou des deux implique l'exécution de A3.</p>
Opérateur AND	

¹⁴ En plus bien sûr de l'opérateur de séquence qui est représenté par une flèche →.

	<p>Jonction en sortie d'activité :</p> <p>L'exécution de A1 sera suivie simultanément par l'exécution de A2 et A3.</p>
	<p>Jonction en entrée d'activité :</p> <p>A3 sera exécutée seulement après l'exécution de A1 et A2.</p>
Opérateur XOR	
	<p>Disjonction exclusive en sortie d'activité :</p> <p>L'exécution de A1 implique exclusivement l'exécution de A2 ou l'exécution de A3. Les conditions C1 et C2 doivent être mutuellement exclusives.</p>
	<p>Jonction exclusive en entrée d'activité :</p> <p>L'exécution de A1 ou A2 exclusivement implique l'exécution de A3. Notons cependant que l'opérateur XOR-JO est similaire à l'opérateur OR-JO.</p>
Structure Itérative	
	<p>A1 peut être exécuté plusieurs fois ou de manière itérative.</p>

Table III.5 : Les opérateurs de contrôle de flux.

Une transition peut être déclenchée par un ou plusieurs événements et être soumise à une condition sous forme d'expression logique. Une expression logique peut utiliser des opérateurs de contrôle de flux (Table III.5) et/ou des opérateurs usuels tel que les opérateurs de comparaison (=, >, <, >=, <=, ≠) ou les connecteurs booléens (AND, OR, NOT). Les opérateurs usuels sont utilisés pour exprimer des conditions supplémentaires reliées au contexte d'une activité particulière. C'est le cas avec l'opérateur XOR_SP.

La classe "*ExpressionLogique*" permet donc de représenter d'une part, le cas où deux ou plusieurs activités sont à l'origine du déclenchement d'une même activité (Pré-condition), et d'autre part, le cas où à la fin de l'exécution d'une activité, il faut une alternative pour poursuivre l'exécution du processus (Post-condition). Afin de mieux illustrer cela, prenons l'exemple suivant qui décrit un processus workflow donné :

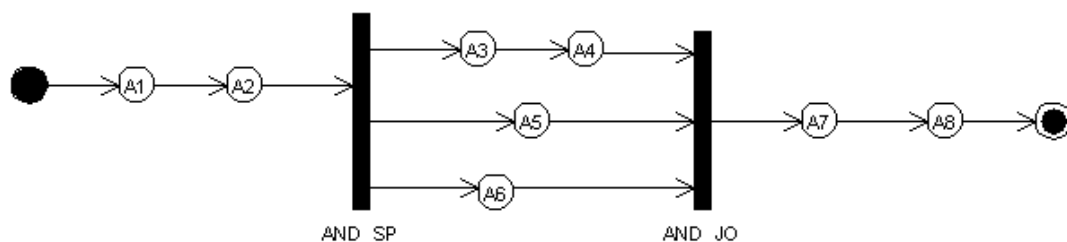


Figure III.9.a : Diagramme d'activité décrivant un processus workflow.

Le premier cas peut être représenté par exemple par la figure III.9.a où les trois activités A4, A5 et A6 du processus sont à l'origine du déclenchement de l'activité A7 du même processus.

Quant au deuxième cas, il peut être représenté à la même figure par la fin de l'activité A2 qui permettra de déclencher en parallèle les activités A3, A5 et A6. Dans ces deux cas, on fera recours à l'usage des expressions logiques pour gérer les différentes combinaisons possibles des activités ou pour réaliser l'alternative avant de passer à l'activité suivante. En supposant que l'événement de fin de chaque activité A_i est noté E_i , les expressions logiques associées à la figure III.9.a sont :

- Pré-condition de A7 : $(E4 \text{ AND_JO } E5 \text{ AND_JO } E6) A7$ qui signifie que l'activité A7 sera lancée à la suite de l'occurrence des événements de fin d'exécution E4, E5 et E6.
- Post-condition de A2 : $E2 (A3 \text{ AND_SP } A5 \text{ AND_SP } A6)$ qui signifie qu'après l'occurrence de l'événement E2, les activités A3, A5 et A6 seront lancées parallèlement.

Un bloc d'exécution "**BlocExécution**" est constitué d'un ensemble d'activités ou de transitions¹⁵ à exécuter. Un bloc d'exécution peut lui même être constitué d'autres blocs d'exécution. Dans ce cas, on prendra le soin de faire la distinction entre le bloc d'exécution principal (ou global) et les blocs d'exécution secondaires (le composant). De plus, un bloc d'exécution peut se spécialiser en :

- Bloc parallèle "**BlocPar**" : qui regroupe des activités parallèles. On dira que deux ou plusieurs activités sont parallèles si elles peuvent être déclenchées et s'exécuter de façon simultanée. Autrement dit, elles sont toutes issues de la même activité. La classe "**BlocPar**" permettra de reconnaître qu'elles activités il faudra lancer en parallèle.
- Bloc séquentiel "**BlocSeq**" : cette classe permet de regrouper des activités séquentielles. Dans un bloc séquentiel, les activités sont ordonnées de façon séquentielle, c'est à dire le déclenchement d'une activité dépend uniquement de celle qui la précède directement dans la séquence chronologique, excepté la première activité du bloc.

Par exemple, soit le diagramme d'activités de la figure III.9.a :

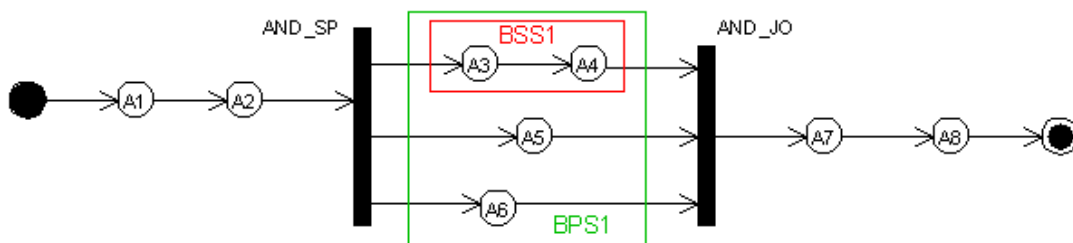


Figure III.9.b : Diagramme d'activité décrivant un processus workflow.

¹⁵ Ne perdons pas de vue qu'une activité est une séquence de tâches, donc de transitions.

Le processus peut être décomposé en trois blocs dont deux sont séquentiels (A1, A2 et A7, A8) et un bloc parallèle (BSP1) contenant lui même un bloc séquentiel (BSS1). La séquence $\boxed{A1}$, $\boxed{A2}$, $\boxed{BSP1}$, $\boxed{A7}$, $\boxed{A8}$ constitue le Bloc Séquentiel Principal (BSP) associé au processus. Toutes les autres séquences de niveau inférieur seront considérées comme des blocs séquentiels ou parallèles secondaires (BSS ou BPS). BPS1 est décrit par la séquence $\boxed{BSS1} // \boxed{A5} // \boxed{A6}$ où BSS1 correspond à la séquence $\boxed{A3}$, $\boxed{A4}$.

III.3.4 Aspect informationnel

Représente la partie système d'information permettant d'extraire les informations nécessaires aux opérations d'apprentissage et de tutorat, ainsi que l'exécution du processus workflow pour le e-learning. Une activité ou une tâche peut nécessiter pour son accomplissement des documents¹⁶. Ces derniers peuvent être des fichiers de type texte, multimédia, formulaires, des questionnaires ou bien des portions de bases de données.

Parmi les concepts importants les notions de **Document Individuel** et **Document Collectif**. Un document individuel peut être produit par un apprenant suite à l'exécution d'une activité ou tâche individuelle. Un document collectif pourra être produit par une équipe suite à l'accomplissement d'une activité collective.

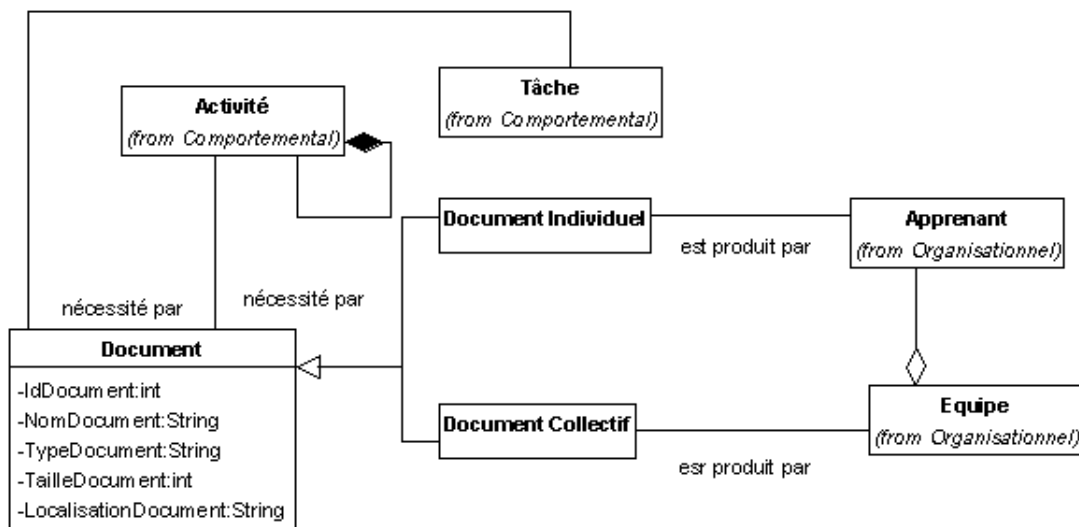


Figure III.10 : Détails de l'aspect Informationnel

La section suivante permettra de mettre en application les différents aspects de ce méta-modèle. Ceci se fera à travers la présentation d'une étude de cas relative à la gestion des travaux pratiques pour les élèves ingénieurs de la deuxième année de la filière informatique. Le cahier des charges a été élaboré selon la démarche ASPI.

¹⁶ Afin de simplifier le problème, on entend par document dans ce travail, toute ressource au sens **électronique du terme** puisque toute ressource est traduisible sous forme d'un fichier avec un certain format et un type donné.

III.4 Présentation d'une étude de cas : Tutorat distant pour un TP du cours "Système d'Information"

III.4.1 Phase 1 : Analyse des besoins

Afin d'expérimenter notre méta-modèle, nous nous sommes intéressés à l'enseignement donné dans le cadre des travaux pratiques de la deuxième année de la filière d'ingénierat en informatique. Pour ce faire, nous avons élaboré le cahier des charges décrivant le dispositif dénommé "TP-SI-01" dont les détails se trouvent en annexe B.

Ce choix ne s'est pas fait sans motivations puisqu'un tel dispositif devrait répondre à des besoins réels pour l'amélioration de la qualité des travaux pratiques dispensés dans le module système d'information.

En effet, avec un programme de cours très riche et un volume de TP en présentiel de une heure et demi par semaine seulement, il est quasiment impossible d'aborder tous les concepts à caractère pratique. Nous pensons qu'une telle activité pédagogique (TP) pourrait être suivie et tutorée à distance en complément de ce qui est déjà dispensé en présentiel. Evidemment, ici n'est pas le contexte de débattre des problèmes logistiques nécessaires à la mise en place d'un tel système.

Voici une brève description du dispositif TP-SI-01 :

Le dispositif doit être inséré dans le cadre du module « Introduction aux Systèmes d'Information » (I.SY.I) prévoyant une série de COURS, TD et TP destinés en particulier aux élèves ingénieurs de la 2^{ème} année informatique. TP-SI-01 est un cours à distance visant à couvrir la partie TP du module.

Le programme de TPSI 01 est prévu en trois périodes de temps devant se dérouler en parallèle avec les séances de cours/TD présentiels. L'apprenant aura à réaliser un ensemble d'activités dans chacune des périodes, en plus d'une séance présentielle hebdomadaire de TP de 1h30 mn prévue pour chaque groupe de TP (auparavant, il n'y avait que cette séance et cela s'est avéré très insuffisant !).

Une fois son login et mot de passe obtenu par l'enseignant du module, l'apprenant peut se connecter au site du TP pour commencer ses travaux.

Les tuteurs du TP évaluent les rapports de TP individuels ou collectifs (par binôme ou trinôme) remis par les apprenants en fin de troisième période et fournissent les feedbacks selon les points prévus à cela dans le planning de l'activité. Les tuteurs procéderont ensuite à l'évaluation des produits logiciels réalisés par les apprenants selon un calendrier présentiel de démonstration prévu en fin d'année.

Les activités sont décrites pour chaque période selon un calendrier précis et chaque activité peut correspondre à plusieurs tâches à exécuter. En fin d'apprentissage l'apprenant est censé acquérir les concepts du domaine abordé ainsi que la maîtrise des outils et procédures permettant de mettre en pratique ses connaissances, c'est à dire,

apprendre à réaliser une application logicielle (Base de données + Traitements) après une étude conceptuelle menée sur un domaine d'étude spécifique.

Dans TP-SI-01, l'apprenant devra apprendre à appliquer les connaissances acquises dans le cours I.SY.I notamment la conception avec la méthode MERISE et fera sa partie pratique avec le langage CLIPPER 5.2. Le choix de ce langage repose sur des critères plutôt pédagogiques car il met bien en évidence les différentes phases d'implémentation d'une base de données.

III.4.2 Phase 2 : Analyse Conceptuelle

A partir du cahier des charges nous avons donc extrait les éléments importants qui vont nous aider à construire nos modèles pour le tutorat de TP à distance. Rappelons toutefois que le cahier des charges couvre un spectre d'information plus large conformément à la démarche ASPI qui est une démarche d'accompagnement, de soutien et de pilotage d'un dispositif d'enseignement basé sur les TICE(s).

Le scénario pédagogique¹⁷ présente un découpage périodique de l'enseignement qui s'étale sur trois périodes. Chaque période porte sur un sujet particulier, présente un ensemble d'objectifs à atteindre¹⁸ à travers des activités d'apprentissage à accomplir par l'apprenant comme le récapitule le tableau suivant :

Période (Du -- Au --)	Activités d'Apprentissage	Brève Description	A accomplir
Du 15/10/2006 Au 15/12/2006	A1	Etudier et Découvrir ¹⁹ .	Individuellement
	A2	Effectuer une première prise en main du langage CLIPPER 5.2.	
	A3	Travailler avec le NG ²⁰ .	
Du 16/12/2006 Au 31/01/2007	A1	Manipuler du code CLIPPER 5.2.	Individuellement
	A2	Ecrire des programmes en CLIPPER 5.2.	
Du 01/02/2007 Au 25/05/2007	A1	Se préparer au travail de TP.	Collectivement
	A2	Effectuer le TP.	
	A3	Préparer la démonstration de TP.	

Table III.6 : Répartition des activités d'apprentissage par période

D'autre part, l'analyse du scénario pédagogique a permis de recenser les activités de tutorat que devront assurer les assistants de TP (Table III.7) :

¹⁷ Voir la section 3 de l'annexe B.

¹⁸ Les détails sur les sujets abordés et les objectifs se trouvent en Annexe B.

¹⁹ Conformément à la taxonomie de Blum, il faut veiller à associer des verbes actifs aux descriptions des activités et des tâches. Cela a pour effet de mieux guider l'apprenant et de lui faciliter la compréhension de ce qu'il a à faire comme travail.

²⁰ Nantucket Guide, c'est l'aide du langage CLIPPER.

Période (Du -- Au --)	Activités de Tutorat pour	Brève Description
Du 15/10/2006 Au 15/12/2006	A1	Lecture du rapport individuel. (A4)
	A2	Correction des exercices. (A5)
	A3	Correction du devoir. (A6)
Du 16/12/2006 Au 31/01/2007	A1	Il n'y a rien à faire.
	A2	Evaluer les travaux programmés. (A3)
Du 01/02/2007 Au 25/05/2007	A1	Elaborer et envoyer le FeedBack. (A4)
	A2	Evaluer le rapport de TP. (A5)
	A3	Il n'y a rien à faire.

Table III.7 : Répartition des activités de tutorat par période

L'analyse de la section relative aux rôles et fonctions que doivent assurer les acteurs du dispositif TP-SI-01 permet de recenser différents rôles dont celui de l'apprenant et du tuteur qui nous intéressent dans ce cas. Le digramme de cas d'utilisation suivant montre les associations entre les activités et les rôles impliqués :

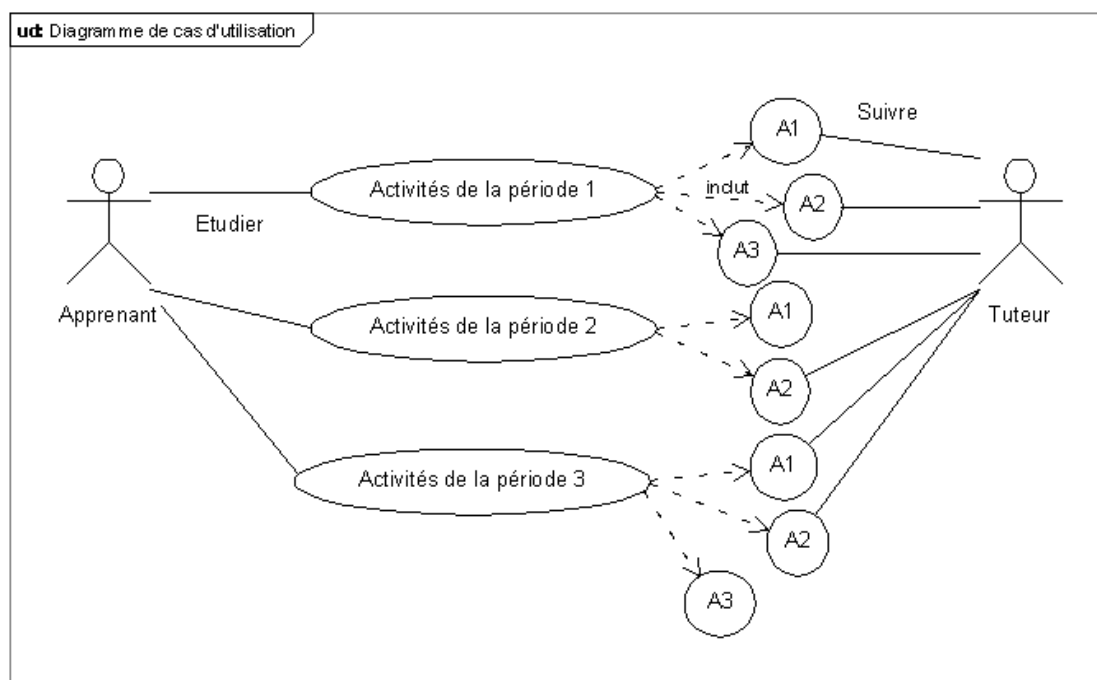


Figure III.11 : Association des rôles aux activités

Comme le met en évidence le diagramme de cas d'utilisation ci-dessus (figure III.11), le tuteur doit intervenir au niveau de cinq activités parmi les huit activités dégagées du cahier des charges.

Afin de mettre en évidence la séquence d'interaction qui aura lieu entre le tuteur et les apprenants, nous utilisons les diagrammes de séquence comme le montre la figure (III.12) ci-après :

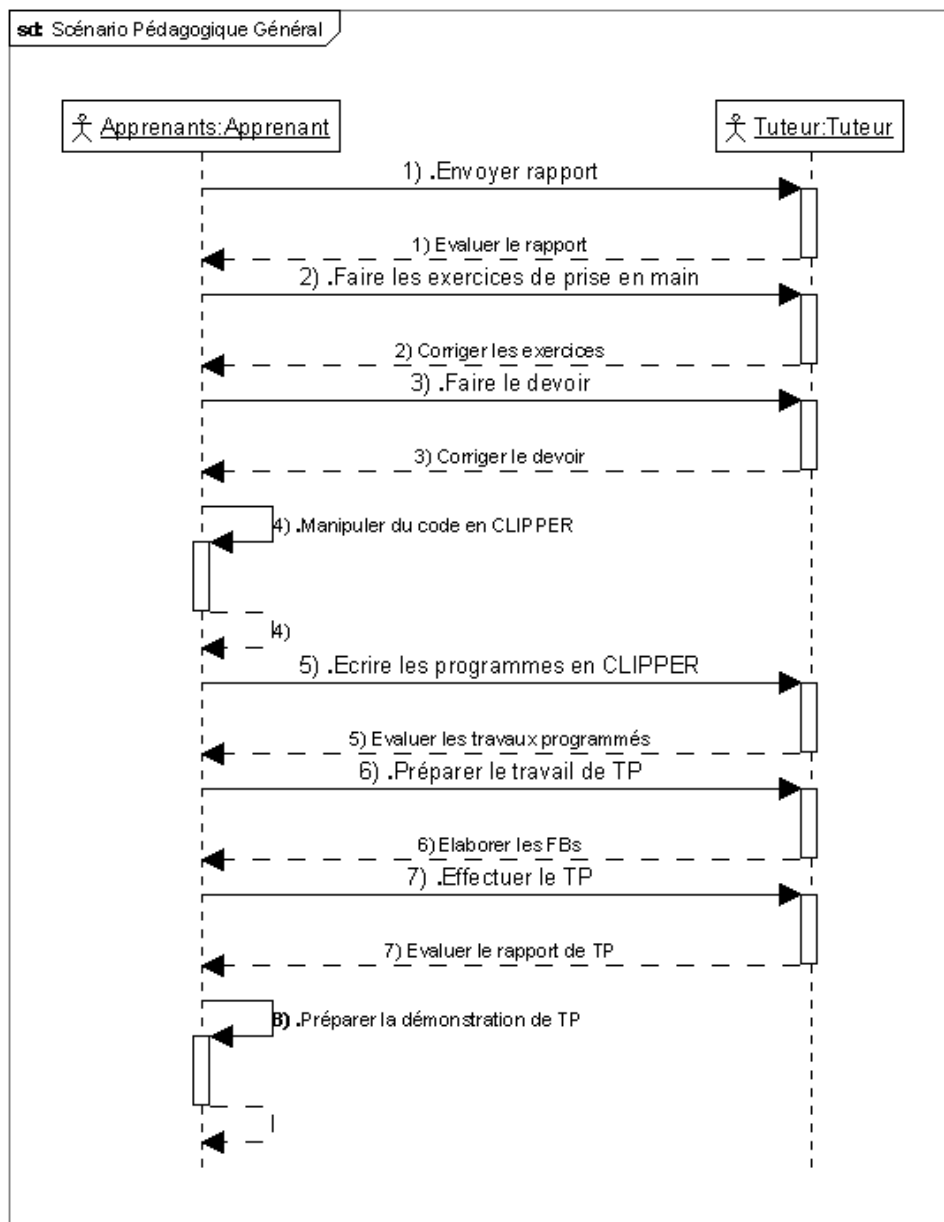


Figure III.12 : Scénario d’interaction durant les phases d'apprentissage.

L’analyse de la section relative aux ressources pédagogiques dans le cahier des charges nous a permis d’extraire les ressources nécessaires pour le bon accomplissement des activités comme suit :

Période	Activité	Ressources
P1	<i>Activité 1</i>	○ Version d’évaluation d’un logiciel à expérimenter (logiciel compressé au format ZIP).
	<i>Activité 2</i>	○ Support de cours du langage CLIPPER. (format pdf). ○ Le langage CLIPPER, logiciel compressé au format « ZIP ».
	<i>Activité 3</i>	○ Grille du format des réponses. (format word ou pdf)
P2	<i>Activité 1</i>	○ Manipulations #01, #02, #03, #04 et #05. (format word ou pdf)

	<i>Activité 2</i>	○ Série d'exercices à programmer. (format word ou pdf)
P3	<i>Activité 1</i>	○ Sujet de TP à réaliser. (format word ou pdf)
	<i>Activité 2</i>	○ Prototype du rapport de TP. (format word ou pdf)
	<i>Activité 3</i>	○ Explication du procédé de préparation des disquettes ou CD-ROM pour le jour de la démonstration. (format word ou pdf).

Table III.8 : Ressource pédagogiques nécessaires par activité et par période.

Comme nous venons de le voir, l'analyse du cahier des charges nous a permis de dégager les éléments pertinents pour la construction du modèle de tutorat distant. Les diagrammes de cas d'utilisation ont permis de montrer l'implication des acteurs dans les activités pédagogiques chacun selon son rôle.

Les diagrammes de séquence ont mis en évidence le déroulement des échanges entre apprenant et tuteur de manière générale. La phase de conception montrera plus en détail la nature des échanges avec les apprenants qui peut se faire individuellement ou collectivement.

L'extraction des éléments nécessaires à la modélisation s'est faite en prenant en compte les quatre aspects à considérer dans la construction d'un modèle, à savoir, l'organisationnel, le fonctionnel, le comportemental et l'informationnel.

A – Pour l'Aspect Organisationnel

Pour les acteurs humains engagés dans ce processus, on considère une famille formée de 20 étudiants (apprenants). La famille est divisée en deux groupes de 10 étudiants chacune. Enfin, les équipes de binômes ou trinômes seront formées au sein d'une même famille mais pas forcément dans le même groupe. Dans ce cas d'étude, nous considérerons qu'il y a un seul assistant de TP qui jouera le rôle de tuteur.

B – Pour l'Aspect fonctionnel

Le processus global correspondra au dispositif "TP-SI-01". Il sera découpé en trois sous processus représentant chacun les activités relatives à une période donnée comme imposé par le cahier des charges. Les deux premiers processus sont de type "PAI" alors que le troisième sera de type PAC. Chaque sous processus comprend un ensemble d'activités décomposables elles mêmes en tâches élémentaires comme le récapitule le tableau suivant :

Processus	Sous-Processus	Type	Activités Apprenants	Tâches Apprenants	Activités Tuteur	Tâches Tuteur
TP-SI-01	SP1	PAI	A1	T1, T2, T3, T4	A4	T1, T2
			A2	T1, T2, T3	A5	T1
			A3	T1, T2, T3	A6	T1, T2
	SP2	PAI	A1	T1, T2, T3, T4	∅	∅
			A2	T1, T2	A3	T1, T2, T3
	SP3	PAC	A1	T1, T2, T3	A4	T1
			A2	T1, T2, T3	A5	T1, T2
			A3	T1	∅	∅

Table III.9 : Découpage du processus relatif au dispositif "TP-SI-01".

Dans ce qui suit, nous donnerons le détail des tâches pour les apprenants et les tuteurs. Pour une meilleure codification, nous attacherons les sous-processus à leur processus principal, les activités à leur processus et les tâches aux activités à partir desquelles elles découlent (codification hiérarchique). Cela a pour avantage de faciliter le repérage d'une activité ou d'une tâche dans son contexte.

Par exemple, la tâche T.2.2.1 (T.<n° du sous-processus>.<n° de l'activité>.<n° de la tâche>), signifie qu'il s'agit de la tâche 1 relative à l'activité A2 du sous-processus 2 (SP2) (à lire de droite à gauche). Voici les tâches d'apprentissage (pour les apprenants) :

Tâche	Brève Description²¹	Objectif pédagogique	Nature²²
T.1.1.1	☛ Télécharger le logiciel d'évaluation.	∅	passive
T.1.1.2	☛ Explorer le logiciel après son installation.	OPO	passive
T.1.1.3	☛ Discuter sur le forum.	∅	passive
T.1.1.4	☛ Rédiger un rapport individuel.	∅	active
T.1.2.1	☛ Télécharger le manuel du langage CLIPPER.	∅	passive
T.1.2.2	☛ Télécharger et installer CLIPPER.	OPO	passive
T.1.2.3	☛ Faire les exercices et les envoyer.	OPO	active
T.1.3.1	☛ Travailler avec le NG "Norton Guide".	OPO	passive
T.1.3.2	☛ Expliquer des instructions par le NG.	OPO	passive
T.1.3.3	☛ Faire le devoir et le remettre.	OPO	active
T.2.1.1	☛ Effectuer la manip #01.	OPO	passive
T.2.1.2	☛ Trouver l'erreur dans la manip #02.	OPO	passive
T.2.1.3	☛ Trouver l'erreur dans la manip #03.	OPO	passive
T.2.1.4	☛ Trouver l'erreur dans la manip #04.	OPO	passive
T.2.2.1	☛ Faire les 05 exercices proposés.	∅	passive
T.2.2.2	☛ Rendre code sources et exécutables.	∅	active
T.3.1.1	☛ Lire le TP et discuter à travers le forum.	OPO	passive
T.3.1.2	☛ Partagez-vous le travail.	∅	passive
T.3.1.3	☛ Envoyez vos impressions.	∅	active
T.3.2.1	☛ Programmer les traitements du TP.	OPO	passive
T.3.2.2	☛ Rédiger le rapport de TP.	OPO	passive
T.3.2.3	☛ Rendre le rapport avant la date limite.	∅	active
T.3.3.1	☛ Préparez votre démonstration de TP.	OPO	passive

Table III.10 : Tâches d'apprentissage du processus "TP-SI-01".

Notons que les tâches actives constitueront les points par lesquels le tuteur interagira avec ses apprenants. Cet aspect sera mis en évidence dans l'aspect comportemental.

²¹ La description textuelle détaillée se trouve en annexe B (cahier des charges).

²² Rappelons que lorsqu'une tâche se termine par un renvoi vers le tuteur, on dira qu'elle est "active" sinon elle est considérée comme "passive".

Les tâches de tutorat (pour les tuteurs) :

Tâche	Brève Description	Nature
T.1.4.1	☛ Evaluer le rapport individuel.	passive
T.1.4.2	☛ Préparer le FB et l'envoyer.	active
T.1.5.1	☛ Envoyer le corrigé des exercices.	active
T.1.6.1	☛ Corriger le devoir.	passive
T.1.6.2	☛ Envoyer la note et les observations.	active
T.2.3.1	☛ Tester les exécutables.	passive
T.2.3.2	☛ Voir les codes sources.	passive
T.2.3.3	☛ Noter et envoyer les résultats.	active
T.3.4.1	☛ Envoyer FB et orientations.	active
T.3.5.1	☛ Evaluer le rapport.	passive
T.3.5.2	☛ Envoyer le FB sur le rapport.	active

Table III.11 : Tâches de tutorat dans le processus "TP-SI-01".

Comme nous pouvons le constater, il n'a pas été utile de préciser l'objectif pédagogique pour le tuteur étant donné que toutes les tâches auront la même mention "TUT" (voir le méta-modèle). Les tables suivantes donnent les valeurs des attributs qui seront utilisées lors de la création des différents éléments de l'aspect fonctionnel du modèle de "TP-SI-01".

Processus	Valeurs des attributs ²³						
	①	②	③	④	⑤	⑥	⑦
P.1	TP-SI-01	Travaux pratiques du module 'Système d'Information'	15/10/2006	25/07/2007	Système d'Information	TP	OPG

Table III.12 : Détails du processus global P.1.

Sous-Processus	Valeurs des attributs							
	①	②	③	④	⑤	⑥	⑦	⑧
P.1.1	CLIPPER	Découverte du langage CLIPPER	15/10/2006	15/12/2006	ILP	TP	OPI	P.1
P.1.2	Exercices	Exercices de manipulation du langage CLIPPER	16/12/2006	31/01/2007				
P.1.3	TP-CLIPPER	Réalisation du TP avec CLIPPER	01/02/2007	25/05/2007				

²³ ①: Nom court; ②: Description; ③: Date de début; ④: Date de fin; ⑤: Thème; ⑥: Type; ⑦: Objectif pédagogique.

Table III.13 : Détails des sous processus du processus P.1.

Activité	Valeurs des attributs ²⁴							
	①	②	③ ²⁵	④	⑤	⑥	⑦	⑧
A.1.1	Etude	Etudier et Découvrir le langage CLIPPER	10 j	TP	P.1.1	OPO	Active	Ø
A.1.2	1° Tour	Effectuer une première prise en main du langage CLIPPER 5.2	20 j					
A.1.3	Le N.G	Travailler avec le Nantucket Guide	15 j					
A.1.4	Lecture	Lecture du rapport individuel	1 j	SUPPO RT	P.1.1	TUT	Active	Ø
A.1.5	Corriger	Correction des exercices	2 j					
A.1.6	Evaluer	Evaluer le devoir	1 j	TP	P.1.2	OPO	Passive	Ø
A.2.1	Manip(s)	Manipuler du code CLIPPER	7 j					
A.2.2	Programmer	Ecrire des programmes en CLIPPER 5.2	25 j					
A.2.3	Evaluer	Evaluer les travaux programmés	10 j	SUPPORT	P.1.2	TUT	Active	Ø
A.3.1	Préparation	Se préparer au travail de TP	10 j	TP	P.1.3	OPO	Active	Collective
A.3.2	Le TP	Effectuer le TP	90 j					
A.3.3	La Démo.	Préparer la Démonstration	7 j					
A.3.4	FB	Elaborer et envoyer le FB	2 j	SUPPORT	P.1.3	TUT	Active	Individuelle
A.3.5	Rapport	Evaluer le rapport de TP	3 j					

Table III.14 : Détails des activités du processus P.1.

Tâche	Valeurs des attributs ²⁶				
	①	②	③	④	⑤ ²⁷

²⁴ ③: *Durée Estimée*; ④: *Etat*; ⑤: *Type*; ⑥: *Sous processus*; ⑦: *Objectif pédagogique*; ⑧: *Nature*; ⑨: *Catégorie (Individuelle ou Collective)*. A préciser uniquement en cas de PAC. Les documents nécessaires seront spécifiés au niveau des tâches.

²⁵ Il est possible de spécifier la durée selon l'unité qui convient : minute, heure, jours, ... etc. Pour le tuteur, ceci n'est pas obligatoire car étant celui qui suit les instances de processus, il doit veiller au respect des délais pour son travail. Néanmoins, il est utile de lui rappeler les délais.

²⁶ ①: *Nom court*; ②: *Description*; ③: *Objectif pédagogique*; ④: *Nature*; ⑤: *Document*;

²⁷ Lorsqu'un même document est nécessaire à plusieurs tâches, il sera attaché uniquement à la première tâche de la séquence. Ceci est logique car un document peut subir des transformations d'une tâche à une autre. Si une tâche nécessite plusieurs documents, on "zipera" les fichiers en un seul.

Proposition d'une Infrastructure Statique et Dynamique pour le Support du Travail Collaboratif : « Application au tutorat dans le e-learning »

T.1.1.1	Téléchargement1	Télécharger le logiciel d'évaluation.	Ø	passive	D1
T.1.1.2	Exploration	Explorer le logiciel après son installation.	OPO	passive	Ø
T.1.1.3	Discussion	Discuter sur le forum.	Ø	passive	Ø
T.1.1.4	Rédaction	Rédiger un rapport individuel.	Ø	active	D2
T.1.2.1	Téléchargement2	Télécharger le manuel du langage CLIPPER.	Ø	passive	D3
T.1.2.2	Téléchargement3	Télécharger et installer CLIPPER.	OPO	passive	D4
T.1.2.3	Exercices	Faire les exercices et les envoyer.	OPO	active	D5
T.1.3.1	Guide	Travailler avec le NG "Norton Guide".	OPO	passive	D6
T.1.3.2	Langage	Expliquer des instructions par le NG.	OPO	passive	D7
T.1.3.3	Devoir	Faire le devoir et le remettre.	OPO	active	D8
T.1.4.1	Evaluation	Evaluer le rapport individuel.	Ø	passive	D9
T.1.4.2	FB	Préparer le FB et l'envoyer.	Ø	active	Ø
T.1.5.1	Corrigé Type	Envoyer le corrigé des exercices.	Ø	active	D10
T.1.6.1	Correction	Corriger le devoir.	Ø	passive	D11
T.1.6.2	Résultats	Envoyer la note et les observations.	Ø	active	Ø
T.2.1.1	Manip1	Effectuer la manip #01.	OPO	passive	D12
T.2.1.2	Manip2	Trouver l'erreur dans la manip #02.	OPO	passive	D13
T.2.1.3	Manip3	Trouver l'erreur dans la manip #03.	OPO	passive	D14
T.2.1.4	Manip4	Trouver l'erreur dans la manip #04.	OPO	passive	D15
T.2.2.1	Exercices	Faire les 05 exercices proposés.	Ø	passive	D16
T.2.2.2	A Remettre	Rendre code sources et exécutables.	Ø	active	Ø
T.2.3.1	Tests	Tester les exécutables.	Ø	passive	D17
T.2.3.2	Voir	Voir les codes sources.	Ø	passive	Ø
T.2.3.3	Résultats	Noter et envoyer les résultats.	Ø	active	Ø
T.3.1.1	Lecture TP	Lire le TP et discuter à travers le forum.	OPO	passive	D18
T.3.1.2	Partage	Partagez-vous le travail.	Ø	passive	Ø
T.3.1.3	Rapport	Envoyez vos impressions.	Ø	active	Ø
T.3.2.1	Programmation	Programmer les traitements du TP.	OPO	passive	D19
T.3.2.2	Rédaction	Rédiger le rapport de TP.	OPO	passive	Ø
T.3.2.3	Remise	Rendre le rapport avant la date limite.	Ø	active	Ø

T.3.3.1	Préparation	Préparez votre démonstration de TP.	OPO	passive	D20
T.3.4.1	FB	Envoyer FB et orientations.	Ø	active	D21
T.3.5.1	Evaluation	Evaluer le rapport.	Ø	passive	D22
T.3.5.2	FB/Rapport	Envoyer le FB sur le rapport.	Ø	active	Ø

Table III.15 : Détails des tâches du processus P.1.

C – Pour l'Aspect Informationnel

Les tables suivantes décrivent toutes les ressources ou documents²⁸ nécessaires pour l'accomplissement des tâches et/ou des activités. La table III.16 décrit l'ensemble des ressources nécessaires aux activités d'apprentissage et document à produire :

Activités d'apprentissage	Nom de la Ressource	Tâches concernées	Document à produire	Type
A.1.1	Demo.zip	T.1.1.1 T.1.1.2	Rapport1.doc	Ind. ²⁹
	ModèleRapport.dot	T.1.1.4		
A.1.2	ManuelCLIPPER.pdf	T.1.2.1 T.1.2.3	Solutions.doc	Ind.
	CLIPPER.zip	T.1.2.2 T.1.2.3		
	Exercices.doc	T.1.2.3		
A.1.3	InstructionsNG.pdf	T.1.3.1	Devoir.doc	Ind.
	AnnexeSupport.pdf	T.1.3.2		
	GrilleRéponse.doc	T.1.3.3		
A.2.1	Manip01.pdf	T.2.1.1	Ø	Ø
	Manip02.pdf	T.2.1.2		
	Manip03.pdf	T.2.1.3		
	Manip04.pdf	T.2.1.4		
A.2.2	SérieExercices.pdf	T.2.2.1	SérieSol.zip	Ind.
A.3.1	SujetTP.pdf	T.3.1.1	Impressions.doc	Col. ³⁰
A.3.2	PrototypeRapportTP.dot	T.3.2.1	RapportTP.doc	Col.
A.3.3	Explications.pdf	T.3.3.1	Ø	Ø

Table III.16 : Ressources nécessaires pour l'apprentissage.

Ressources nécessaires aux activités de tutorat et document à envoyer par le tuteur à ses apprenants :

²⁸ Dans ce travail, nous avons utilisé indifféremment les termes de ressource ou de document au sens électronique du terme puisque toute ressource est traduite sous forme d'un fichier sous un format et un type donné.

²⁹ Document Individuel.

³⁰ Document Collectif.

Activités de Tutorat	Nom de la ressource	Tâches concernées	Document à rendre	Type
A.1.4	Rapport1.doc	T.1.4.1 T.1.4.2	FeedBack.pdf	Ind.
A.1.5	Solutions.doc	T.1.5.1	Corrigés.pdf	Ind.
A.1.6	Devoir.doc	T.1.6.1 T.1.6.2	RésultatsDevoir.pdf	Ind.
A.2.3	SérieSol.zip	T.2.3.1 T.2.3.2 T.2.3.3	RésultatsSérie.pdf	Ind.
A.3.4	Impressions.pdf	T.3.4.1	Orientations.pdf	Col.
A.3.5	RapportTP.doc	T.3.5.1	FeedBackTP.pdf	Col.

Table III.17 : Ressources nécessaires pour le tutorat.

Le tableau suivant résume l'ensemble des ressources documentaires à créer en les codifiant dans le système :

Id. Doc.	Valeurs des attributs ³¹			
	①	② ³²	③	④
D1	Demo.zip	WinZip.exe	ftp://Tutorat.ressources.edu/zip	Ind.
D2	ModèleRapport.dot	WinWord.exe	ftp://Tutorat.ressources.edu/doc	Ind.
D3	ManuelCLIPPER.pdf	AcroRead.exe	ftp://Tutorat.ressources.edu/pdf	Ind.
D4	CLIPPER.zip	WinZip.exe	ftp://Tutorat.ressources.edu/zip	Ind.
D5	Exercices.doc	WinWord.exe	ftp://Tutorat.ressources.edu/doc	Ind.
D6	InstructionsNG.pdf	AcroRead.exe	ftp://Tutorat.ressources.edu/pdf	Ind.
D7	AnnexeSupport.pdf	AcroRead.exe	ftp://Tutorat.ressources.edu/pdf	Ind.
D8	GrilleRéponse.doc	WinWord.exe	ftp://Tutorat.ressources.edu/doc	Ind.
D9	Rapport1.doc	WinWord.exe	ftp://Tutorat.ressources.edu/doc	Ind.
D10	Solutions.doc	WinWord.exe	ftp://Tutorat.ressources.edu/doc	Ind.
D11	Devoir.doc	WinWord.exe	ftp://Tutorat.ressources.edu/doc	Ind.
D12	Manip01.pdf	AcroRead.exe	ftp://Tutorat.ressources.edu/pdf	Ind.
D13	Manip02.pdf	AcroRead.exe	ftp://Tutorat.ressources.edu/pdf	Ind.
D14	Manip03.pdf	AcroRead.exe	ftp://Tutorat.ressources.edu/pdf	Ind.
D15	Manip04.pdf	AcroRead.exe	ftp://Tutorat.ressources.edu/pdf	Ind.
D16	SérieExercices.pdf	AcroRead.exe	ftp://Tutorat.ressources.edu/pdf	Ind.
D17	SérieSol.zip	WinZip.exe	ftp://Tutorat.ressources.edu/zip	Ind.
D18	SujetTP.pdf	AcroRead.exe	ftp://Tutorat.ressources.edu/pdf	Ind.
D19	PrototypeRapportTP.dot	WinWord.exe	ftp://Tutorat.ressources.edu/doc	Ind.
D20	Explications.pdf	AcroRead.exe	ftp://Tutorat.ressources.edu/pdf	Ind.
D21	Impressions.pdf	AcroRead.exe	ftp://Tutorat.ressources.edu/pdf	Col.

³¹ Dans cet exemple, on ne précisera pas la taille des fichiers puisque le système peut être programmé pour vérifier les contraintes de taille. Dans la réalité, il est plus recommandable de le faire.

³² On suppose que toutes les ressources sont stockées dans un même serveur mais qu'elles sont classées dans des dossiers.

D22	RapportTP.doc	WinWord.exe	ftp://Tutorat.ressources.edu/doc	Col.
-----	---------------	-------------	----------------------------------	------

Table III.18 : Ressources documentaires pour l'apprentissage et le tutorat.

D – Pour l'Aspect Comportemental

Enfin, dans cet aspect nous mettrons en évidence la dynamique des interactions entre un tuteur et ses apprenants. Nous considérerons donc un processus global composé de trois phases (selon les périodes décrites dans le cahier des charges) où sous-processus devant s'exécuter successivement. Les deux premières phases sont individuelles (PAI) et la troisième est collective (PAC). Les diagrammes d'activités suivants décrivent respectivement les sous processus SP1, SP2 et SP3 :

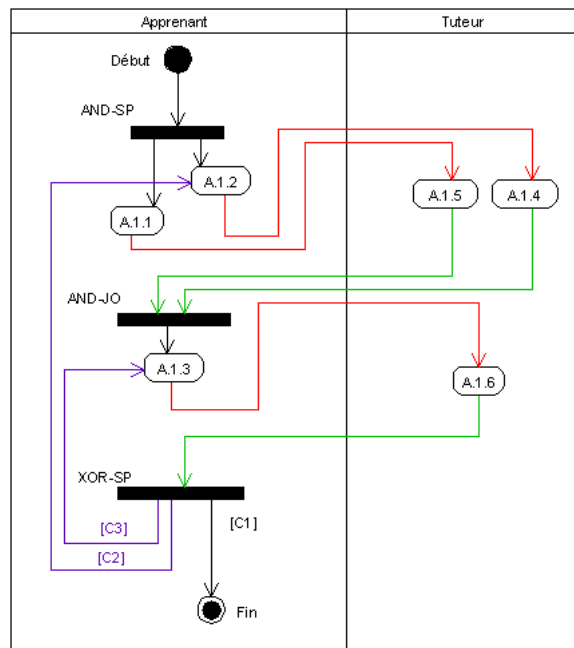


Figure III.13 : Sous-processus "SP1".

Comme nous pouvons le constater, les activités d'apprentissage A.1.1 et A.1.2 sont indépendantes et peuvent donc être exécutées selon l'ordre qui conviendra le mieux aux apprenants. Il en découle que le tuteur exécutera selon le cas et de façon indépendante les activités de tutorat A.1.4 et A.1.5. Evidemment, cette façon de faire n'est pas obligatoire mais cela permettra d'être moins contraignant pour les apprenants sur le plan du temps d'apprentissage. Ainsi, un apprenant gèrera lui même le temps à impartir à chaque activité selon ses aptitudes.

Notons aussi que le diagramme d'activités peut refléter des décisions tutorielles qui peuvent ne pas avoir été décrites en détail dans le cahier des charges. Par exemple, au niveau de la troisième activité d'apprentissage et qui consiste à faire un devoir, le tuteur peut très bien recommander que l'apprenant refasse une certaine partie du sous processus ou bien le tout selon les résultats obtenus au devoir comme le montre le schéma. De cette manière, on forcera l'apprenant à reprendre des parties jugées critiques par le tuteur. Il s'agit alors de décrire un parcours d'apprentissage guidé par les résultats de l'apprenant. Cependant, rien ne force à ignorer cette contrainte et à se suffire des remarques et recommandations que l'apprenant recevra.

Par conséquent, les choix découlant du XOR_SP après l'activité A.1.3 dépendent des résultats de l'activité A.1.6 et dans laquelle le tuteur peut prendre des décisions en fonction des résultats des activités précédentes. Les conditions expriment généralement des résultats d'évaluation sous forme de notes. Par exemple, la condition [Bon résultats] signifie que toutes les notes sont supérieures ou égales à 12 (NoteRapport \geq 12 AND NoteExercices \geq 12 et NoteDevoir \geq 12). De plus, toutes les conditions sont sensées être exclusives, ce qui ne permettra de déclencher qu'une et une seule parmi les alternatives possibles.

D'autres facteurs tels que la limite des délais pour les activités d'apprentissage ou encore la complexité de la description des processus peuvent influencer sur le degré de détail à adopter pour le tutorat. Il est aussi possible de décrire le sous processus en allant dans les détails jusqu'au niveau de la tâche, chose que nous ne ferons que pour ce premier sous processus, la démarche étant similaire pour le reste :

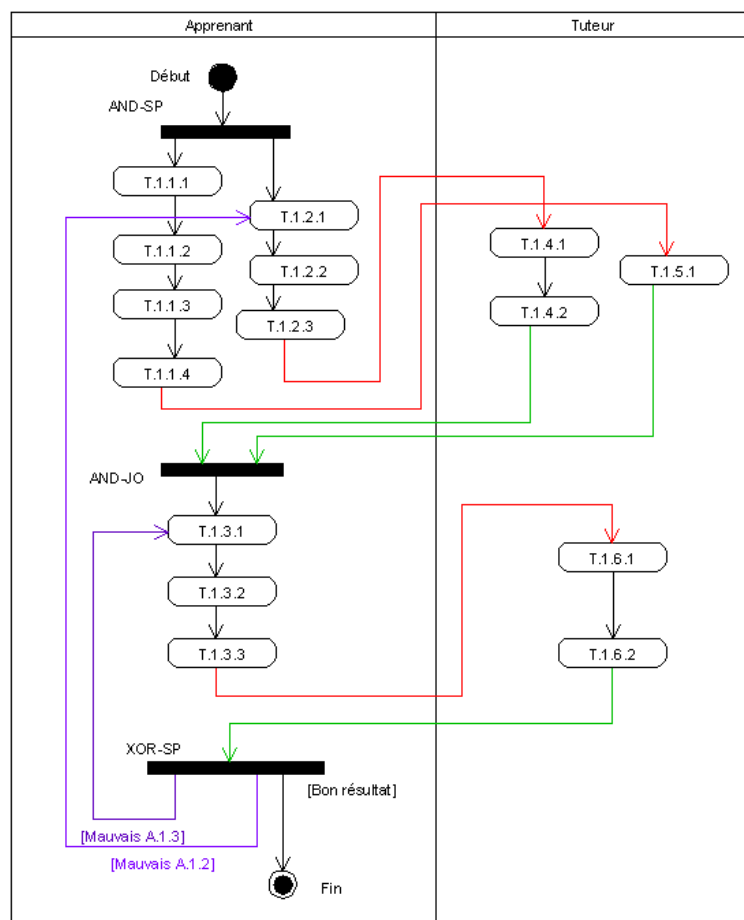


Figure III.14 : Sous-processus "SP1" détaillé en tâches élémentaires.

La seconde phase constitue un processus très simple comme le montre la figure suivante où les activités sont à accomplir séquentiellement :

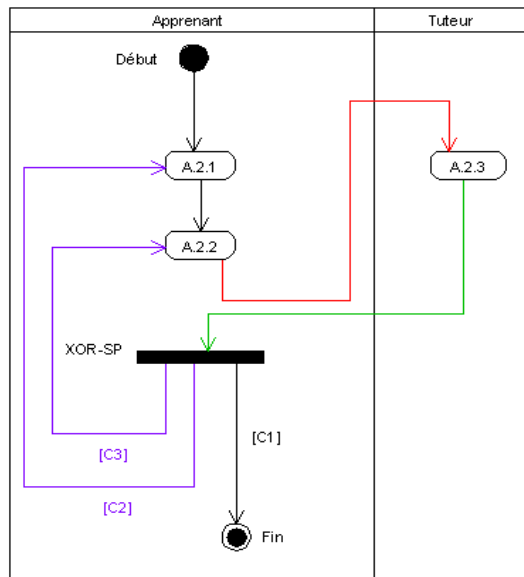


Figure III.15 : Sous-processus "SP2".

Là encore, il est utile d'orienter l'apprenant quant aux activités à reprendre. Ici, il s'agira de refaire les manips ou de revoir le code programmé.

SP3 est un sous processus d'apprentissage collectif (figure III.16) où chaque équipe est formée de deux ou trois apprenants (binôme ou trinôme). Nous avons expliqué dans la section qui présentait le méta-modèle que la répartition des tâches aux apprenants pouvait se faire de deux manières possibles : par répartition anticipée des activités ou par compétition entre les apprenants. Dans ce cas d'étude, la répartition du travail est laissée aux apprenants (seconde approche). Cela semble être plus réaliste par rapport à ce que font les étudiants dans la réalisation de leurs travaux pratiques.

La table suivante montre le mode d'exécution des activités d'apprentissage de la phase collective :

Activité	Description	Type	Acteur(s) Impliqué(s)	Mode
A.3.1	Se préparer au travail de TP.	Collective	Tous	Asynch. ³³
T.3.1.1	Lire le TP et discuter à travers le forum.			
T.3.1.2	Partager le travail.			
T.3.1.3	Envoyer vos impressions.			
A.3.2	Effectuer le TP.	Collective	Tous	Asynch.
T.3.2.1	Programmer les traitements du TP.			
T.3.2.2	Rédiger le rapport de TP.			
T.3.2.3	Envoyer le rapport.			
A.3.3	Préparer la démonstration.	Collective	Tous	Synch. ³⁴
T.3.3.1	Préparer la démonstration.			

Table III.19 : Mode d'exécution des activités collectives.

³³ Dans le mode Asynchrone, les mem

³⁴ Dans ce mode, tous les membres doivent être connectés en même temps.

L'activité A.3.3 est cruciale pour cette étape d'où elle doit être encadrée par le tuteur qui doit signaler les défaillances du rapport de TP.

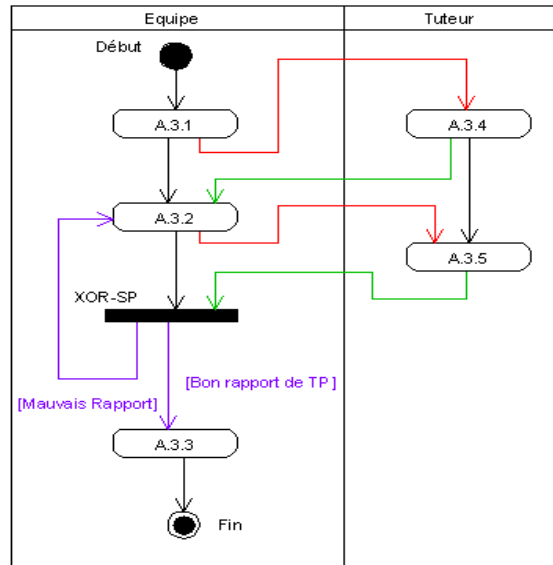


Figure III.16 : Sous-processus "SP3".

Enfin, il est à souligner que le modèle de processus pourrait prendre en charge toutes les situations ou scénarios pédagogiques possibles. Mais cela risquerait de compliquer inutilement le modèle. Les modèles ci-dessus ont présenté un scénario de déroulement normal ou prévisible des activités. Apparaît alors l'intérêt de la flexibilité grâce à l'utilisation de méta-modèle.

En effet, rien n'empêchera un apprenant de solliciter de l'aide à un niveau non prévu à l'avance par le tuteur. Dans ce cas le tuteur traitera le cas directement sur l'instance concernée. Si le même point fait l'objet de demande d'assistance par différentes instances (apprenant) alors la décision de modifier le modèle lui même peut être prise. On appliquera alors la stratégie appropriée quant à la population des instances qui sera concernée par la modification. Ceci n'entrant pas dans l'objet de cette thèse, nous soulignons quand même que ce travail constitue actuellement un de nos axes d'intérêt et une première approche basée sur les systèmes décisionnels se trouve dans (Selmoune & al, 2006).

A titre d'exemple, supposons qu'au niveau de l'activité A.2.1, l'apprenant souhaite avoir de l'aide à propos des manips à effectuer. Cela pourra donner lieu au déroulement suivant :

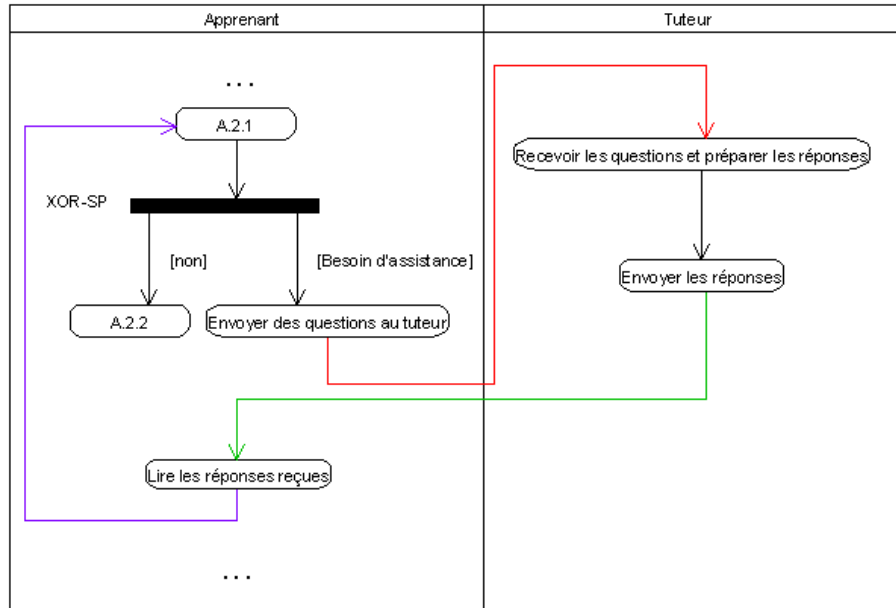


Figure III.17 : Demande d'assistance par un apprenant.

Rappelons qu'une tâche est caractérisée par un ou plusieurs états et qu'une activité sera considérée comme accomplie si toutes les tâches qui la composent le sont. Par exemple, voici les états par lesquels passera l'activité A.1.1 du sous processus SP1 (figure III.14).

Comme il a été clarifié lors de la présentation du méta-modèle, les diagrammes d'activité permettent de spécifier les opérateurs, les événements et les conditions auxquelles sont soumises les différentes activités d'un sous-processus. Les blocs d'exécution permettent de contrôler le flux, nous allons donc procéder à la décomposition des sous processus en blocs d'exécution et associer les expressions logiques nécessaires au déclenchement des tâches comme le montrent respectivement les figures suivantes (III.18, III.20 et III.22):

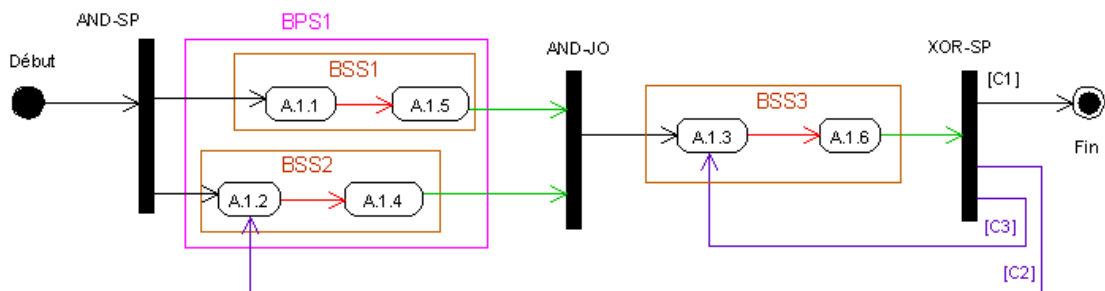


Figure III.18 : Les blocs d'exécution de "SP1".

On en déduit les blocs d'exécution détaillés jusqu'au niveau tâche suivants :

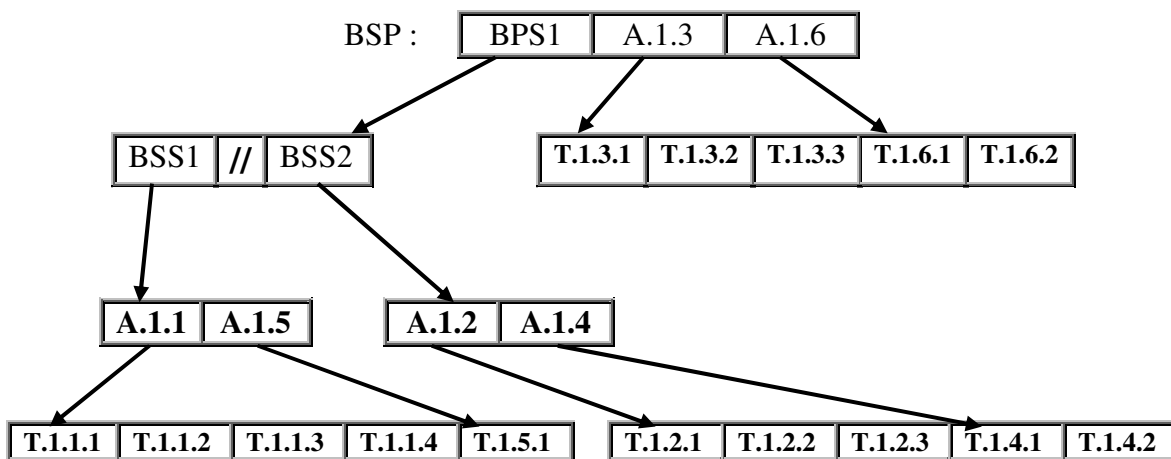


Figure III.19 : Décomposition de "SP1" par blocs d'exécution

Quant aux expressions logiques associées à SP1, nous avons³⁵ :

Après l'événement externe de début d'exécution Start() de P1 (donc SP1), on associera la post-condition " Start() (A.1.1 AND_SP A.1.2) " ou plus exactement, " Start() (T.1.1.1 AND_SP T.1.2.1) " car ne l'oublions pas, une activité est une suite de tâches ou transitions séquentielles. Donc les autres tâches de chaque activité suivront séquentiellement. Dans tout ce qui reste, nous écrirons les expressions logiques en fonction des tâches telles qu'elles seront représentées dans le système.

L'activité A.1.3 (T.1.3.1) comprend la pré-condition suivante " (E.1.4.1 AND_JO E.1.5.1) T.1.3.1".

La fin de l'activité A.1.6 comporte la post-condition suivante : " E.1.6 ([C1] Stop) XOR_SP [C2] A.1.2 XOR_SP [C3] A.1.3". Les conditions :

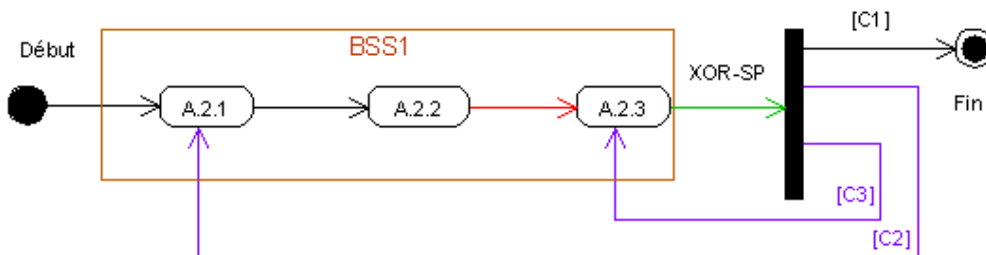


Figure III.20 : Les blocs d'exécution de "SP2".

Les blocs d'exécution détaillés jusqu'au niveau tâche :

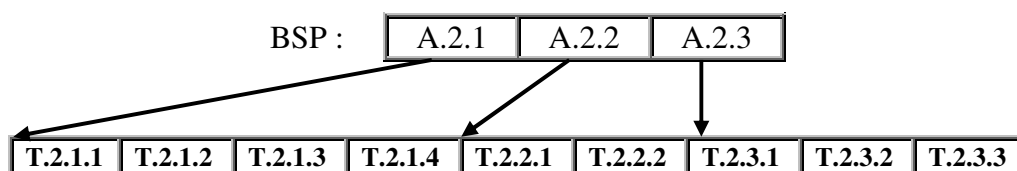


Figure III.21 : Décomposition de "SP2" par blocs.

Expressions logiques associées à SP2 :

³⁵ Tout événement de fin d'exécution d'une activité sera noté par "E" suivi du numéro d'activité. Le même principe sera utilisé pour les tâches.

L'activité A.2.3 comprend une post-condition "E.2.3 ([C1] Stop() XOR_SP [C2] A.2.1 XOR_SP [C3] A.2.3)".

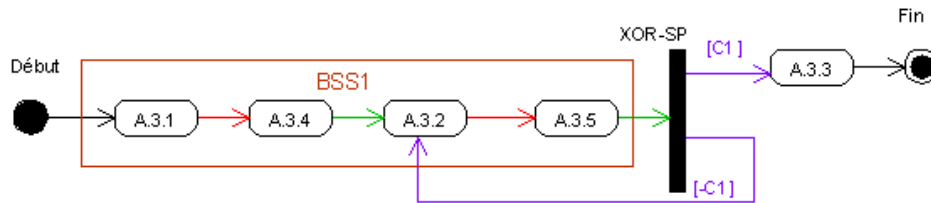


Figure III.22 : Les blocs d'exécution de "SP3".

Les blocs d'exécution détaillés jusqu'au niveau tâche :

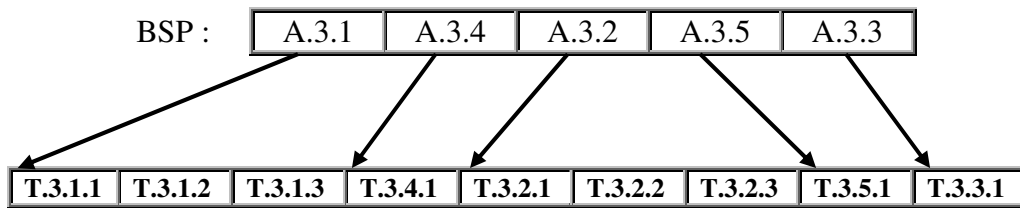


Figure III.23 : Décomposition de "SP3" par blocs.

Expressions logiques associées à SP3 :

L'activité A.3.5 comprend une post-condition " E.3.5 ([C1] A.3.3 XOR_SP [-C1] A.3.2) ".

III.4.3 Phase 3 : Construction des modèles de "TP-SI-01"

Durant cette phase, nous allons construire l'ensemble des tables nécessaires couvrant les quatre aspects du méta-modèle. Ensuite, les modèles de processus seront générés à l'aide du langage BPEL4WS. Certains travaux comme (Gardner, 2003) et (Mantell, 2005) décrivent comment convertir un diagramme d'activité en processus BPEL comme le montre la table III.20 :

Élément UML	Concept BPEL4WS
« process » class	BPEL process definition
Activity graph on a « process » class	BPEL activity hierarchy
« port » associations	BPEL partner declarations
« process » class attributes	BPEL containers or variables
Hierarchical structure and control flow	BPEL sequence and flow activities
Decision nodes	BPEL switch activities and transition conditions
« receive », « reply », « invoke » activities	BPEL receive, reply, invoke activities
« protocol » package with « role » classes	BPEL service links types and roles

Table III.20 : Mapping de UML vers BPEL (Gardner, 2003) & (Mantell, 2005).

Pour décrire un processus de tutorat en BPEL, nous utilisons ces règles de transformation (table III.21) :

Concept du Méta-modèle	Concept BPEL
Rôle	<PartnerLink>
Sous-Processus	<Process>
Description d'Activité	<Sequence>, <Scope>
Tâche Passive	<Receive>
Tâche Active	<Invoke>
Opérateurs	<Flow>
Bloc Parallèle	<Flow>
Bloc Séquentiel	<Sequence>

Table III.21 : Transformation des concepts du méta-modèle vers BPEL

Conformément à cela, le listing III.1 suivant décrit un fragment de code représentant la description en BPEL du sous processus SP2 représenté en figure III.20 :

```

//-----
//Process SP.1.2
//-----
<process name="SP12" ...>

//-----
//Actors/roles or Partner Links
//-----
<PartnerLinks>
  //Learner
  <partnerLink name="Learner"
  partnerLinkType="client:SP12"
  myRole="SP12Provider"
  partnerRole="SP12Requester"/>
  //Tutor
  <partnerLink name="Tutor"
  partnerLinkType="client:SP12"
  myRole="SP12Provider"
  partnerRole="SP12Requester"/>
</PartnerLinks>

//-----
//Variables
//-----
<variables>
...
</variables>
...
//-----
//Activities and control flow
//-----
<sequence name="SP.1.2">
  <sequence name="SSB1">
    <scope name="A.1.2.1">
      <sequence name="Manipuler Du Code Clipper">
        <receive name="Effectuer la Manip #01"
        createInstance="yes" partnerLink="Learner"
        portType="client:SP12"
        operation="initiate"/>
        ...
      </sequence>
    </scope>
  </sequence>
</sequence>

```

```
</sequence> //End Of "Manipuler du code Clipper"  
</scope> //End of "A.1.2.1"  
  
//Make Some Variable Assignations  
<assign name="Assign_1" ... />  
  
<scope name="A.1.2.2">  
  <sequence name="Ecrire des programmes en Clipper">  
    <receive name="Faire les exercices proposés"  
      createInstance="no" partnerLink="Tutor"  
      portType="client:SP12" operation="initiate"/>  
    <invoke name="Rendre les solutions et exécutables"  
      partnerLink="Tutor"  
      portType="client:SP12Callback"  
      operation="onResult"/>  
  </sequence> //End of "Ecrire des programmes en Clipper"  
</scope> //End of "A.1.2.2"  
  
...  
  
</sequence> //End of SSB1  
  
</sequence> //End of PSB1 (Primary Sequential Bloc)  
  
</process> //End of the Sub_Process 1.2
```

Listing III.1 : Description du sous processus SP2 en BPEL

Comme nous pouvons le remarquer, le processus global représentant le dispositif global "TP-SI-01" correspondra à un ensemble de processus BPEL représentant chacun un sous processus. Les tâches sont décrites à l'aide de primitives BPEL où les tâches actives sont traduites vers la primitive <Invoke>. <Invoke> est utilisée car les processus sont asynchrones. Les activités du sous processus SP2 sont considérées comme une simple séquence de primitives BPEL. La primitive <Assign> est utilisée pour transmettre des variables et des résultats d'une activité (au sens méta-modèle) ou <Scope> (au sens BPEL) vers l'autre.

III.4.4 Phase 4 : Instanciation des modèles de "TP-SI-01"

Cette phase consistera à générer le modèle de processus à l'aide de l'outil e-tutoring conçu à cet effet en créant les tables nécessaires dans la base de données représentant les quatre aspects. Le chapitre V présentera quelques outils réalisés dans le cadre de ce projet de thèse.

III.5 Fonctionnement global du système proposé

La section précédente nous a permis de mettre en application les différents aspects du méta-modèle pour le tutorat en e-learning proposé et ce à travers une étude de cas complète. Il nous paraît opportun maintenant de nous pencher sur un autre aspect relié au fonctionnement global du système proposé afin d'essayer de répondre à la question suivante : *Comment et Qui va utiliser ce système ?*.

La réponse à cette question nécessite de passer en revue un certain ensemble d'éléments liés principalement au domaine abordé et aux acteurs y intervenant. Pour cela, nous avons procédé à l'analyse et la conception du système en se basant sur un noyau de besoins recensé et jugés comme critiques pour tout système de workflow.

III.5.1 Quels sont les fonctionnalités du système ?

Le tuteur est un acteur principal du système car c'est à lui qu'incombera le rôle d'initier et de suivre les processus de tutorat distant. Cependant, *est-il réaliste de penser qu'il sera capable à lui seul de prendre en charge toutes les tâches liées à la création des modèles ?*. Nous l'avions précisé précédemment, l'élaboration des processus de tutorat distant est un travail d'équipe dont le tuteur fait partie. Mais dans l'équipe, d'autres rôles peuvent être intégrés comme par exemple celui de "Constructeur de modèles" dont la mission sera de créer les modèles après leur conception grâce à l'outil approprié.

Ce dernier pourrait être par exemple un enseignant formé spécialement à cet effet, chose qui nous semble la plus efficace. Au fait, l'ajout d'autres rôles permettra de catégoriser les problèmes à gérer aussi. Ainsi, un tuteur s'occupera de tutorer, l'apprenant d'apprendre, le constructeur de modèles des aspects d'implémentation et l'administrateur de tous les problèmes reliés à la gestion du système. Toutefois, rien n'empêchera qu'un tuteur accomplisse d'autres rôles si vraiment les processus ne sont pas de grande envergure.

En fonction de cela, nous avons recensé les principaux besoins suivants :

- ✿ Définir un processus de tutorat distant.
- ✿ Exécuter un processus de tutorat distant.
- ✿ Instancier un modèle de processus.
- ✿ Superviser un processus.

Le diagramme de cas d'utilisation suivant illustre les principales fonctionnalités du système ainsi que les rôles devant intervenir.

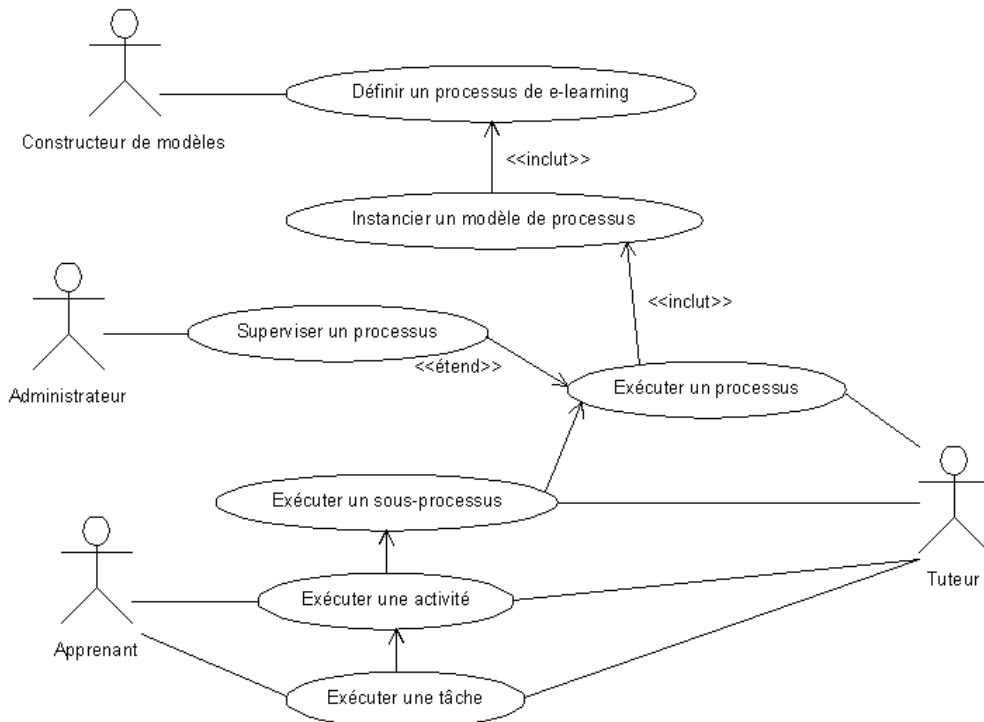


Figure III.24 : Fonctionnalités du système de tutorat distant

Dans ce diagramme, quatre types de rôles sont à distinguer :

- ❖ Le **tuteur** : il peut solliciter l'exécution d'un processus workflow qui lui sera instancié par l'administrateur du système. Comme le tuteur a charge de suivre des groupes d'apprenants, il lui revient de lancer l'exécution des sous-processus d'un processus donné et d'affecter à ses apprenants les activités d'apprentissage. D'un autre côté, il aura lui-même à exécuter ses propres activités et tâches de tutorat.
- ❖ L'**apprenant** : constitue un facteur qui contribue à l'évolution du processus de tutorat distant puisque c'est en fonction de ses activités que celles du tuteur sont définies. Il aura donc à accomplir les activités d'apprentissage.
- ❖ L'**administrateur** : c'est la personne chargée de la supervision de l'exécution d'une instance de processus. On entend par supervision d'un processus, l'instanciation des modèles de processus, l'affectation des rôles et des droits d'accès aux différents acteurs du système, la visualisation de la carte d'instance, la consultation de la liste des activités et des instances, ... etc, pour des raisons de maintenance par exemple.
- ❖ Le **constructeur de modèles** : c'est une personne chargée d'implémenter les modèles de processus définis et conçus par le staff pédagogique.

III.5.2 Les scénarios du système de tutorat distant

Notre démarche étant guidée par les cas d'utilisation, nous allons documenter les différents cas d'utilisation par les scénarios correspondants. Ces scénarios seront relatés à l'aide des diagrammes d'activités et des diagrammes de séquence d'UML.

De plus, l'objectif étant d'expliciter le fonctionnement du système, nous épargnerons les détails relatifs aux situations exceptionnelles sauf si le cas le nécessite vraiment.

1 – Définition d'un processus :

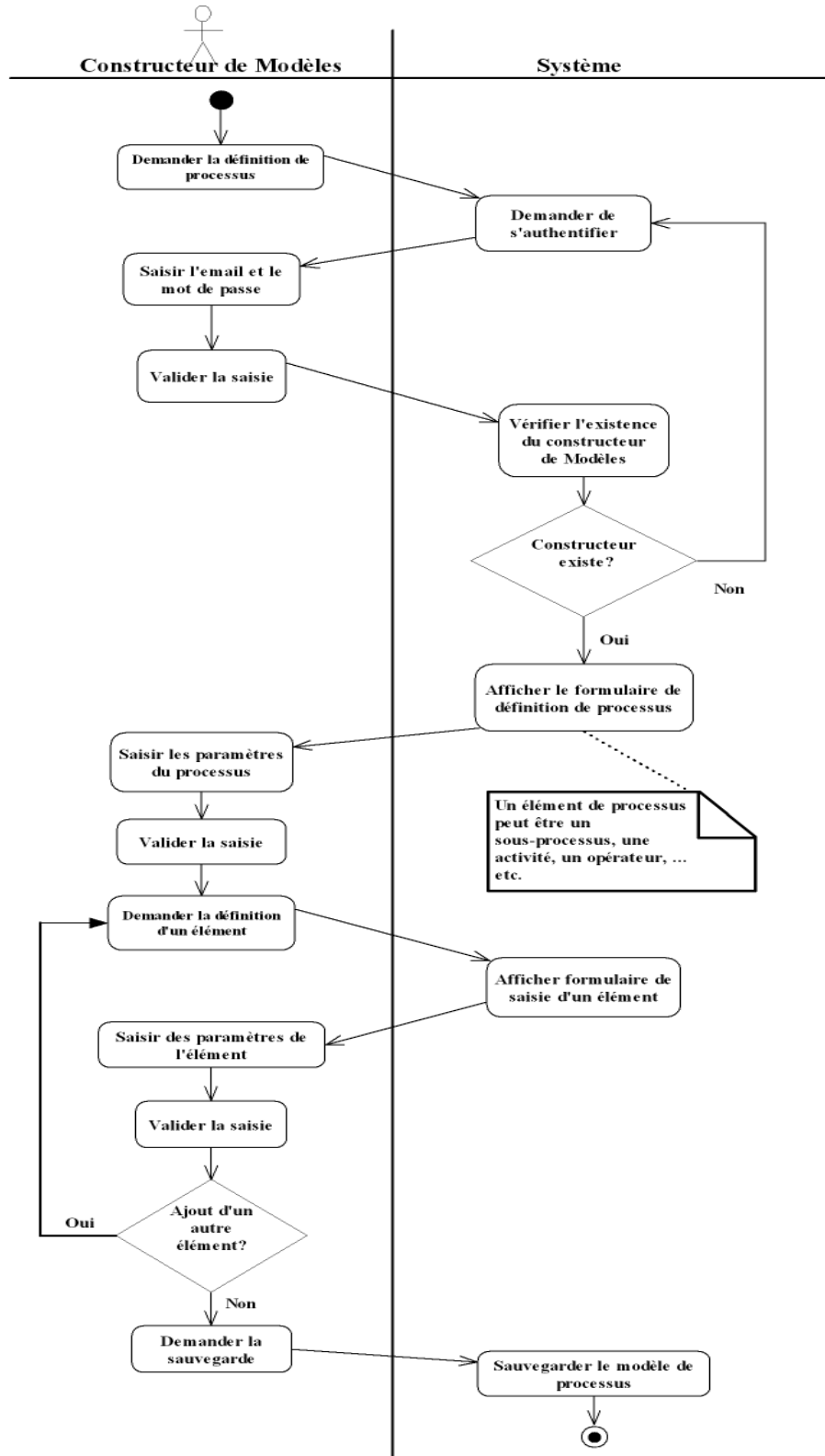


Figure III.25 : Diagramme d'activités du scénario « Définir un processus »

Commentaire : le constructeur de modèle reçoit un document établi par le staff pédagogique qui détaille tous les aspects du modèle de processus à construire. Ce dernier accède au système logiciel lui permettant d'implémenter le modèle et de le stocker dans les bases de données nécessaires.

Le diagramme de séquence correspondant est illustré par la figure 4.3 suivante :

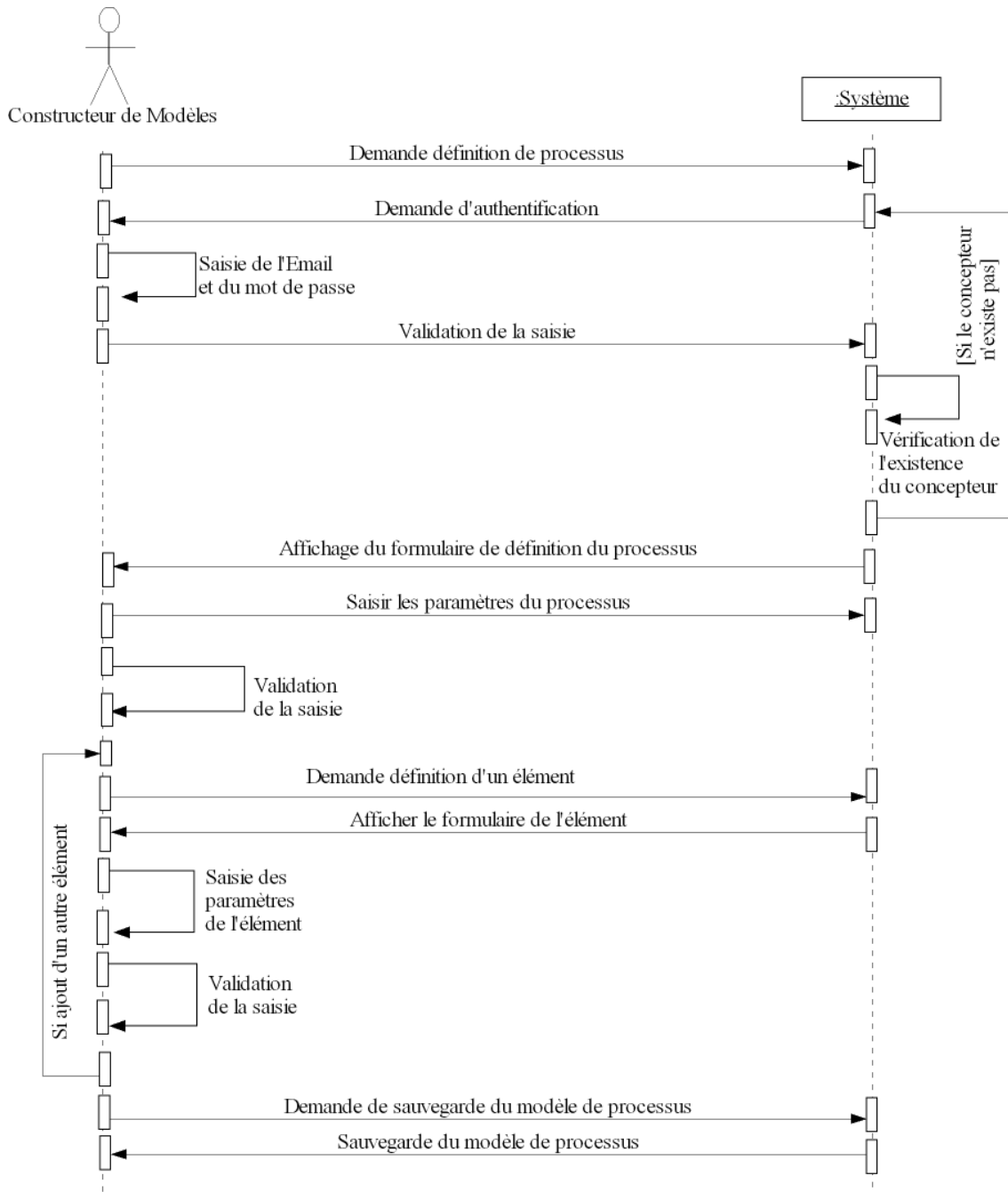


Figure III.26 : Diagramme de séquence illustrant les interactions du scénario « Définir un processus »

2 - Instancier un processus :

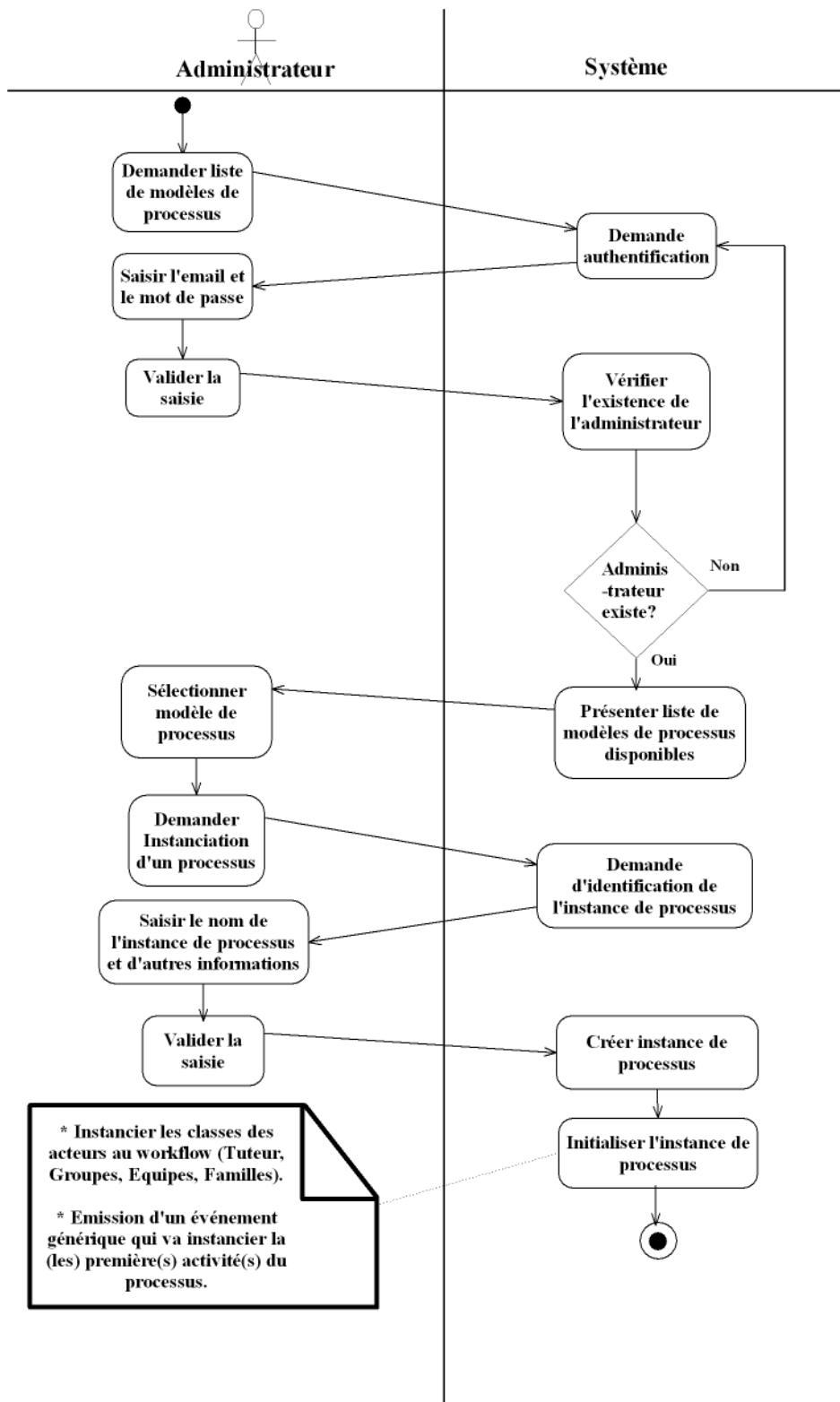


Figure III.27 : Diagramme d'activité du cas « Instancier un processus »

Commentaire : l'administrateur reçoit une demande du tuteur pour la création d'une instance de processus d'un modèle précédemment construit et existant dans la base de données des

processus. La demande peut être formulée par écrit ou par e-mail avec toutes les informations nécessaires. L'instance créée sera mise à la disposition du tuteur à qui reviendra la charge de suivre le restant des étapes comme nous le verrons plus loin.

Le diagramme de séquence correspondant au scénario "Instancier un processus" est présenté par la figure suivante :

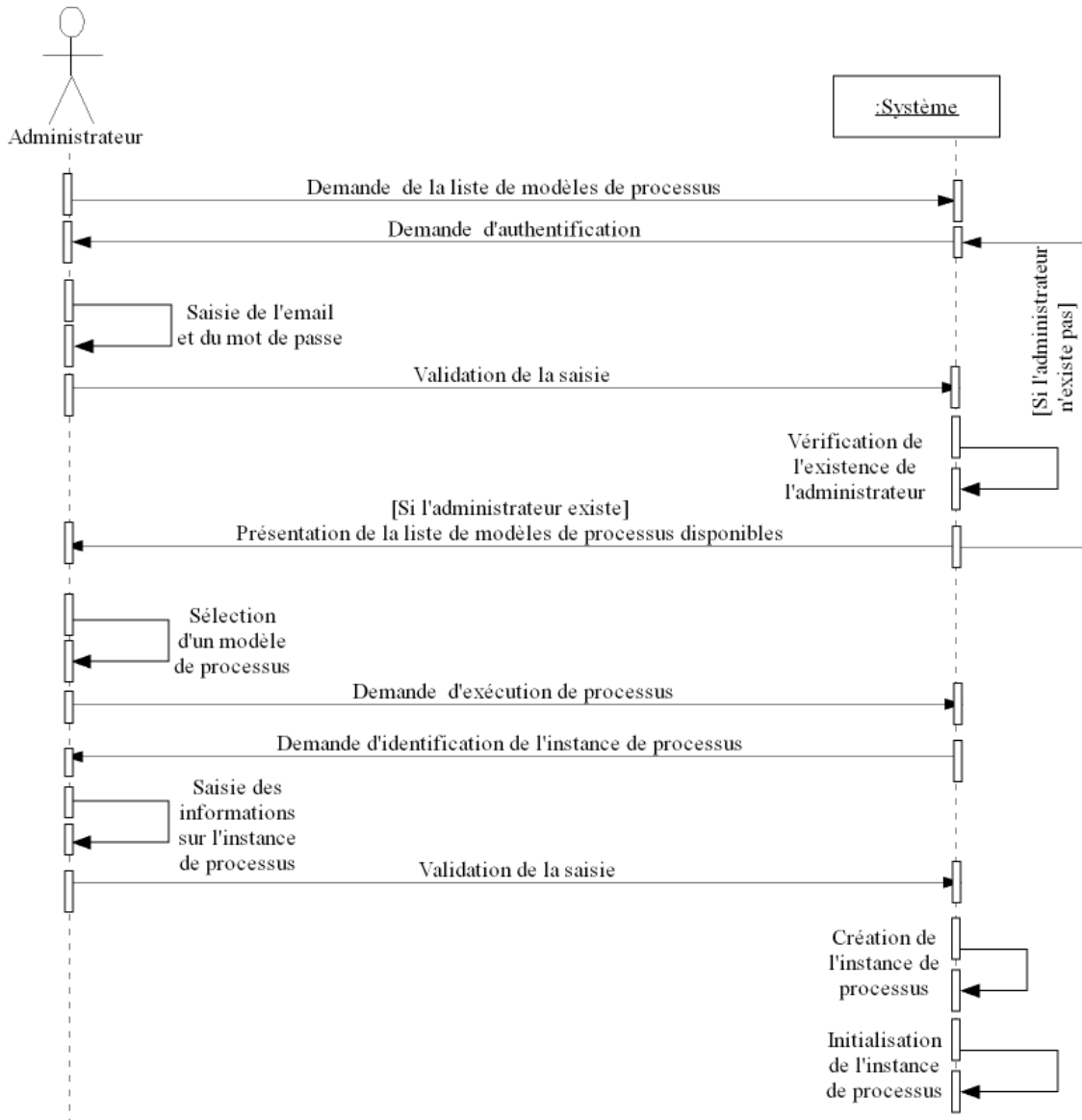


Figure III.28 : Diagramme de séquence de «Instancier un processus»³⁶

3 - Instancier un sous-processus :

L'instanciation d'un sous-processus revient à la charge du tuteur qui aura le droit de décider du sous processus à activer et de choisir les groupes à lancer. Ces derniers seront notifiés au début de chaque nouveau sous-processus.

4 – Exécuter une activité :

³⁶ Pour le restant des scénarios, on se suffira des diagrammes d'activités.

Une activité est exécutée par un acteur à travers le rôle qui lui a été assigné (tuteur ou apprenant). Au cours de l'exécution d'un processus, un même acteur peut être appelé à exécuter plusieurs activités. Qui plus est, une activité est réalisée par un rôle qui peut être détenu par plusieurs acteurs. Dans ce contexte, la politique d'affectation des tâches consiste après création d'un bon de travail, à l'affecter celui-ci au premier acteur libre trouvé parmi les acteurs détenant le rôle qui réalise ce bon de travail. Si aucun acteur n'est libre, on met le bon de travail créé dans la worklist du dernier acteur.

5 - Superviser un processus :

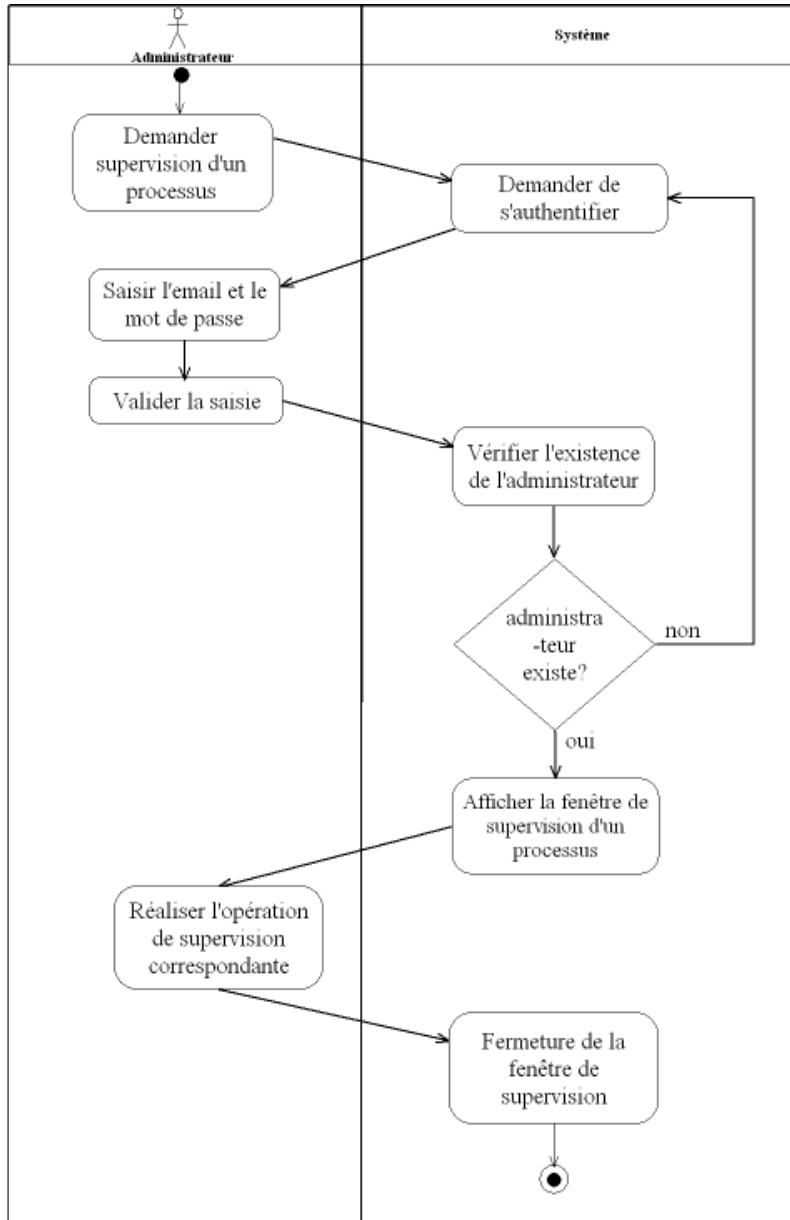


Figure III.29 : Diagramme d'activité du scénario « Superviser un processus »

Commentaire : La supervision d'un processus peut consister par exemple à visualiser sa carte d'instance, consulter la liste de ses activités ou bien à consulter la liste des instances d'activités, ... etc.

Chapitre IV

**Un Système d'Assistance à Base
d'Agents pour le Tutorat Distant**



Chapitre IV

Un Système d'Assistance à Base d'Agents pour le Tutorat Distant

Comme nous venons de le voir dans la partie précédente, l'automatisation des interactions entre tuteurs et apprenants est possible grâce à la modélisation par workflow. Les modèles de tutorat distant dans un cadre de e-learning sont construits à partir de cahier de charge pédagogique élaboré pour convenir aux besoins du domaine abordé. Les processus doivent répondre plus à des besoins de pédagogie que d'efficacité au sens "Business" du terme. Ceci, nous le pensons est assuré par la démarche ASPI, qui permet de représenter toutes les variables nécessaires à l'enseignement. Les points critiques nécessitant l'interaction sont plus efficacement repérés par le staff pédagogique et tout redressement de la situation reste possible grâce à la flexibilité offerte par le méta-modèle.

Cependant, la première question qui se pose est : "*le tutorat veut-il dire uniquement l'interaction ?*". Il est évident que non, bien que l'interaction soit une composante essentielle du tutorat. En effet, à travers son interaction avec les apprenants, le tuteur guide, oriente et conseille l'apprenant. Plus encore et de par sa formation d'enseignant, il est sensé deviner certaines situations psychopédagogiques inconfortables pour ses apprenants et essayer d'agir en sorte de leur remonter le moral, les motiver, etc. Ce genre de tutorat "idéal", ne saurait être atteint pour le moment dans un contexte distant et ceci n'est guère la prétention de ce travail.

Vient alors une seconde question : "*ne peut-on pas améliorer le système proposé ?*". En effet, les processus workflows sont généralement des processus asynchrones, ceci peut poser un problème d'efficacité du « tutorat » car les acteurs ne travaillent pas obligatoirement au même moment, ce qui peut pénaliser les apprenants et même parfois les tuteurs (obligation d'attendre que l'autre réagisse). Afin de régler ce problème, nous avons pensé à introduire un autre type d'acteurs dans la composante organisationnelle du méta-modèle. Ces acteurs sont des "Agents" software. Nous proposons d'organiser ces acteurs en un système multi-agents (SMA) destiné à assister les apprenants et les tuteurs de manière automatique autant que possible. Ceci se fera en délégrant certaines tâches (automatisables) de « tutorat » à des agents.

Enfin, il nous semble important de citer ici que le système de tutorat à base de workflow peut fonctionner avec ou sans le système d'assistance à base de SMA. Ceci laissera le choix aux responsables du dispositif de tutorat de décider de son activation ou pas sans grande perturbation des programmes de formation qui peuvent être en cours.

IV.1 Présentation générale du système d'assistance

L'objectif du système d'assistance n'est pas d'éliminer la communication humaine entre tuteurs et apprenants mais plutôt de fournir une aide et un suivi répondant plus rapidement aux attentes des apprenants et des tuteurs. Ainsi, ces derniers sentiront la présence d'un tuteur/assistant artificiel et bénéficieront du maximum de guidage/information jugé possible par des agents logiciels. La figure suivante illustre un peu le déroulement général d'un tel système assisté par un SMA :

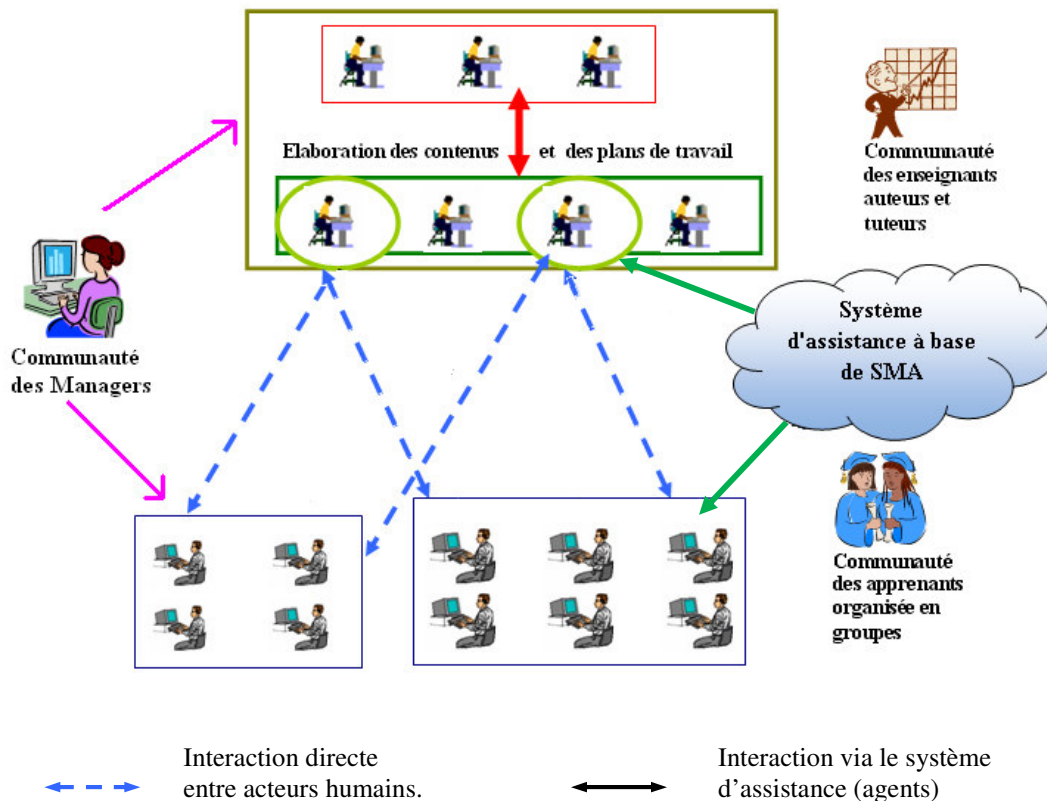


Figure IV.1 : Assistance virtuelle à l'aide d'un SMA.

Donc, les acteurs artificiels sont des agents dotés de la capacité de raisonner et de prendre des décisions (Garro & Al., 2003) (Ouahrani & Al., 2003). Les agents peuvent communiquer entre eux et avec les acteurs humains par échange de messages. Ils peuvent aussi exécuter des tâches complexes seuls ou en coopération avec d'autres agents (Ouahrani & Al., 2003). Chaque agent joue un rôle spécifique dans le système. L'organisation du SMA est inspirée des celles des Systèmes Tuteurs Intelligents (STI(s)) (Aimeur & al, 1999) (Mahdaoui, 2002).

Parmi les avantages que ce SMA peut apporter :

- ▲ Rendre le tutorat plus facile grâce à la délégation de certaines tâches jugées automatisables aux agents. Cela permettra au tuteur de se focaliser plus sur les vrais problèmes de tutorat et qui ne peuvent être réglés que par lui-même.

- ▲ Offrir une certaine interactivité "instantanée", nécessaire pour une bonne évolution de l'apprenant et une meilleure progression de l'apprentissage grâce à l'application de stratégies pédagogiques.
- ▲ De plus, le SMA collectera beaucoup de connaissances de par son exercice. Ces connaissances pourront contribuer à l'amélioration de la qualité des stratégies de tutorat et pédagogie appliquées. Une stratégie de tutorat concerne la manière de suivre d'un tuteur alors qu'une stratégie pédagogique concerne la manière d'étudier d'un apprenant.

Ainsi, même en absence du tuteur, le système multi-agents (SMA) construit et synthétise des rapports sur l'état d'évolution des instances de workflow (donc l'évolution des apprenants) de façon individuelle ou groupée. La solution que nous présentons décrit donc un système d'information coopératif à base d'agents destiné à supporter le travail de nos acteurs.

Notons enfin que pour l'instant le SMA proposé est relatif au tutorat individuel donc ne prenant en charge que des processus de type PAI. Le tutorat collectif - dans le cadre de PAC (Plan d'Apprentissage Collectif) - fera partie de nos futurs travaux.

IV.2 Extension de l'aspect organisationnel du méta-modèle aux acteurs artificiels

Nous présentons dans ce qui suit (figure IV.2), l'ensemble des acteurs artificiels organisés en un SMA et avec lequel nous avons étendu notre méta-modèle de workflow pour le e-learning. Les liens entre les agents eux-mêmes seront explicités durant la phase de conception du système sur le diagramme de classe AUML.

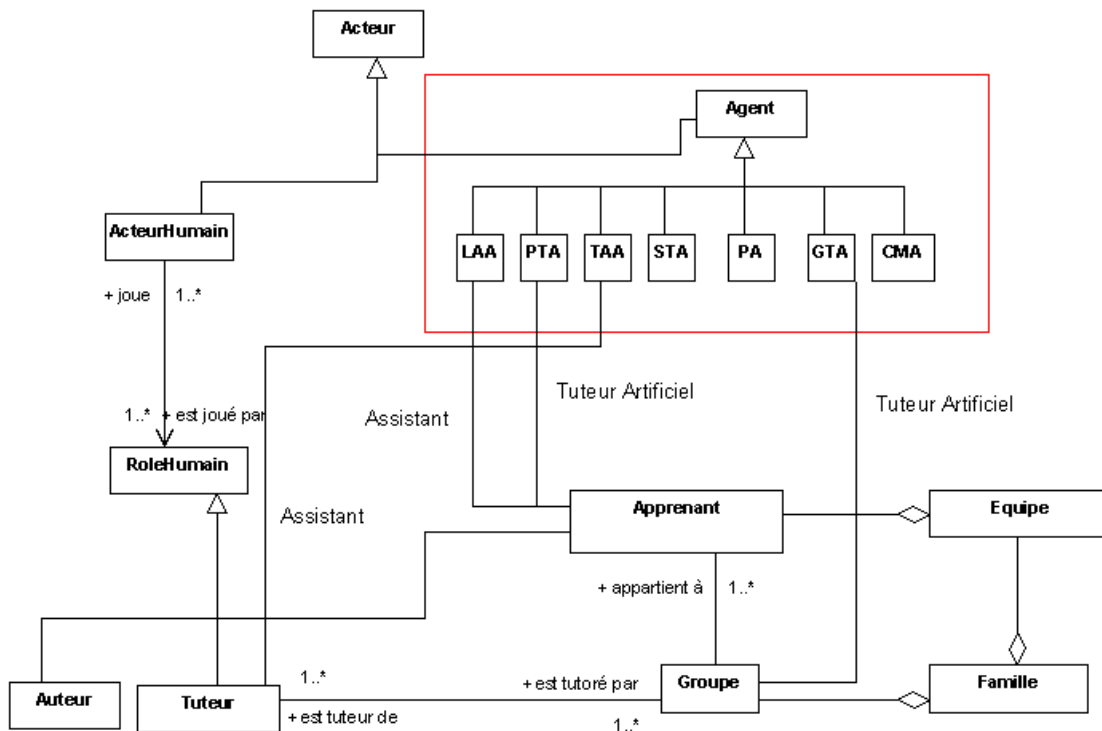


Figure IV.2 : Les acteurs artificiels du Méta-modèle de workflow pour le e-learning

IV.2.1 Description des acteurs du SMA

Notre système d'acteurs automatiques est formé de sept (06) agents cognitifs et d'un agent réactif (l'agent CMA) formant un système multi-agents (SMA) que nous présentons dans le tableau suivant :








Agent	Description	Rôle
LAA 	Learner's Assistant Agent	Assister l'apprenant et le représenter vis-à-vis des autres agents du SMA.
PTA 	Personal Tutor Agent	Tuteur personnel et artificiel de l'apprenant.
PA 	Pedagogical Agent	Chargé d'appliquer des stratégies pédagogiques aux apprenants.
TAA 	Tutor's Assistant Agent	Assister le tuteur et le représenter vis-à-vis des autres agents du SMA.
GTA 	Group Tutor Agent	Tuteur artificiel d'un groupe d'apprenants.
STA 	Super Tutor Agent	Capitaliser la connaissance de tous les agents GTA.
CMA 	Connexion Manager Agent	Offre toutes les informations nécessaires concernant l'état de connexion/déconnexion des apprenants et des tuteurs.

Table IV.1 : Description des agents du système d'assistance

Décrivons dans ce qui suit le fonctionnement de chaque agent.

IV.2.1.1 LAA : l'agent assistant de l'apprenant

Le rôle de LAA à assister l'apprenant à chaque session d'apprentissage. C'est le représentant de l'apprenant envers les autres le concernant (l'apprenant). Un agent LAA est associé à un apprenant à la fois. Il présente les fonctionnalités suivantes :

- ✓ LAA est créé lorsque l'apprenant se connecte pour la toute première fois au système d'enseignement qui peut être par exemple, une plate forme de e-learning.
- ✓ LAA aide l'apprenant dans le choix du module à étudier. Ensuite, il réveille et informe l'agent PTA que l'apprenant est en ligne.
- ✓ Lorsque l'apprenant veut obtenir de l'aide d'un tiers, LAA met à sa disposition un buffer spécial où il pourra écrire sa requête qui est sous forme textuelle pour les humains; pour l'agent PTA, l'espace est structuré de telle façon qu'il puisse arriver à interpréter la requête de l'apprenant. L'usage de formulaires adéquats serait recommandable dans ce cas. Donc, l'apprenant pourra préciser le destinataire de sa requête pouvant être son tuteur artificiel personnel (agent PTA), son tuteur humain ou un autre apprenant de son groupe.

- ✓ Si la requête est adressée à un humain (tuteur ou apprenant) alors LAA demandera à l'agent CMA de l'informer si le tuteur est connecté. Si la réponse de CMA est affirmative alors LAA demandera à l'humain (via l'agent lui correspondant : LAA pour l'autre apprenant ou TAA pour le tuteur) s'il veut bien converser ou voir la requête de l'apprenant demandeur de l'aide. Après réception d'une réponse, LAA informera l'apprenant quel que soit le cas, i-e, soit que "personne n'est disponible" soit que "quelqu'un est prêt à l'aider". Commencera alors une conversation directe entre humain via un outil de "Chat".
- ✓ Si l'apprenant ne précise pas le destinataire – par exemple, il cherche n'importe quel apprenant connecté de son groupe -, LAA demandera à l'agent CMA de lui fournir la liste des apprenants connectés. Après réception de la liste, LAA tentera de trouver un apprenant favorable au dialogue et transmettra le résultat à l'apprenant. Il est possible qu'il trouve plus d'un apprenant favorable au dialogue, dans ce cas la liste sera remise à l'apprenant.
- ✓ De plus, LAA peut accéder à la liste des emails de l'apprenant dès qu'il est activé et classifier les messages non encore traités. Les messages du tuteur seront toujours placés en tête de liste. Généralement, LAA utilisera la liste des messages pour conseiller l'apprenant sur le choix du module à étudier par exemple.

IV.2.1.2 PTA : l'agent tuteur personnel de l'apprenant

PTA est un tuteur artificiel personnel de l'apprenant et qui le suit dans chaque activité d'apprentissage. A chaque apprenant est associé un agent PTA. PTA connaît la nature et les objectifs pédagogiques des activités et des tâches les composants¹. Pour les tâches passives, PTA fera attention à certains paramètres comme la durée estimée de la tâche, la date prévue pour sa fin, ...etc. Notons cependant que ce type de tâche n'est pas évident à suivre, d'où PTA essayera d'attirer l'attention de l'apprenant que ladite tâche doit être accomplie.

En général, une tâche ou une activité active se termine toujours par un feedback de l'apprenant. Les activités comme les exercices par exemple sont particulièrement importantes pour un bon apprentissage et représentent des objectifs pédagogiques comme :

- Le test des connaissances acquises.
- Accroître les capacités de l'apprenant à analyser et faire des preuves.
- Evaluer les aptitudes et les lacunes de l'apprenant.

Les fonctionnalités de PTA se résument en :

- ✓ PTA reçoit un plan d'apprentissage (PAI) de la part de l'agent GTA et charge la "worklist" de l'apprenant avec l'activité à exécuter. Chaque tâche de l'activité correspond à un "workitem" et PTA suivra son accomplissement.

¹ Voir chapitre III pour le concept de tâche passive ou active.

- ✓ Si la tâche est passive, l'apprenant peut décider de la sauter (parce qu'il connaît le contenu par exemple), mais PTA notera le fait. Et plus tard il pourra l'utiliser dans ses décisions tutorielles.
- ✓ Si la tâche est active, PTA attendra le feedback de l'apprenant. Si nécessaire, PTA activera son agent subordonné PA pour appliquer une stratégie pédagogique appropriée à l'apprenant. Une fois terminé, PTA recevra un rapport de PA.
- ✓ Les connaissances de PTA lui permettent de juger si une tâche s'est correctement terminée ou pas, i-e, si les résultats de l'apprenant sont satisfaisant ou pas. PTA prendra alors l'une des décisions suivantes :
 - Si les résultats sont jugés bons alors PTA passera à l'étape suivante du travail qui peut être la prochaine tâche ou la prochaine activité à effectuer. Si tout le PAI est correctement terminé alors PTA demandera à l'agent GTA la prochaine étape du travail. Selon la réponse de GTA, PTA devra poursuivre son tutorat ou s'arrêter. Au fait, la réponse de GTA dépendra de ce qui est prévu par le tuteur humain. En cas d'arrêt, PTA enverra un rapport détaillé sur l'apprenant à l'agent GTA.
 - Si les résultats ne sont pas bons, PTA tentera de résoudre le problème selon les étapes suivantes :
 - Il proposera d'abord à l'apprenant de refaire le travail à l'étape qui semble avoir causé le problème.
 - Si le problème persiste, i-e, les résultats ne sont toujours pas concluants, PTA utilisera sa propre expérience et proposera d'autres travaux (des tâches complémentaires) à l'apprenant pour renforcer ses connaissances.
 - Si PTA n'a pas les compétences requises ou que les résultats demeurent mauvais, GTA sera sollicité pour de l'aide.
 - La réponse de GTA sera soit une nouvelle stratégie de tutorat soit l'information que le problème a été transmis au tuteur humain. Dans ce cas, PTA arrêtera son travail jusqu'à la réception de nouvelles directives.

Enfin, PTA peut aussi coopérer pour résoudre des demandes provenant de l'agent GTA pour le compte d'autres agents PTA(s). PTA peut bien sûr ignorer la requête s'il n'a pas de réponse à envoyer ou bien qu'il est trop occupé avec son propre apprenant.

IV.2.1.3 PA : l'agent pédagogique

Le rôle de PA est de soumettre l'apprenant à différentes stratégies d'apprentissage afin de s'assurer qu'il a (l'apprenant) bien compris et pour combler les lacunes de ce dernier. PA est un agent subordonné à PTA et ce dernier le sollicitera si nécessaire.

Selon le but pédagogique visé, PA pourra être un "compagnon", un "perturbateur" ("Troublemaker") ou bien un "élève de l'apprenant" (Aimeur & al, 1999), (Mahdaoui, 2002). Ses fonctionnalités sont :

- ✓ PA peut appliquer l'une de ses stratégies :
 - Apprendre avec un compagnon : PA devient un co-apprenant (un ami) avec un niveau de connaissances légèrement supérieur à celui de l'apprenant. PA guidera l'apprenant durant son apprentissage par des conseils et des suggestions (selon ce que PA connaît). L'apprenant sera libre de choisir et de décider.
 - Apprendre par perturbation ("Learning by disturbing") : PA devient un perturbateur. Avec cette stratégie, le niveau des exercices devient plus élaboré et vise à provoquer des situations de conflit (dissonance cognitive) entre les connaissances de PA et les convictions de l'apprenant (Aimeur & al, 1999). Ainsi, PA peut détecter les aptitudes et les lacunes de l'apprenant en matière de preuve et d'analyse.
 - Apprendre en enseignant : C'est l'ultime étape où PA devient l'élève de l'apprenant, i-e, l'élève a un très haut niveau de connaissances et de capacités. Donc l'élève devra enseigner à l'agent PA.
- ✓ Une fois son travail terminé, PA transmettra à PTA un rapport contenant les évaluations, les notes et d'éventuelles observations.

IV.2.1.4 TAA : l'agent assistant du tuteur

Similairement à LAA, l'agent TAA représente le tuteur vis-à-vis des autres agents du système et doit assister ce dernier durant les sessions de tutorat. Un agent TAA est associé à un et un seul tuteur. Il a principalement les fonctionnalités suivantes :

- ✓ TAA est créé dès que le tuteur se connecte pour la première fois à la plate forme. Il est activé/désactivé en fonction des connexions/déconnexions du tuteur.
- ✓ Le tuteur peut avoir à suivre différents groupes dans différents modules. Dans ce cas TAA aidera le tuteur à faire ses choix. Ensuite, TAA informera GTA que le tuteur est en ligne.
- ✓ Lorsque le tuteur veut communiquer avec des agents ou ses apprenants, il décrit sa requête dans un espace spécial fourni par TAA avec la possibilité de choisir le destinataire.
- ✓ Lorsque le tuteur veut envoyer sa réponse concernant un problème soulevé par GTA, il utilisera TAA aussi.
- ✓ TAA traite les requêtes comme décrit pour l'agent LAA.
- ✓ TAA accède aux emails du tuteur dès qu'il est activé et classe les messages arrivés et non encore traités. Les messages des apprenants sont toujours placés

en tête de liste et en général il utilisera la liste classée pour conseiller et orienter sur le groupe et la matière à tutorer en priorité.

IV.2.1.5 GTA : l'agent assistant d'un groupe d'apprenants

L'agent GTA est associé à un groupe d'agents PTA qui représentent au fait le groupe d'apprenants humains et un tuteur pour une matière donnée. GTA est un tuteur artificiel pour tous les membres du groupe dans le sens où pour tout problème que les agents PTA ne peuvent pas résoudre, il (GTA) essaiera de le faire. GTA coopère avec les agents PTA et l'agent STA. De plus, il informe régulièrement le tuteur sur l'évolution de son groupe. Ses fonctionnalités se résument en :

- ✓ Quand GTA reçoit une requête de l'agent PTA, il tente de répondre selon sa propre expérience qui se base sur les connaissances acquises antérieurement de chez les agents PTA, de l'agent STA ou du tuteur lui-même.
- ✓ Si GTA ne trouve pas de solution, il demandera à l'agent CMA de lui fournir la liste des PTA(s) actifs, les PTA(s) inactifs ayant fourni leurs informations à GTA juste avant leur arrêt.
- ✓ Selon la liste obtenue, GTA diffusera une requête pour l'assistance en tutoring. Si aucun résultat concluant n'est obtenu alors GTA fera appel à l'aide de l'agent STA.
- ✓ Une fois les résultats obtenus, GTA les analyse et les organise en un rapport qu'il enverra au PTA demandeur. Si aucun résultat n'a été trouvé, GTA informera PTA que le problème sera soumis au tuteur.
- ✓ Eventuellement, GTA peut recevoir une demande d'assistance émanant de l'agent STA pour le compte d'autres agents GTA, auquel cas il pourra soit participer ou ignorer carrément la requête.

IV.2.1.6 STA : l'agent super tuteur

STA est un agent qui capitalise toute l'expérience relative aux agents GTA qui tutoient le même module ou matière (pour le même profil d'apprenants) ainsi que toute l'expérience concernant le même groupe d'apprenants pour différents modules. Pour l'instant nous considérons qu'il existe un seul STA pour tous les groupes d'une même famille (au sens du méta-modèle) d'apprenants. Voici ses fonctionnalités :

- ✓ Lorsque l'agent STA reçoit une requête de l'un des agents GTA, il essaye de la traiter selon ses propres connaissances qui résultent entre autre de l'expérience recueillie auprès des agents GTA(s).
- ✓ S'il n'arrive pas à trouver de résultats, STA diffusera sa requête envers les GTA(s) actifs dont il aura obtenu la liste par l'agent CMA. Le reste de la procédure est similaire au scénario GTA/PTA.
- ✓ Quand STA reçoit une requête du tuteur humain, selon ce qui lui est demandé, STA traitera la demande selon l'un des desseins qui viennent d'être décrits.

IV.2.1.7 CMA : l'agent gestionnaire de connexion

L'agent CMA est un agent d'arrière plan. Il a la responsabilité de sauvegarder toutes les informations concernant la connexion/déconnexion des apprenants et des tuteurs ainsi que les agents actifs/inactifs. Actuellement, on considère un seul CMA pour tout le système. CMA est un agent persistant lancé au démarrage du système. Sa principale fonctionnalité est :

- ✓ Traiter les demandes provenant de tous les agents du SMA et des tuteurs humains. Bien sûr, on supposera que le nombre d'agents et de tuteurs est raisonnablement choisi par rapport aux capacités de traitement de l'agent CMA afin d'éviter tout blocage ou chute de performances du système.

IV.3 Propriétés des agents du SMA (selon la grille de J. Ferber)

Afin de décrire les propriétés de nos agents nous allons utiliser une grille adaptée de la grille de J. Ferber qu'il propose dans (Ferber, 1995). En effet, J. Ferber propose d'associer six fonctions aux agents : la fonction représentationnelle liée à la représentation des connaissances ; la fonction organisationnelle liée à la planification d'actions, à la nature rationnelle et adaptative de l'agent ; la fonction conative liée aux besoins, désirs, à la nature autonome; la fonction interactionnelle liée aux communications; la fonction productive liée aux différentes actions que peut entreprendre l'agent et la fonction conservatrice liée à la protection, à la conservation de l'agent ou du système (Ferber, 1995). La table IV.2 résume l'ensemble des propriétés :

	Personnel	Environnemental	Social	Relationnelle	Physique
Représentationnel					
Organisationnel					
Conative					
Interactionnel					
Productif					
Conservatif					

Table IV.2 : Grille d'analyse fonctionnelle des propriétés d'agents

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

Donc, la grille d'analyse fonctionnelle permet de classer les différents types d'agents et de systèmes multi-agents, d'où il nous a semblé intéressant d'utiliser cette grille sur les agents que nous proposons pour constituer le SMA d'assistance au tutorat.

Selon le type d'agents, les fonctions sont plus ou moins développées, voire inexistantes. Par exemple, les agents assistants que l'on voit apparaître dans les suites

bureautiques ont un comportement essentiellement de réaction à des événements. En revanche, les systèmes experts distribués ont des capacités de réflexion plus élevés (Adam, 2000).

Dans ce travail de thèse nous nous intéresserons particulièrement à décrire les agents LAA, PTA, PA, GTA et CMA car ils concernent directement les apprenants et les tuteurs² (STA sera plus détaillé dans nos travaux futurs). Nous décrivons les caractéristiques de ces agents en utilisant une adaptation de la grille de J. Ferber restreinte aux deux dimensions "Individuelle" et "Sociale" (Ferber, 1995). Les propriétés décrites sont les connaissances, l'organisation et les interactions.

A. Connaissances :

		Dimensions Individuelle et Sociale
Agents	LAA	Il a une représentation de lui même (identité, états et actions) et de celle de l'apprenant auquel il est associé. Il connaît l'agent PTA associé à l'apprenant ainsi que l'agent CMA. Il connaît aussi les adresses des tuteurs humains associés aux matières suivies par l'apprenant.
	PTA	Il se connaît lui même et a une représentation générale du sous-processus ou plan d'apprentissage que l'apprenant doit suivre. PTA connaît LAA, GTA, CMA et le tuteur humain associé à l'apprenant et à la matière qu'il a en charge.
	PA	Se connaît lui même et a une représentation des stratégies pédagogiques à appliquer. Il connaît aussi l'agent PTA auquel il est associé [14].
	GTA	Dispose d'une description de lui même et connaît tous les agents PTA appartenant à son groupe. GTA connaît aussi CMA et le tuteur humain auquel il est associé.
	CMA	A une représentation de lui même et des actions qu'il peut entreprendre. Il connaît la liste de tous les acteurs humains connectés ou déconnectés.

Table IV.3 : Connaissances des agents

B. Organisation :

		Dimensions Individuelle et Sociale
Agents	LAA	Il est muni d'une interface lui permettant de communiquer avec l'apprenant humain. Il représente l'apprenant vis à vis des autres agents du système (CMA, PTA et GTA). Dès la connexion de l'apprenant, LAA active l'agent PTA qui lui est associé. LAA peut élaborer des plans d'actions conformément aux choix et souhaits de l'apprenant.

² Les agents STA et PA dont l'envergure est plus grande seront développés dans de futurs travaux.

	PTA	Elabore les plans d'actions conformément aux plans d'apprentissage dont il a la charge. PTA assure un guidage et une assistance aussi bien au niveau sous-processus qu'au niveau activité. Il activera PA uniquement si nécessaire. Si l'apprenant est le premier de son groupe à se connecter, PTA aura la charge d'activer l'agent GTA du groupe.
	PA	Il est sous la responsabilité de PTA et applique ses propres plans d'actions pédagogiques.
	GTA	Elabore les plans d'actions nécessaires pour le suivi du processus d'apprentissage global d'un groupe donné pour une matière donnée.
	CMA	Il tient à jour la liste de tous les acteurs humains connectés ou déconnectés.

Table IV.4 : Organization des agents

C. Interactions :

		Dimensions Individuelle et Sociale
Agents	LAA	Interagit avec l'apprenant auquel il est associé par le biais d'interfaces et de langage approprié conformément aux plans d'actions lui correspondant.
	PTA	Interagit avec LAA, GTA, CMA.
	PA	Interagit avec PTA.
	GTA	Interagit avec PTA et CMA.
	CMA	Peut communiquer avec tout agent qui le sollicite.

Table IV.5 : Interactions des agents

IV.4 Lien entre le système workflow et le SMA

Le lien entre le système workflow pour le tutorat et le SMA sera établi via les activités et les tâches qui seront reçues dans les worklists et les workitems respectifs des apprenants et des tuteurs. Dès qu'une instance de processus de tutorat est créée, le tuteur fournira à l'agent GTA les plans d'apprentissage que les apprenants du groupe doivent effectuer et ce dernier commencera à fournir aux agents PTA de son groupe les instances de PAI à exécuter.

Ensuite, chaque fois qu'un apprenant recevra une activité à accomplir, le rôle des agents qui lui sont associés sera de le suivre jusqu'à ce qu'il accomplisse avec succès son apprentissage. Particulièrement, l'agent PTA suivra l'accomplissement d'un PAI jusqu'à sa fin pour passer à un autre. Pour le tuteur, il s'agit plus de lui fournir un moyen de mieux définir ses priorités en fonction de ce que lui fournira l'agent GTA comme informations souvent sous forme de rapports synthétiques et d'éventuelles statistiques si le tuteur le demande.

Notons toutefois qu'une instance d'agent existera tant que l'apprenant et le programme (processus global) d'enseignement de la matière existera.

Enfin, Le schéma de la figure IV.3 résume l'ensemble des interactions possibles entre les différents agents du système d'assistance. Certains agents peuvent communiquer et/ou coopérer entre eux. La communication avec les acteurs humains est aussi possible.

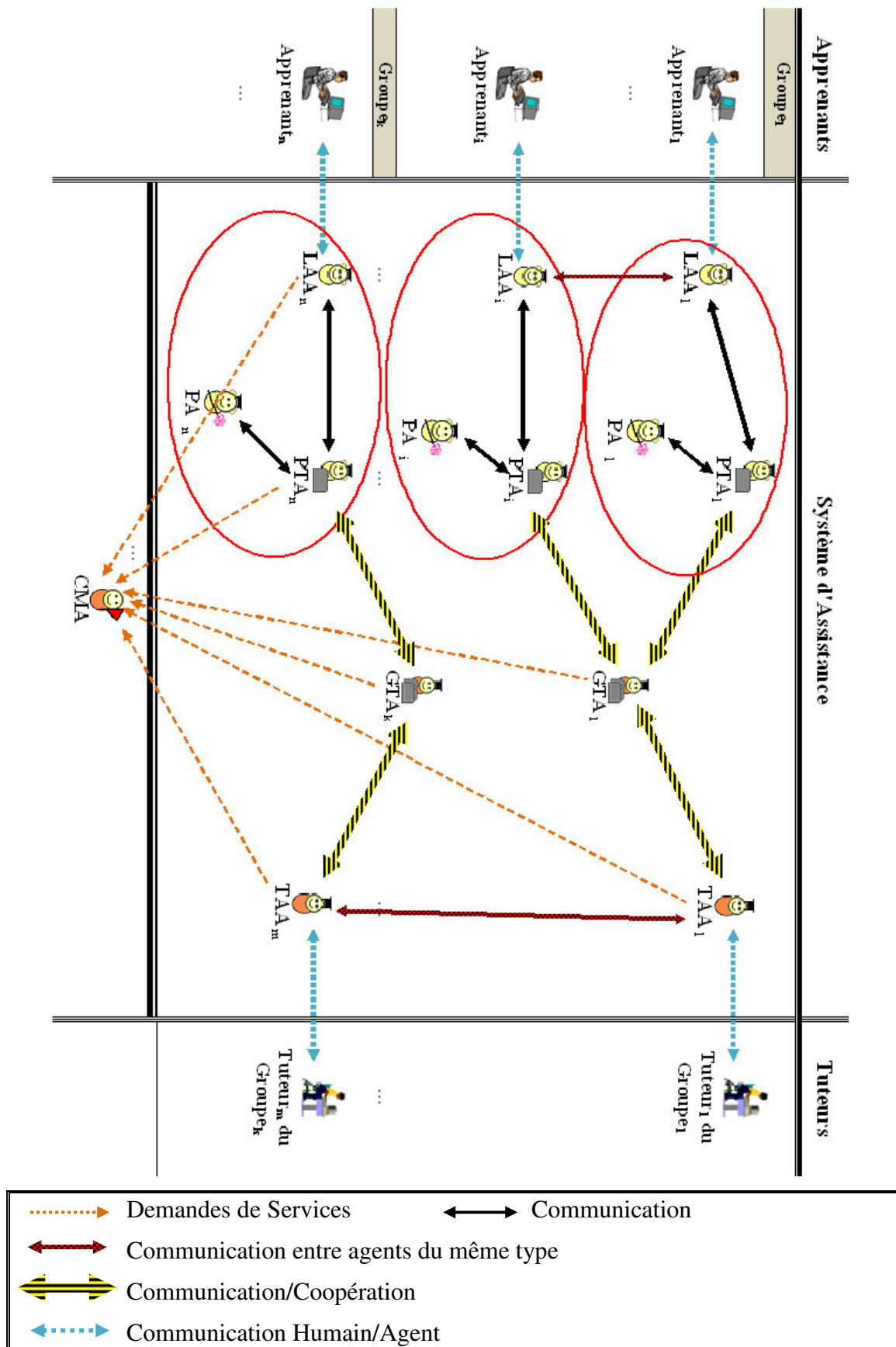


Figure IV.3 : Interactions possibles dans le système d'assistance

IV.5 Conception du SMA

IV.5.1 Démarche de conception du SMA

Nous avons choisi de faire la conception du SMA à l'aide du langage AUML car il offre un ensemble de diagrammes³ enrichis de concepts relatifs aux agents et ne pose de contraintes par rapport aux outils à utiliser plus tard pour la phase de développement. D'un autre côté, AUML étant une extension d'UML permet d'en utiliser tous les diagrammes pour exprimer divers aspects du système projeté.

Pour notre cas, nous avons sélectionné l'ensemble des diagrammes suivants :

- Les diagrammes de cas d'utilisation : vont nous servir à analyser toutes les fonctionnalités attendues du système.
- Les diagrammes de séquence : vont documenter les différents scénarios associés aux fonctionnalités.
- Les diagrammes d'états/transition : vont permettre de décrire tous les états que traversera un agent durant sa période d'activité.
- Les diagrammes de classe : permettent de décrire les liens et associations entre les agents et la structure de chaque agent.

IV.5.2 Fonctionnalités du SMA

La majorité des fonctionnalités résumées par la figure IV.4 ont été expliquées en section IV.2.1. Notons toutefois que chaque fonctionnalité peut faire l'objet de un ou plusieurs scénarios possible selon la situation et le ou les acteurs impliqués comme nous le verrons dans les prochaines étapes.

³ Voir le chapitre II.

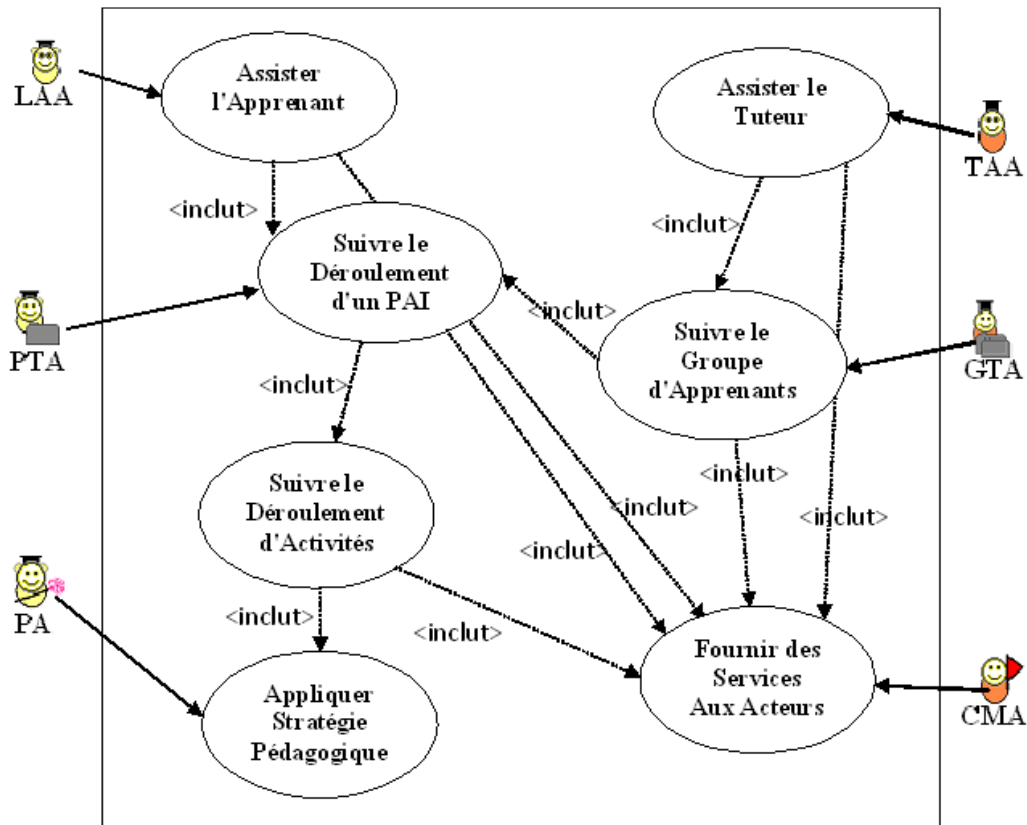


Figure IV.4 : Fonctionnalités associées aux agents

IV.5.3 Description des scénarios par les diagrammes de séquence

En plus des scénarios de comportement interne de chaque agent selon la matière enseignée et le rôle des agents eux-mêmes, il existe un ensemble de scénarios standards relatifs au fonctionnement du système de tutorat à base de workflow et dans lequel les agents ont leur part de travail à accomplir si le système d'assistance est activé.

L'espace de cette thèse ne permettant de décrire tous les scénarios à titre exhaustif, nous allons nous restreindre aux plus importants que nous citons dans ce qui suit et que nous détaillerons en prochaine section :

- * **Scénario 1** : Connexion d'un apprenant
- * **Scénario 2** : Suivre le déroulement d'un PAI
- * **Scénario 3** : Suivre le déroulement d'une activité
- * **Scénario 4** : Sollicitation d'assistance par un apprenant

IV.5.3.1 Connexion d'un apprenant

Pour un processus de e-learning donné, une instance est déclenchée lorsqu'un apprenant rentre dans une session d'apprentissage et se termine dès qu'il sort de la session (on suppose qu'une session se termine normalement). Plusieurs instances de processus peuvent exister en même temps (connexion simultanée de plusieurs apprenants), chacun pouvant se trouver à une étape donnée de l'évolution du processus.

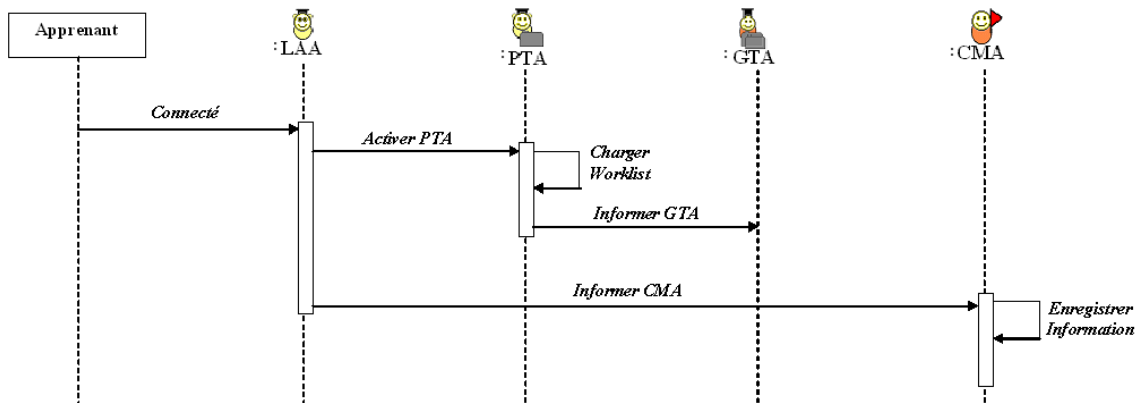


Figure IV.5 : Connexion d'un apprenant

Lors de sa connexion, l'apprenant s'identifie au système par un nom d'utilisateur et un mot de passe. Le système active alors l'agent LAA, qui à son tour active PTA et informe CMA de la connexion d'un nouvel apprenant. LAA présente à l'apprenant son dernier état d'évolution pour chaque matière. PTA charge la « worklist » de l'apprenant conformément à l'état de l'instance du workflow, puis informe GTA qu'il est actif.

IV.5.3.2 Suivre le déroulement d'un PAI

Dans ce scénario, PTA vérifie d'abord l'état de l'instance en cours d'exécution. Si le plan n'est pas à sa fin, il continuera à suivre l'accomplissement des activités, sinon il informe GTA que le plan d'apprentissage est terminé. GTA vérifie le résultat global de l'instance exécutée, si ce dernier est satisfaisant il passe au prochain sous-processus prévu dans le processus global, sinon il prendra une décision selon un scénario tutoriel prévu par le tuteur humain. LAA affiche à l'apprenant la prochaine étape du processus d'apprentissage.

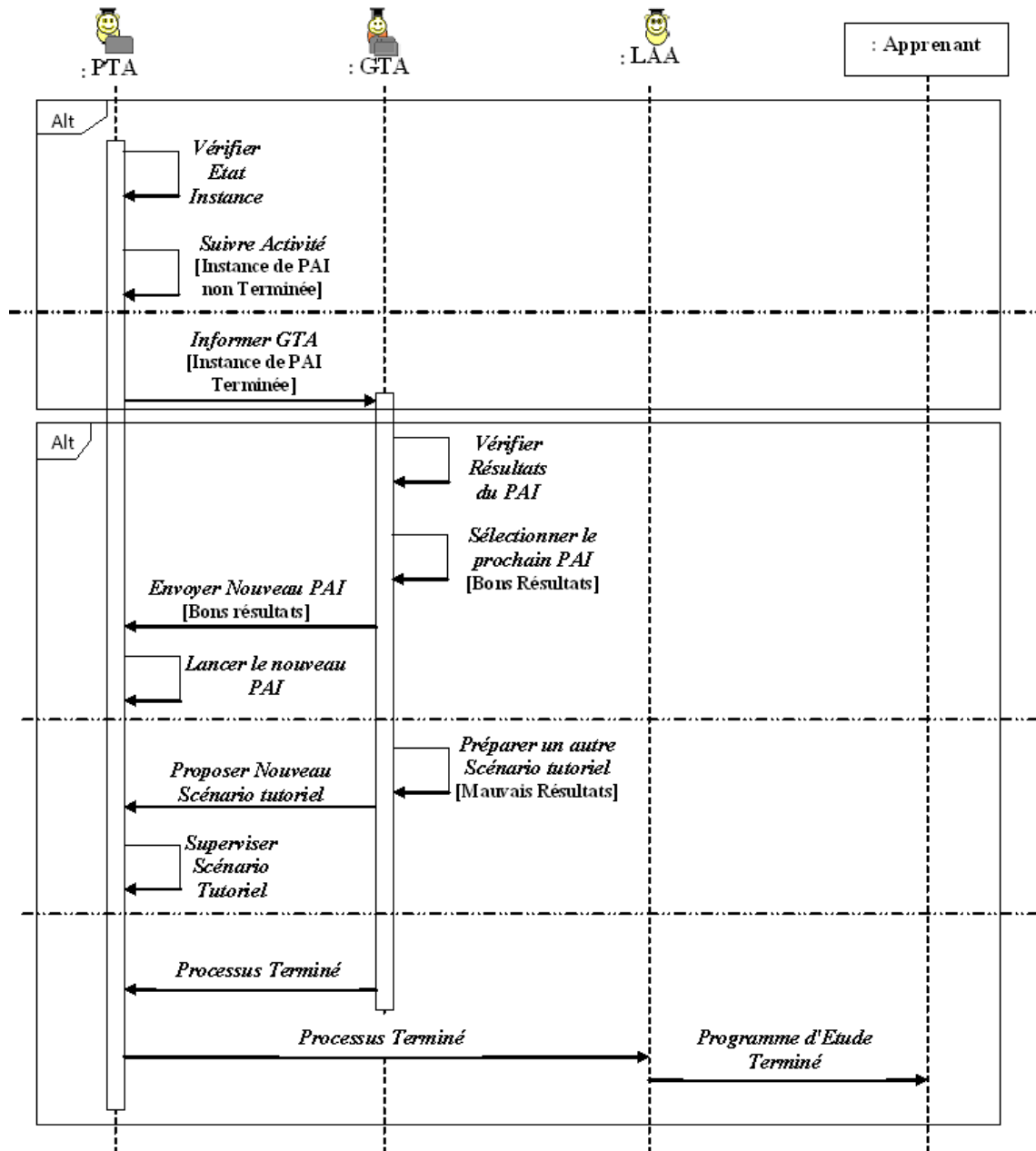


Figure IV.6 : Assistance à l'exécution d'un plan d'apprentissage de l'apprenant

PTA dispose de la liste des tâches à effectuer, celles-ci sont séquentielles et ordonnées. PTA envoie à LAA l'activité à accomplir, LAA l'affiche sous forme d'une liste de tâches à l'apprenant qui sélectionne la tâche à exécuter. Une fois toutes les tâches accomplies (activité terminée), PTA procède à l'évaluation du travail. Selon le cas, PTA peut solliciter PA pour appliquer une stratégie pédagogique à l'apprenant, soit pour parfaire les connaissances de l'apprenant, soit pour parer à ses lacunes. PTA peut avoir à solliciter GTA si ses compétences sont insuffisantes pour juger le travail.

IV.5.3.4 Sollicitation d'assistance par un apprenant

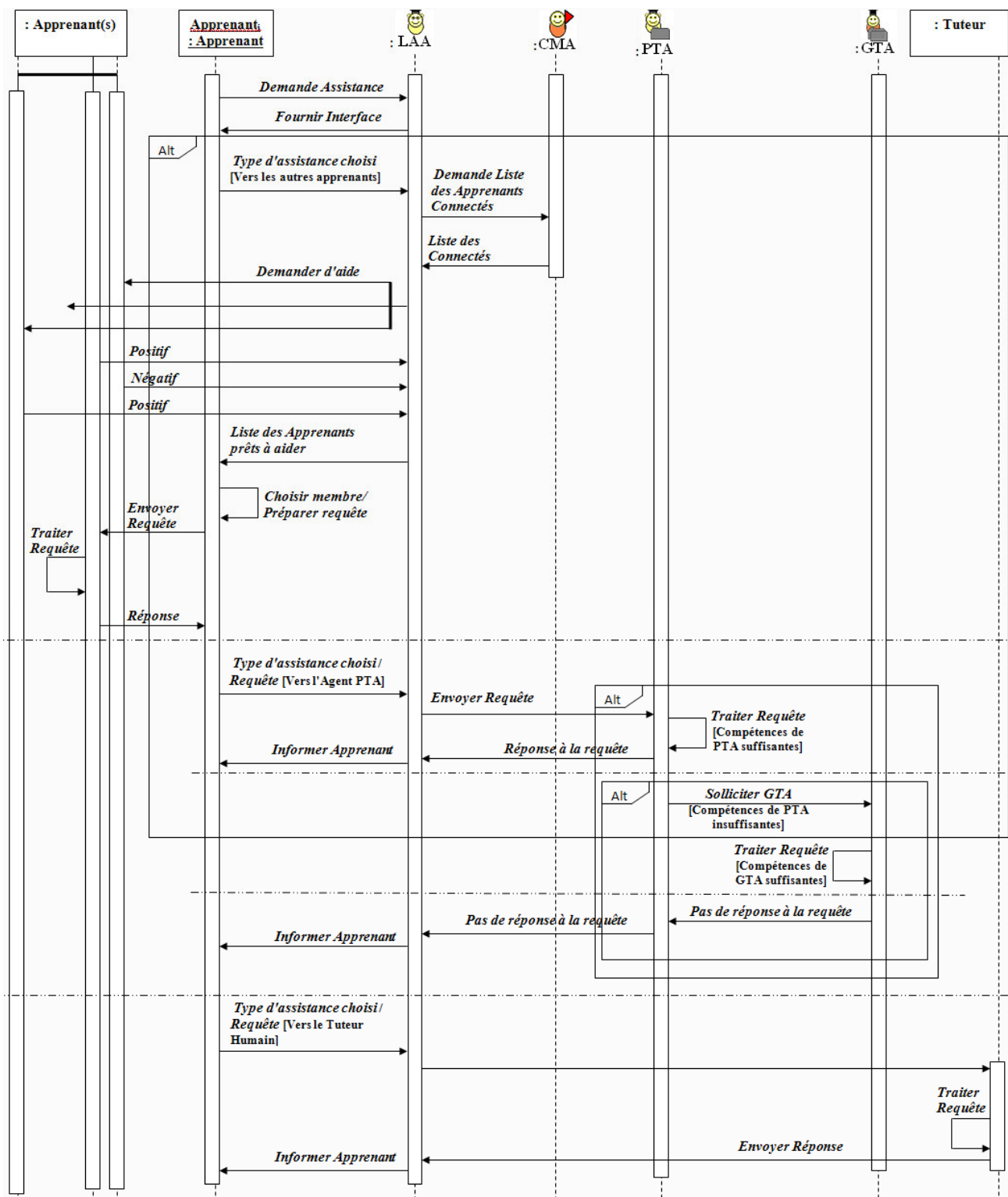


Figure IV.8 : Sollicitation d'assistance par l'apprenant

L'apprenant peut demander de l'aide par le biais de LAA. Ce dernier lui fournit une interface permettant de choisir le type d'assistance : le type1 est l'assistance demandée à un autre apprenant du groupe, le type2 concerne l'assistance fournie par PTA et le type3 est l'assistance fournie par le tuteur humain.

Lors de la résolution d'une requête de l'apprenant, les résultats sont véhiculés d'un agent à un autre jusqu'à arriver à l'apprenant, ceci a pour effet d'enrichir les connaissances et les expériences de chaque agent.

IV.5.4 Diagramme d'état/transition

Durant leur exercice les agents traverseront un ensemble d'états qui reflèteront les situations diverses dans lesquelles un agent peut se trouver. La figure suivante décrit les états concernant tous les agents sauf l'agent CMA qui n'est pas concerné par l'état "Attente" car il fournira ses services automatiquement sur sollicitation externe sans avoir à exécuter aucun plan.

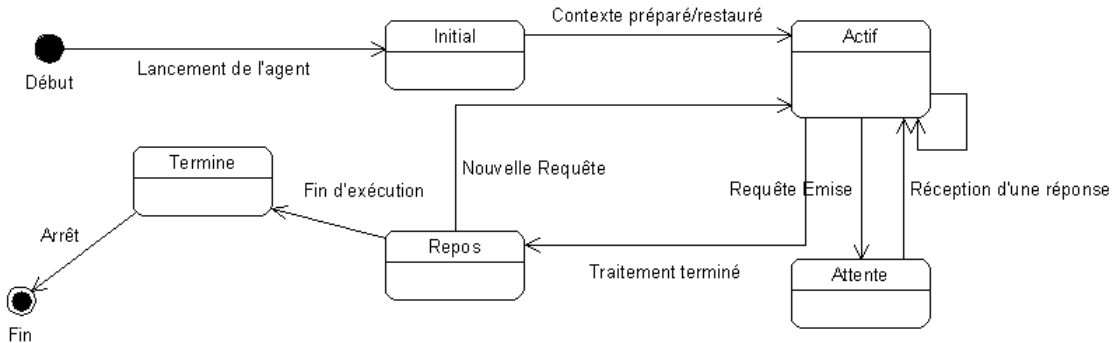


Figure IV.9 : Etats des agents cognitifs

- ✓ **Initial** : dès le lancement de l'agent, il passe à l'état initial où préparera son contexte s'il s'agit de sa première activation. Il restaurera son contexte s'il a déjà exercé auparavant, i-e, son activation consiste à le réveiller.
- ✓ **Actif** : l'agent passera à l'état actif une fois toutes les données sur son environnement disponibles. A cet état l'agent est prêt à recevoir des requêtes et à les traiter en plus de son propre travail. L'agent peut boucler sur l'état actif tant qu'il ne traite pas avec une requête dont le résultat est primordial pour la suite de son travail.
- ✓ **Attente** : dès qu'une requête est émise par un agent, il passera à l'état attente si le résultat de celle-ci est capital pour la suite du travail de l'agent. Il sortira de cet état dès qu'il recevra une réponse.
- ✓ **Repos** : cet état caractérisera l'agent lorsqu'il ne fera rien, i-e, il a terminé tout son travail et attend les prochaines instructions devant lui parvenir de l'agent GTA (par exemple, attendre le prochain PAI).
- ✓ **Termine** : le passage à cet état permettra à l'agent de faire la sauvegarde de son contexte avant de se désactiver. Un agent se désactivera à la déconnexion du ou des acteurs humains le concernant.

IV.5.5 Diagramme de classes

Les diagrammes de classe AUML permettent de mettre en évidence les liens de coopération entre les agents que ce soit en matière de collaboration ou de communication. Les extensions introduites sur l'élément "Classe UML" permet aussi de décrire la structure interne de nos agents comme nous l'avions relaté au chapitre II de l'état de l'art.

La figure IV.10 décrit les liens inter-agents :

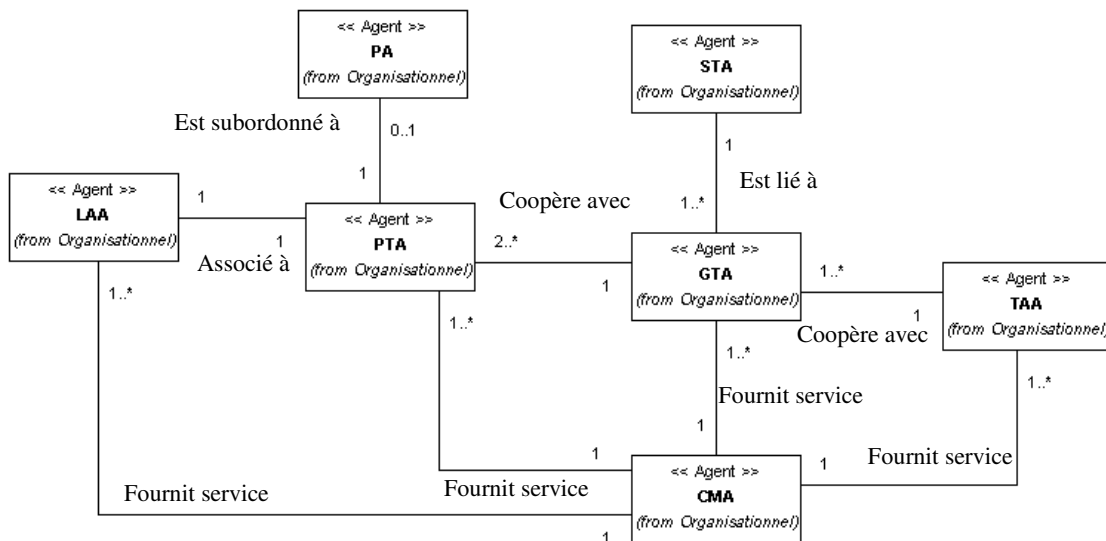


Figure IV.10 : Diagramme de classe des agents (niveau conceptuel)

Remarquons d'abord que tous nos agents jouent un et un seul rôle chacun.

A chaque apprenant est associé un exemplaire d'agent LAA qui communique avec le PTA de l'apprenant, donc ces deux agents se connaissent.

A chaque agent PTA peut correspondre ou non un agent PA qui lui est subordonné. PA ne connaît que le PTA auquel il est subordonné.

A chaque tuteur est associé un exemplaire d'agent TAA et pour chaque groupe qu'il tutorera il y aura un exemplaire d'agent GTA. Donc, TAA connaît tous les agents GTA(s) qui sont associés au tuteur en fonction des groupes qu'il suit pour un même processus (matière, module, ...etc).

L'agent CMA est un fournisseur de service pour tous les agents du système. Comme il est réactif, il ne fera que répondre à des sollicitations de ces derniers.

En matière d'architecture interne des agents, nous avons opté pour l'architecture BDI qui présente l'avantage d'être indépendante des outils d'implémentation comme c'est le cas de JAM par exemple qui bien que fondé sur un noyau BDI se rattache beaucoup aux technologies JAVA. Rappelons qu'un agent modélisé selon l'approche BDI possède des connaissances, des compétences et des intentions.

Les connaissances représentent ses croyances par rapport au monde dans lequel il évolue. Les compétences de l'agent sont représentées par l'ensemble des plans dont il dispose pour atteindre ses buts. Donc les plans d'un agent décrivent comment ce dernier peut manipuler les buts ou les événements qui surgissent. Les événements entre les agents véhiculés par les messages échangés entre eux (actes de communication). Enfin les buts qu'un agent se met à résoudre à un moment donné reflètent ses intentions qui sont un sous ensemble de l'ensemble de ses désirs.

Conformément à cela, nous allons donner une description de la structure de chacun de nos agents dans ce qui suivra. Une classe d'agent sera composé d'un sous ensemble de compartiment issus de la structure générale proposée par AUML (chapitre II). Certains compartiments étant plus reliés à des détails d'implémentation ne seront pas pris en compte dans cette phase de conception. On aura donc :

- ✓ Un compartiment attribut : contient la description de toutes les propriétés internes de l'agent.
- ✓ Un compartiment connaissance : contient ou fait référence à ce que l'agent a ou peut avoir comme connaissance. Indiquons toutefois que cet aspect étant vaste nous nous suffirons à donner des indications d'ordre général.
- ✓ Un compartiment action : qui représentera tous les plans d'un agent.
- ✓ Un compartiment méthode : contient l'ensemble des opérations d'un agent, il s'agira généralement d'un ensemble de services internes dont l'agent aura besoin pour accomplir ses tâches.
- ✓ Un compartiment états qui représentera tous les états d'un agent.
- ✓ Un compartiment envoi et réception de message : contient l'ensemble des actes de communication de l'agent en entrée et en sortie.



IV.5.5.1 Structure de l'agent LAA (LAA)

Les principaux échanges de cet agent se feront avec les agents PTA et CMA et éventuellement d'autres agents LAA.

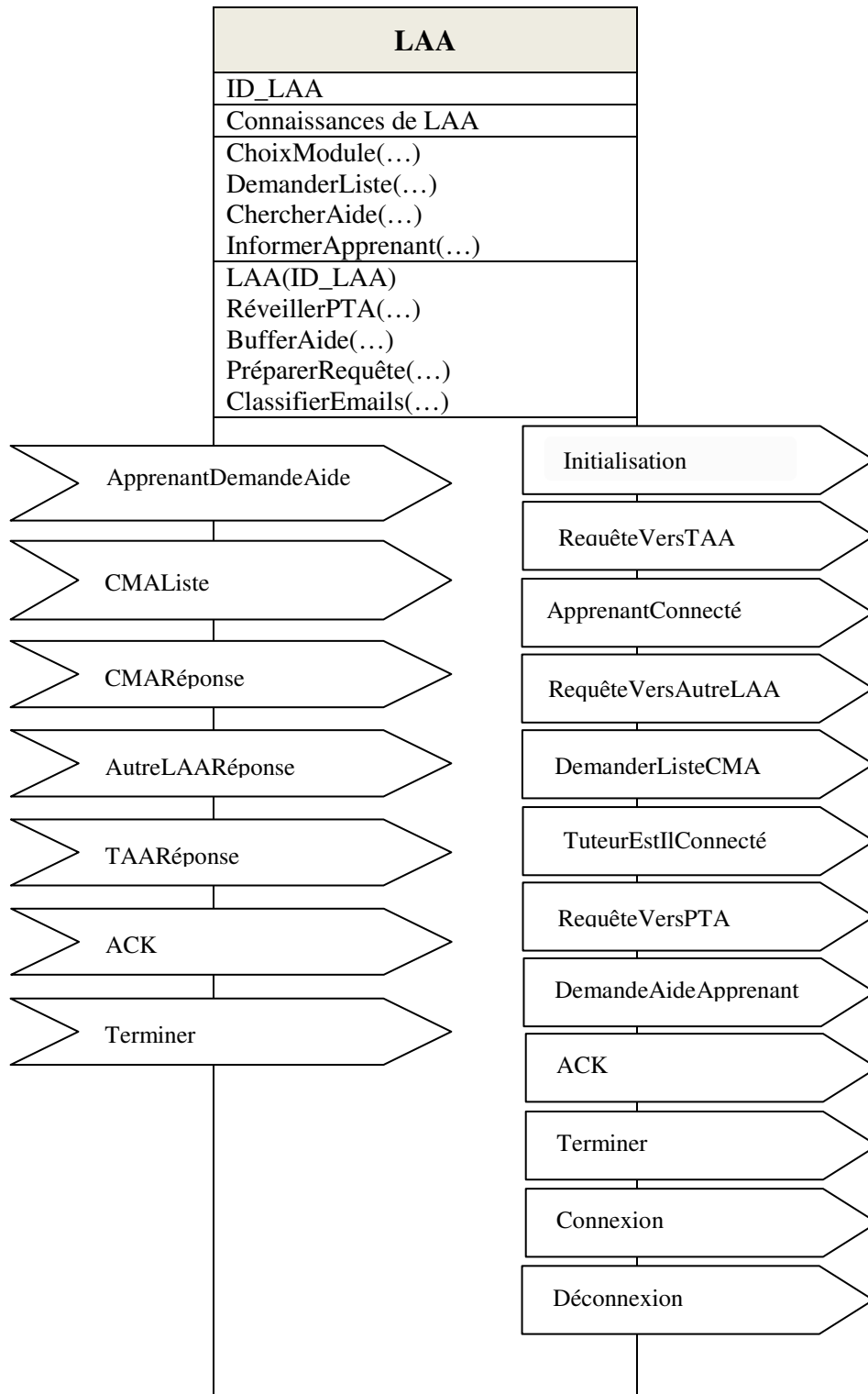


Figure IV.11 : Classe AUML de l'agent LAA

Les attributs :

- ID_LAA : est l'identificateur de LAA auprès des autres agents.

Les connaissances :

- LAA connaît les adresses de l'agent PTA lui correspondant
- Il a les connaissances de base du profil de l'apprenant au départ de l'apprentissage de ce dernier, mais LAA enrichira ses connaissances sur lui au fur et à mesure de l'exercice de ce dernier.

Les plans :

- **ChoixModule(...)** : ce plan s'exécutera dès que l'agent sera réveillé, i-e, dès la connexion de l'apprenant à la plate forme. LAA préparera l'interface appropriée pour aider l'apprenant à choisir la matière d'étude. Au fait, juste après l'authentification de l'apprenant, le système activera l'agent LAA qui prendra la main pour le reste des étapes pour une session d'apprentissage.
- **DemanderListe(...)** : LAA exécutera ce plan pour demander la liste des autres agents LAA(s) et TAA connectés à un moment donné.
- **ChercherAide(...)** : ce plan s'exécutera juste après avoir obtenu la liste des agents connecté, i-e, LAA fera son investigation pour trouver les apprenants prêts à offrir leur aide.
- **InformierApprenant(...)** : l'agent LAA informera l'apprenant par ce plan. Ces réponses seront décidées selon les messages que recevra l'agent LAA des autres agents, i-e, la réponse peut être affirmative ou négative. LAA possède la capacité de distinguer entre les différentes situations.

Les méthodes ou opérations :

- **LAA(ID_LAA)** : c'est le constructeur de l'agent LAA qui initialisera et restaurera éventuellement le dernier contexte valable de ce dernier.
- **RéveillerPTA(...)** : dès son lancement, LAA réveillera le PTA associé à l'apprenant en lui fournissant les informations nécessaires.
- **BufferAide(...)** : cette opération préparera l'interface du buffer à travers lequel l'apprenant exprimera sa requête de demande d'aide.
- **PréparerRequête(...)** : à l'aide de cette opération, LAA formatera la requête selon le destinataire prévu avant de l'envoyer.
- **ClassifierEmails(...)** : cette opération permettra à LAA de classifier les emails par degré d'importance en vue de les présenter à l'apprenant.

Les messages entrants :

- **ApprenantDemandeAide** : ce message parviendra de l'apprenant lorsqu'il sollicitera de l'aide.
- **CMAListe** : ce message parviendra de l'agent CMA lorsqu'il répondra à la demande de liste des agents connectés (LAA(s) et TAA).
- **CMARéponse** : ce message parviendra de l'agent CMA pour informer LAA si le tuteur humain est connecté ou pas.
- **AutreLAARéponse** : ce message sera envoyé par tout autre agent LAA sollicité pour de l'aide. LAA sera capable d'analyser la réponse et de prendre les décisions adéquates.
- **TAARéponse** : ce message apportera la réponse par le biais de TAA si le tuteur humain est prêt à fournir son aide à l'apprenant ou pas.

- **ACK** : ce message est reçu comme accusé réception à certains messages émis par LAA.
- **Terminer** : ce message demandera à LAA de s'arrêter dès que l'apprenant se déconnectera.

Les messages sortants :

- **Initialisation** : ce message sera envoyé pour réveiller l'agent PTA en lui fournissant les informations nécessaires.
- **RequêteVersTAA** : ce message est utilisé pour aiguiller une demande d'aide émanant de l'apprenant vers le tuteur humain.
- **ApprenantConnecté** : ce message sera envoyé pour informer que l'apprenant est prêt à aider un autre apprenant en cas de sollicitation.
- **RequêteVersAutreLAA** : permet d'aiguiller la requête vers un apprenant particulier sollicité par l'apprenant émetteur.
- **DemandeListeCMA** : ce message adressé à CMA correspond à la demande de la liste des agents connectés à un moment donné.
- **TuteurEstIlConnecté** : ce message part vers CMA afin de lui demander si le tuteur est connecté.
- **RequêteVersPTA** : ce message sera utilisé pour aiguiller la requête de l'apprenant vers son agent PTA.
- **DemandeAideApprenant** : ce message est utilisé pour solliciter l'aide d'un apprenant, i-e, savoir si l'apprenant veut bien fournir son aide.
- **ACK** : pour certains types de messages reçus, LAA émettra un accusé réception.
- **Terminer** : ce message sera acheminé vers PTA pour lui demander de s'arrêter car l'apprenant s'est déconnecté.
- **Connexion** : sera envoyé pour informer CMA de la connexion de l'apprenant.
- **Déconnexion** : sera envoyé pour informer CMA de la déconnexion de l'apprenant.



IV.5.5.2 Structure de l'agent PTA (PTA)

Les principaux échanges de cet agent se feront avec les agents LAA et CMA et GTA.

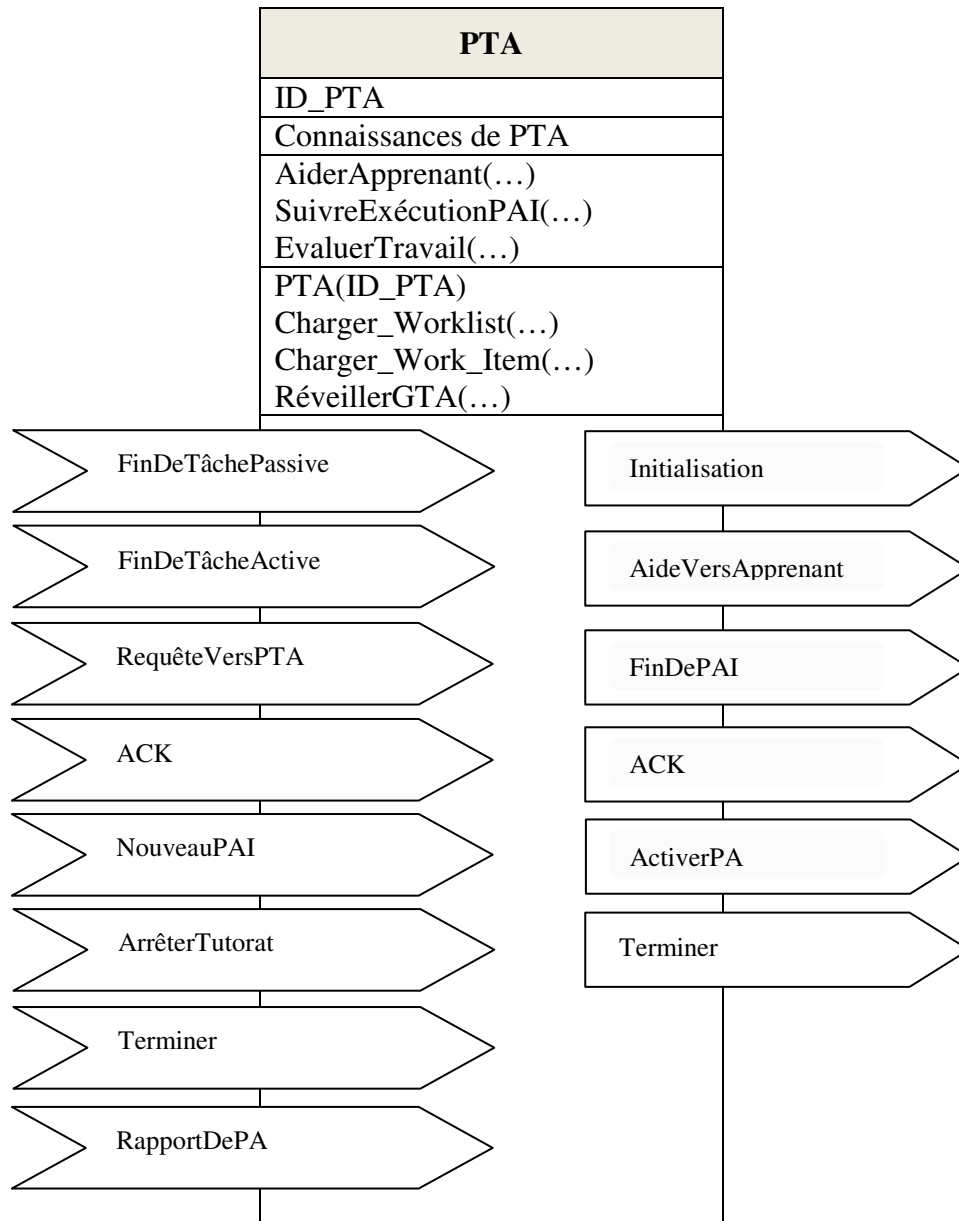


Figure IV.12 : Classe AUML de l'agent PTA

Les attributs :

- ID_PTA : est l'identificateur de PTA auprès des autres agents.

Les connaissances :

- PTA connaît les adresses des agents LAA, CMA et GTA lui correspondant.
- Il connaît l'ontologie de processus, particulièrement ce qui est associé aux PAI(s).

- Il connaît l'ontologie de la matière à suivre.
- Il a les connaissances nécessaires des décisions à prendre durant sa session de tutorat.

Les plans :

- **AiderApprenant(...)** : ce plan procurera de l'aide que l'apprenant sollicitera.
- **SuivreExécutionPAI(...)** : ce plan se chargera de suivre le déroulement de l'instance de PAI en termes de sous-processus.
- **EvaluerTravail(...)** : chaque tâche ou activité accomplie sera évaluée par ce plan. En cas de tâche passive, l'évaluation consistera à noter le fait que la tâche a été accomplie et si elle a été sautée, ce fait sera aussi pris en compte. Pour toute tâche active, PTA procèdera à son évaluation selon ce qui est prévu dans le plan.

Les méthodes ou opérations :

- **PTA(ID_PTA)** : c'est le constructeur de l'agent PTA qui initialisera et restaurera éventuellement le dernier contexte valable de ce dernier.
- **Charger_Worklist(...)** : cette opération récupérera le contexte de l'activité en cours d'exécution par l'apprenant et chargera la worklist de ce dernier.
- **Charger_Work_Item(...)** : chargera le bon de travail correspondant à la tâche à accomplir par l'apprenant.
- **RéveillerGTA(...)** : cette opération tentera de réveiller GTA si ce dernier ne l'est pas encore. C'est le cas par exemple, si l'apprenant est le premier de son groupe à se connecter.

Les messages entrants :

- **FinDeTâchePassive** : ce message parviendra de l'apprenant à chaque fin de tâche passive. Notons que l'événement portera l'information s'il s'agit d'une fin normale ou pas.
- **FinDeTâcheActive** : chaque fin de tâche active se terminera par un feedback de l'apprenant qui se soldera par ce message.
- **RequêteVersPTA** : ce message exprime une requête de demande d'aide parvenant de l'apprenant vers son tuteur artificiel.
- **ACK** : accusé réception qui arrivera par exemple lorsque GTA sera réveillé.
- **NouveauPAI** : la réception d'un nouveau PAI de la part de GTA est notifiée par ce message.
- **ArrêterTutorat** : lorsqu'il n'y aura plus de PAI à suivre ou exceptionnellement si le tuteur humain le souhaite, ce message demandera l'arrêt du tutorat artificiel.
- **Terminer** : lorsque l'apprenant se déconnecte, ce message d'arrêt parviendra de l'agent LAA pour que PTA se désactive.
- **RapportDePA** : dès que PA termine son travail il envoi un rapport qui sera véhiculé par ce message.

Les messages sortants :

- **Initialisation** : ce message sera envoyé pour réveiller l'agent GTA (s'il ne l'est pas encore) en lui fournissant les informations nécessaires.
- **AideVersApprenant** : ce message portera l'aide proposée à l'apprenant qui la sollicite de PTA.
- **FinDePAI** : toute fin d'un PAI sera signalée par ce message.
- **ACK** : accusé réception envoyé par exemple lorsque PTA est activé.

- **ActiverPA** : ce message est envoyé pour activer l'agent PA si PTA juge nécessaire de le faire intervenir.
- **Terminer** : message demandant l'arrêt à PA ou signalant l'arrêt à GTA.



IV.5.5.3 Structure de l'agent PA (PA)

Les principaux échanges de cet agent se feront avec l'agent PTA auquel il est subordonné. Cependant, durant sa session de suivi de l'apprenant d'autres types d'échanges avec l'apprenant auront lieu, mais nous omettons de les mettre ici car ils n'ont pas encore été bien détaillés.

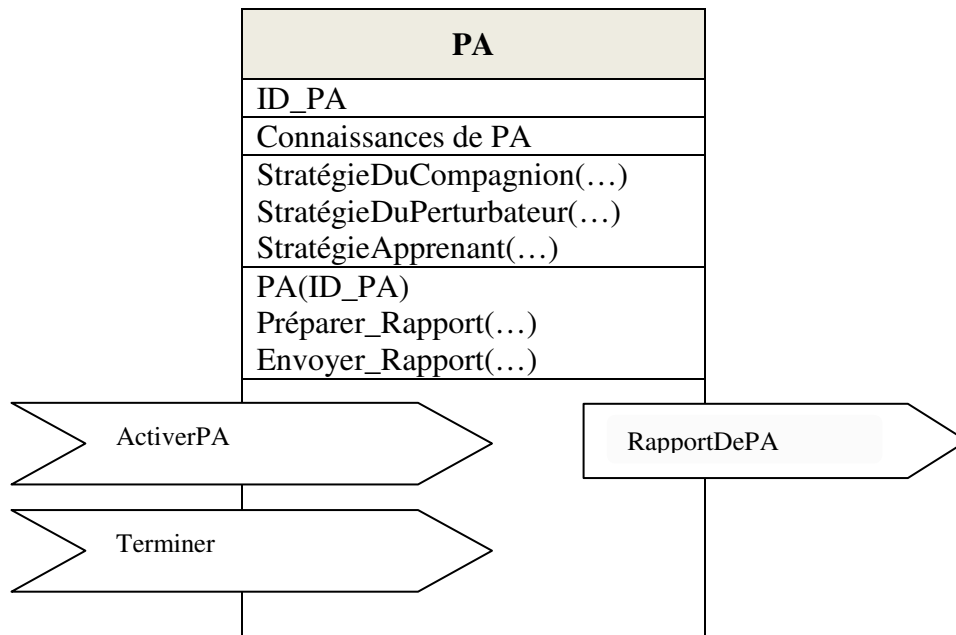


Figure IV.13 : Classe AUMI de l'agent PA

Les attributs :

- **ID_PA** : est l'identificateur de PA auprès de l'agent PTA.

Les connaissances :

- PA connaît la matière à enseigner. De plus, il connaît les diverses stratégies pédagogiques et la manière de les appliquer.

Les plans :

- **StratégieDuCompagnion(...)** : ce plan sera exécuté pour appliquer la stratégie "Apprendre avec un compagnon" comme expliqué au début de ce chapitre.
- **StratégieDuPerturbateur(...)** : plan permettant d'appliquer la stratégie du "Perturbateur".
- **StratégieApprenant(...)** : ce plan permet l'application de la stratégie "Apprendre en enseignant".

Les méthodes :

- **PA(ID_PA)** : constructeur de l'agent PA.

- **Préparer_Rapport(...)** : cette opération permettra de préparer et de mettre en forme le rapport à envoyer à PTA.
- **Envoyer_Rapport(...)** : permettra l'envoi du rapport vers PTA.

Les messages entrants :

- **ActiverPA** : ce message activera l'agent PA dès son arrivée.
- **Terminer** : l'arrivée de ce message ordonnera à l'agent PA de s'arrêter.

Les messages sortants :

- **RapportDePA** : ce message est envoyé soit après une terminaison de travail normal soit suite à la demande d'arrêt émanant de PTA.



IV.5.5.4 Structure de l'agent TAA (TAA)

Les principaux échanges de cet agent se feront avec les agents GTA et CMA et éventuellement d'autres agents TAA.

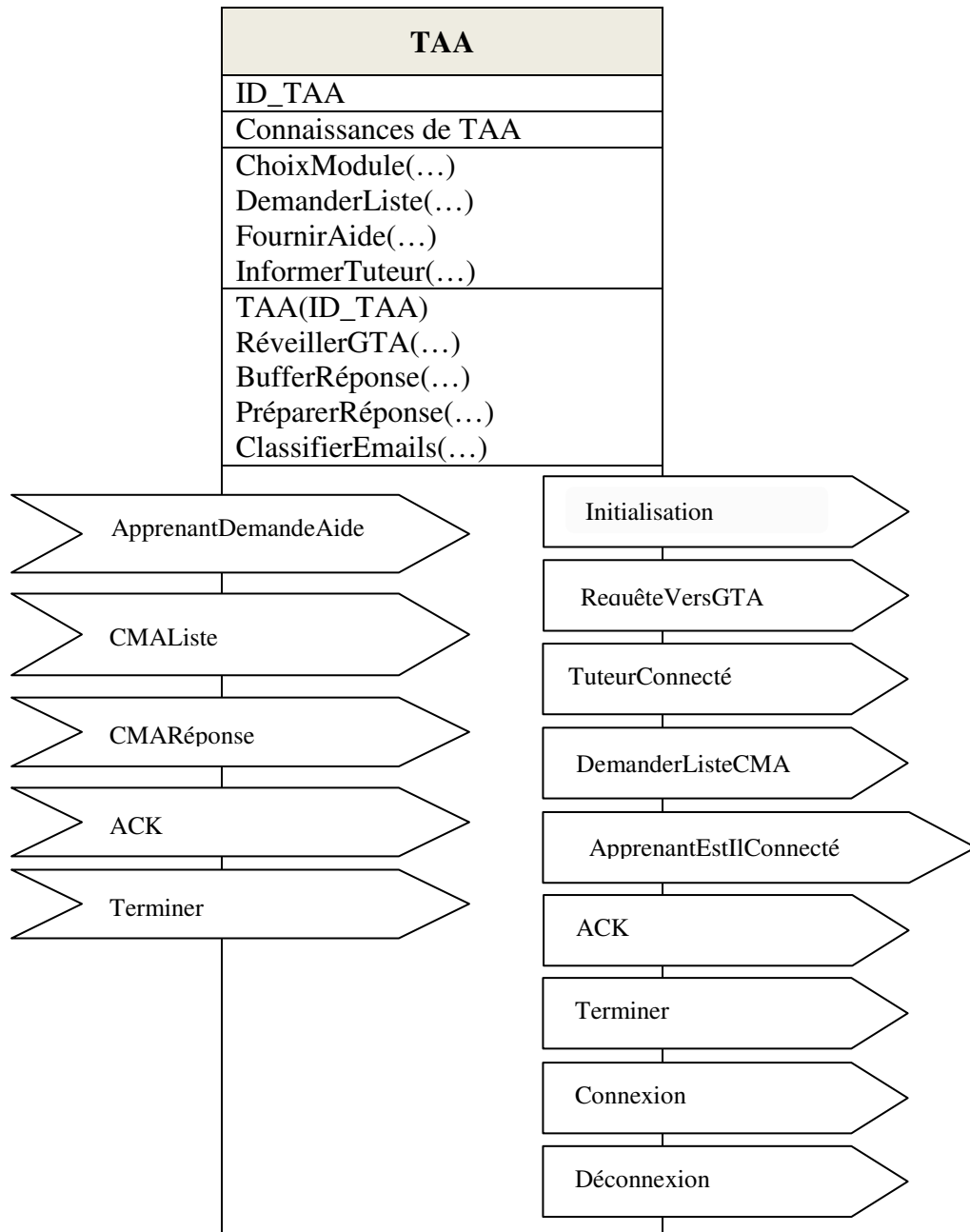


Figure IV.14 : Classe AUML de l'agent TAA

Les attributs :

- ID_TAA : est l'identificateur de TAA auprès des autres agents.

Les connaissances :

- TAA connaît les adresses de l'agent GTA lui correspondant.

- Il a les connaissances de base du profil du tuteur au départ du tutorat de ce dernier, mais TAA enrichira ses connaissances sur lui au fur et à mesure de l'exercice de ce dernier.

Les plans :

- **ChoixModule(...)** : ce plan s'exécutera dès que l'agent sera réveillé, i-e, dès la connexion du tuteur à la plate forme. TAA préparera l'interface appropriée pour aider le tuteur à choisir la matière à suivre. Au fait, juste après l'authentification du tuteur, le système activera l'agent TAA qui prendra la main pour le reste des étapes pour une session de tutorat.
- **DemanderListe(...)** : TAA exécutera ce plan pour demander la liste des agents LAA(s) connectés à un moment donné (LAA(s) appartenant au même GTA).
- **FournirAide(...)** : ce plan s'exécutera après avoir reçu une requête de demande d'aide de la part d'un apprenant par le biais de son agent LAA. Ainsi, le tuteur acheminera sa réponse via son agent TAA.
- **Informertuteur(...)** : l'agent TAA informera le tuteur par ce plan en lui présentant la requête dans un format approprié. TAA possède la capacité de distinguer entre les différentes situations.

Les méthodes ou opérations :

- **TAA(ID_TAA)** : c'est le constructeur de l'agent TAA qui initialisera et restaurera éventuellement le dernier contexte valable de ce dernier.
- **RéveillerGTA(...)** : dès son lancement, TAA réveillera le GTA associé au groupe et à la matière choisie par le tuteur en lui fournissant les informations nécessaires. Bien sûr GTA peut très bien avoir été réveillé par le premier apprenant qui s'est connecté, dans ce cas un accusé réception parviendra vers TAA.
- **BufferRéponse(...)** : cette opération préparera l'interface du buffer de réponse à travers lequel le tuteur exprimera sa réponse à une demande d'aide.
- **PréparerRéponse(...)** : à l'aide de cette opération, TAA formatera la requête selon le destinataire prévu avant de l'envoyer.
- **ClassifierEmails(...)** : cette opération permettra à TAA de classifier les emails par degré d'importance en vue de les présenter au tuteur. Cela permettra à ce dernier de juger des cas les plus urgents à prendre en charge.

Les messages entrants :

- **ApprenantDemandeAide** : ce message parviendra de l'apprenant par le biais de son agent LAA ou PTA selon les situations, lorsqu'il sollicitera de l'aide.
- **CMAListe** : ce message parviendra de l'agent CMA lorsqu'il répondra à la demande de liste des agents connectés (LAA(s) et TAA).
- **CMARéponse** : ce message parviendra de l'agent CMA pour informer LAA si le tuteur humain est connecté ou pas.
- **ACK** : ce message est reçu comme accusé réception à certains messages émis par TAA.
- **Terminer** : ce message demandera à TAA de s'arrêter dès que le tuteur se déconnectera.

Les messages sortants :

- **Initialisation** : ce message sera envoyé pour réveiller l'agent GTA en lui fournissant les informations nécessaires.
- **RequêteVersGTA** : ce message est utilisé pour aiguiller une réponse à une demande d'aide ou une demande d'information émanant du tuteur humain.
- **TuteurConnecté** : ce message sera envoyé pour informer que le tuteur est prêt à aider un autre apprenant en cas de sollicitation.
- **DemandeListeCMA** : ce message adressé à CMA correspond à la demande de la liste des acteurs (agents) connectés à un moment donné.
- **ApprenantEstIlConnecté** : ce message part vers CMA afin de lui demander si un apprenant particulier est connecté.
- **ACK** : pour certains types de messages reçus, TAA émettra un accusé réception.
- **Terminer** : ce message sera acheminé vers GTA pour lui demander de s'arrêter car le tuteur s'est déconnecté. Cependant, GTA peut ne pas s'arrêter s'il reste des apprenants de son groupe encore connectés.
- **Connexion** : sera envoyé pour informer CMA de la connexion du tuteur.
- **Déconnexion** : sera envoyé pour informer CMA de la déconnexion du tuteur.



IV.5.5.5 Structure de l'agent GTA (GTA)

Les principaux échanges de cet agent se feront avec les agents PTA, CMA et l'agent TAA représentant le tuteur du groupe.

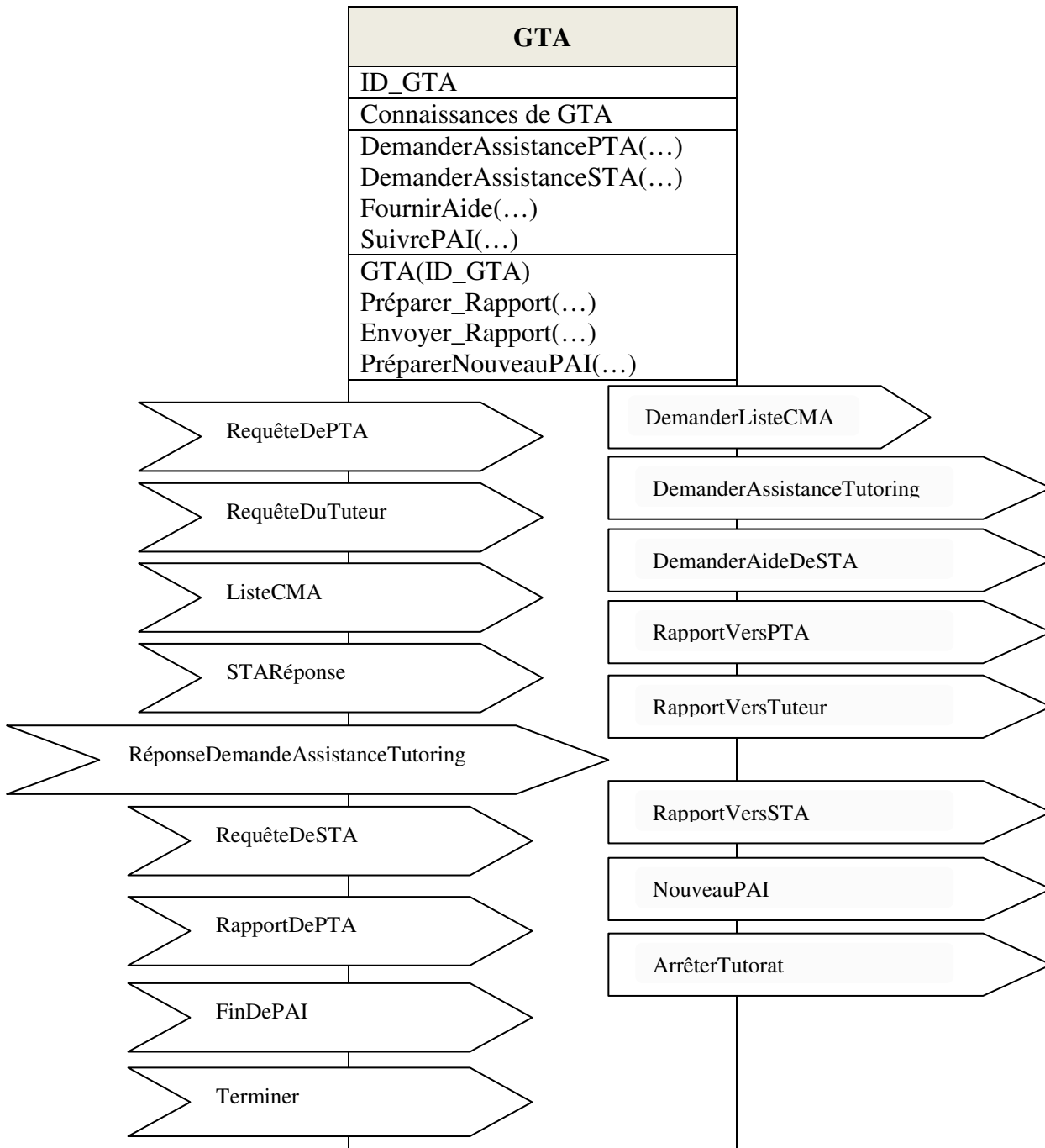


Figure IV.15 : Classe AUML de l'agent GTA

Les attributs :

- ID_GTA : est l'identificateur de GTA auprès des autres agents.

Les connaissances :

- GTA connaît tous les agents avec lesquels il coopère.
- Connaît l'ontologie de processus workflow.
- Connaît l'ontologie de la matière à tutorer ainsi que les décisions tutorielles à prendre.

Les plans :

- **DemanderAssistancePTA(...)** : ce plan s'exécutera lorsque GTA voudra solliciter l'aide d'agents PTA de son groupe pour le compte d'un certain agent PTA du même groupe.
- **DemanderAssistanceSTA(...)** : si le plan précédent échoue alors GTA sollicitera l'aide du super-tuteur STA.
- **FournirAide(...)** : ce plan est lancé pour fournir de l'aide aux demandeurs selon l'expertise et l'expérience de GTA.
- **SuivrePAI(...)** : dès qu'un nouveau PAI est lancé, ce plan permettra de suivre l'instance en cours pour un apprenant donné. Ce plan permettra de suivre autant d'instances qu'il y a d'apprenants dans le groupe.

Les méthodes :

- **GTA(ID_GTA)** : constructeur de l'agent GTA.
- **Préparer_Rapport(...)** : avant tout envoi de rapport, GTA procédera à son analyse et mise en forme selon le destinataire.
- **Envoyer_Rapport(...)** : permet d'envoyer le rapport fin prêt.
- **PréparerNouveauPAI(...)** : consiste à préparer le contexte, les ressources, et les worklists pour la nouvelle instance de PAI.

Les messages entrants :

- **RequêteDePTA** : demande d'assistance provenant de l'agent PTA.
- **RequêteDuTuteur** : requête provenant du tuteur par le biais de l'agent TAA.
- **ListeCMA** : liste des connectés fournie par l'agent CMA.
- **STARéponse** : la réponse de STA à une demande d'aide émanant de GTA.
- **RéponseDemandeAssistanceTutoring** : ce message parviendra de tout PTA ayant reçu la demande d'aide diffusé par GTA pour le compte d'un PTA particulier.
- **RequêteDeSTA** : demande émanant de STA pour le compte d'un autre GTA.
- **RapportDePTA** : ce message véhiculera le rapport pédagogique établi par un agent PTA soit à la fin d'un PAI ou bien à la fin d'une session d'apprentissage, i-e, la déconnexion d'un apprenant.
- **FinDePAI** : ce message signalera la fin d'un PAI et la demande de la suite, i-e, ce qu'il y a lieu de faire.
- **Terminer** : ce message demandera à GTA de se terminer. Si plus personne n'est connecté, GTA de désactivera.

Les messages sortants :

- **DemandeListeCMA** : demande de la liste des connectés à l'agent CMA.
- **DemanderAssistanceTutoring** : message diffusé pour un ensemble d'agents PTA pour obtenir de l'aide (pour le compte d'un autre agent PTA du groupe).
- **DemanderAideDeSTA** : demande d'aide à STA.
- **RapportVersPTA** : rapport envoyé vers PTA.
- **RapportVersTuteur** : rapport destiné au tuteur.

- **RapportVersSTA** : rapport destiné à STA.
- **NouveauPAI** : envoi d'un nouveau PAI.
- **ArrêterTutorat** : demande d'arrêt de tutorat destinée à un agent PTA.



IV.5.5.6 Structure de l'agent CMA (CMA)

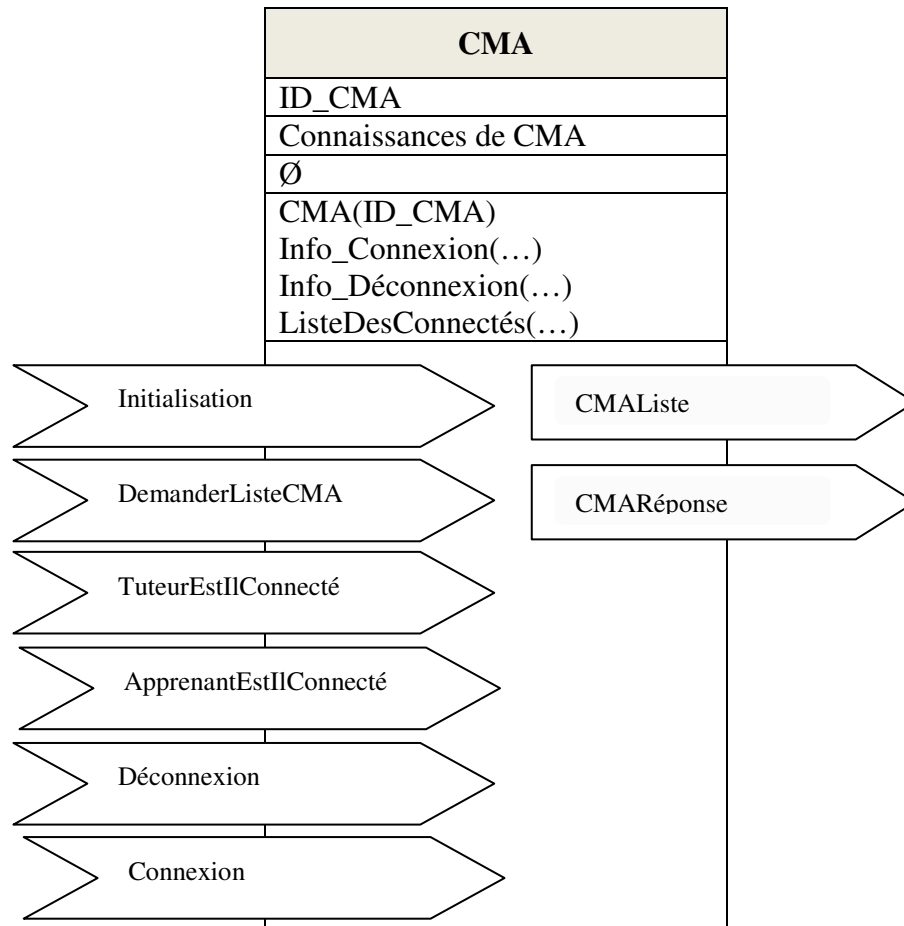


Figure IV.16 : Classe AUML de l'agent CMA

Les attributs :

- **ID_CMA** : est l'identificateur de CMA auprès des autres agents.

Les connaissances :

- Les connaissances de CMA consistent en l'accès à la base de données contenant les informations sur les acteurs artificiels et humains du système.

Les plans :

- Ne possède aucun plan car c'est un agent réactif qui fournira ses services suite à des sollicitations externes.

Les méthodes :

- **CMA(ID_CMA)** : constructeur de l'agent CMA.
- **Info_Connexion(...)** : cette méthode fournira l'information si un acteur particulier est connecté ou pas.

- **Info_Déconnexion(...)** : cette opération permettra de mettre à jour l'historique des connexions/déconnexions des acteurs humains (tuteurs, apprenants).
- **ListeDesConnectés(...)** : permet de préparer la liste des acteurs connectés à un moment donné pour l'envoyer au destinataire.

Les messages entrants :

- **Initialisation** : ce message parviendra du système à l'agent CMA afin de l'activer (en cas de démarrage ou redémarrage du système par exemple).
- **DemanderListeCMA** : message demandant la liste des acteurs connectés.
- **TuteurEstIlConnecté** : message demandant si le tuteur humain est connecté.
- **ApprenantEstIlConnecté** : message demandant si un apprenant est connecté.
- **Déconnexion** : ce message parviendra à chaque fois qu'un acteur humain se déconnectera.
- **Connexion** : ce message parvient dès la connexion d'un acteur humain.

Les messages sortants :

- **CMAListe** : message fournissant la liste acteurs connectés.
- **CMARéponse** : fournit la réponse si le tuteur ou un apprenant est connecté.

IV.6 Conclusion

Le langage AUMML nous a permis de décrire diverses facettes de notre SMA destiné à l'assistance des acteurs humains. Les différents actes de communications ont été spécifiés précisant les divers échanges entre agents.

Cependant, des efforts restent à faire quand à la modélisation des connaissances relatives aux agents de type cognitif. Nous comptons étayer cet aspect dans nos futurs travaux en utilisant des bases de connaissances appropriées plus les ontologies nécessaires à leur bonne exploitation.

L'acquisition récente de l'outil de développement de SMA, "*AgentBuilder*" nous permettra de faire une première implémentation de notre système multi-agent en utilisant les ontologies pour la partie connaissance et un système d'inférence à base de règles pour la partie raisonnement des agents.

Le prochain chapitre présentera l'architecture générale de l'infrastructure proposée ainsi que quelques outils mis en œuvre dans le cadre de ce projet.

Chapitre V

**Architecture générale et
présentation de quelques outils**



Chapitre V

Mise en œuvre d'outils dédiés à l'infrastructure proposée

V.1 Architecture générale de l'infrastructure

La mise en œuvre du système proposé nécessite l'implémentation de différents outils destinés à couvrir les besoins des utilisateurs qui sont principalement les tuteurs et les auteurs. Le schéma d'architecture en figure V.1 met en évidence les liens entre les différents modules du système.

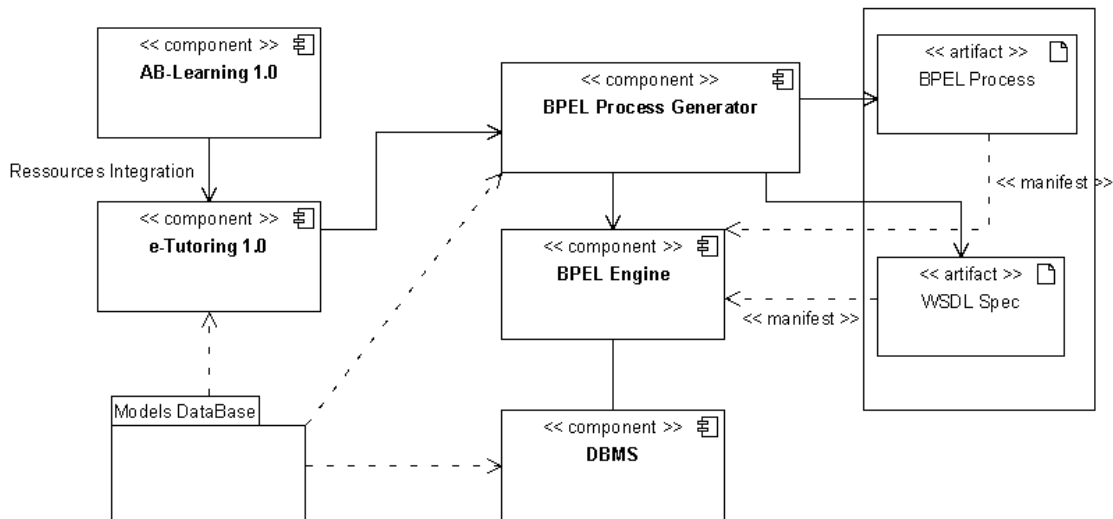


Figure V.1: Architecture Générale sans le SMA

L'outil **AB-Learning 1.0** est un outil auteur destiné à la création de contenus pédagogiques tels que les cours et les exercices sous diverses formes comme les QCM(s) par exemple (Mahdaoui & A11, 2004). Il est destiné pour les enseignants auteurs, mais les tuteurs peuvent aussi y accéder notamment pour préparer des exercices. Les contenus peuvent être utilisés pour alimenter la partie informationnelle des modèles de processus de tutorat.

L'outil **e-Tutoring 1.0** est un outil permettant la création de modèles de PAI, les PAC(s) ne sont pas encore pris en charge par l'outil. e-Tutoring 1.0 a été développé avec HTML/XML et le langage Php. Les données relatives aux modèles créés sont stockés dans une base de données "MySQL" (<http://www.mysql.com>) et les processus sont générés en BPEL par le module "BPEL Process Generator" qui produit les modèles de processus et les fichiers web services (WSDL) nécessaires pour l'exécution des modèles. Les rôles concernés que sont le tuteur et l'apprenant peuvent accéder à la plateforme via le web service approprié en utilisant un browser. Enfin, la communication entre tuteurs et apprenants se fait par e-mails ou par notification.

Comme nous pouvons le constater, cette architecture n'intègre pas encore le système multi-agent pour les raisons suivantes :

- ❖ Le développement du SMA doit faire l'objet de tests à part à fin de s'assurer de la cohérence du fonctionnement des agents entre eux dans un contexte libre et indépendant du workflow. Viendrait alors l'étape d'intégration des agents dans le système global.
- ❖ Les outils permettant la mise en oeuvre du SMA n'ont été acquis que récemment, il s'agit du produit *AgentBuilder* que nous comptons utiliser pour le développement d'un premier prototype. Des essais ont été faits avec l'outil *AgentTool* qui est l'outil associé à MaSE, chose qui a d'ailleurs nécessité que l'on fasse une conception avec ladite méthode. Le problème avec MaSE est qu'elle procède par affinage d'objectifs jusqu'au niveau où chaque objectif soit couvert par un agent : ceci désoriente complètement la composante en agent que nous avons adopté. Par exemple, un agent comme PTA risque de se trouver éclaté en plusieurs agents chose qui n'arrange pas notre approche.

Pour la phase expérimentale de l'infrastructure, nous utilisons une architecture client/serveur où un seul serveur prend en charge tous les module. Avec un nombre assez limité d'instances, cela ne pose pas de problèmes. Mais il est préférable de penser à une architecture trois tiers pour de meilleures performances du système surtout pour un déploiement à large échelle. La figure suivante donne une description globale de la plate forme intégrant le système workflow et les SMA.

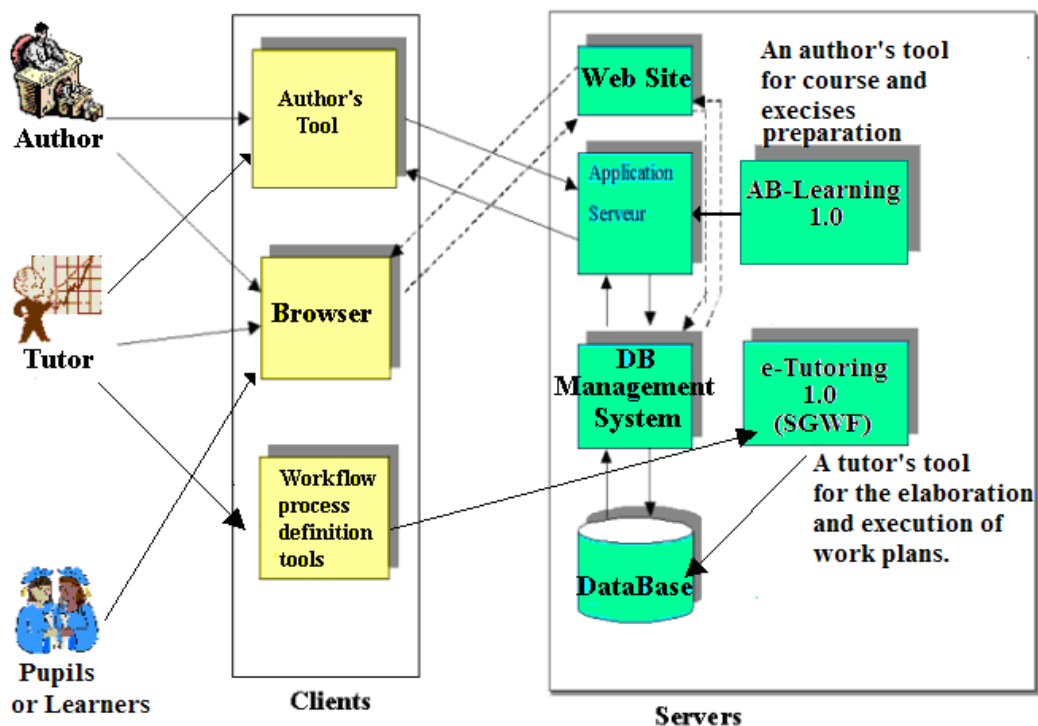


Figure V.2 : Architecture avec le SMA

V.2 L'outil AB-Learning 1.0

La version 1.0 du produit a été développée en Delphi 7.0 Client/Serveur. L'outil peut être utilisé pour préparer les cours, les exercices et les devoirs. Il est utile pour préparer les ressources pédagogiques nécessaires à un processus de tutorat.

AB-Learning 1.0 se compose de deux modules : le premier permet la création et la représentation des documents pédagogique et le deuxième outil permet la prise en charge de stratégies pédagogiques lors de l'élaboration des contenus de type test et exercices utilisable aussi par le tuteur. La figure V.3 représente l'interface qui permet la création de cours.

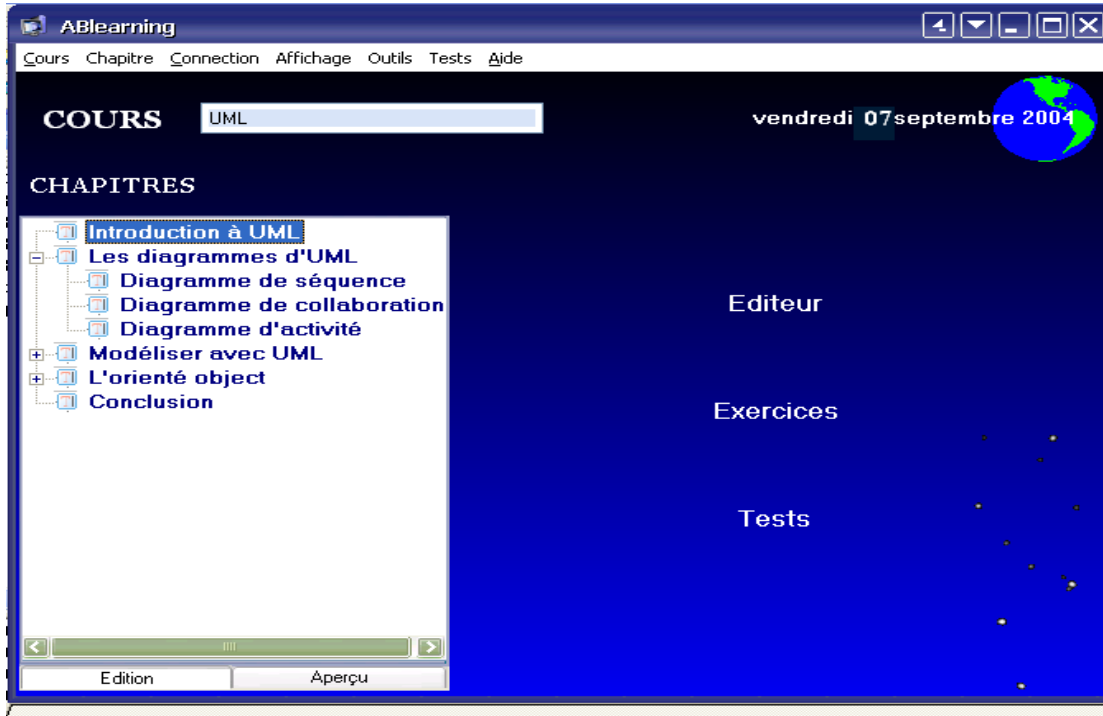


Figure V.3 : Interface de création d'un cours.

La scénarisation du contenu est possible et permet d'établir une structure logique de la composition d'un document pédagogique. L'enseignant aura ainsi la possibilité de définir toutes les parties du cours (chapitres, exercices et tests) et d'établir les liens entre eux.

Après avoir scénarisé le cours ou le test, l'auteur pourra éditer le contenu de chaque partie. Ce module comporte tous les outils nécessaires à l'édition comme l'insertion d'image, du texte, du son, ...etc.

A tout support de cours est associé des exercices et des tests d'évaluations. Cette fonctionnalité donne la possibilité d'insérer des exercices d'auto-évaluation, pour chaque partie de cours ou après une session d'apprentissage afin de permettre à l'apprenant de tester les connaissances acquises.

L'application de stratégies pédagogiques est une fonctionnalité qui donne aux enseignants (tuteurs) la possibilité de suivre les apprenants en appliquant quelques stratégies pédagogiques sur les cours, les exercices ou les tests d'évaluations. En d'autres termes, elle permet le suivi des apprenants. La figure V.4 montre l'interface que verra le tuteur lors de son accès :



Figure V.4 : Page d'accueil du tuteur.

Enfin, en matière de gestion des documents pédagogiques, l'auteur et le tuteur ont la possibilité de :

- Rescénariser le cours.
- Créer, modifier ou supprimer des : exercices, devoir, TP et diverses ressources comme les résumés, corrigés de test, guide pédagogique, ...etc.
- Télécharger un outil permettant de créer des exercices.
- Télécharger un outil permettant de créer un guide pédagogique.

V.3 L'outil e-Tutoring 1.0

Cet outil permet la définition de processus de type PAI. e-Tutoring 1.0 est doté d'interfaces simples et intuitives permettant à un tuteur de créer son processus sans vraiment être surchargé par la terminologie liée au workflow. Le tuteur peut ainsi créer ses PAI(s) en suivant un ensemble d'étapes. La figure V.5 présente la page web principale de l'outil à laquelle le tuteur accèdera après avoir fourni un login et un mot de passe.

Nous allons dans ce qui suit suivre les étapes de création d'un PAI associé à un cours intitulé "Product Management" dont l'un des PAI(s) se nomme "Products, Companies and Competitors Analysis" (Mahdaoui & al, 2008).

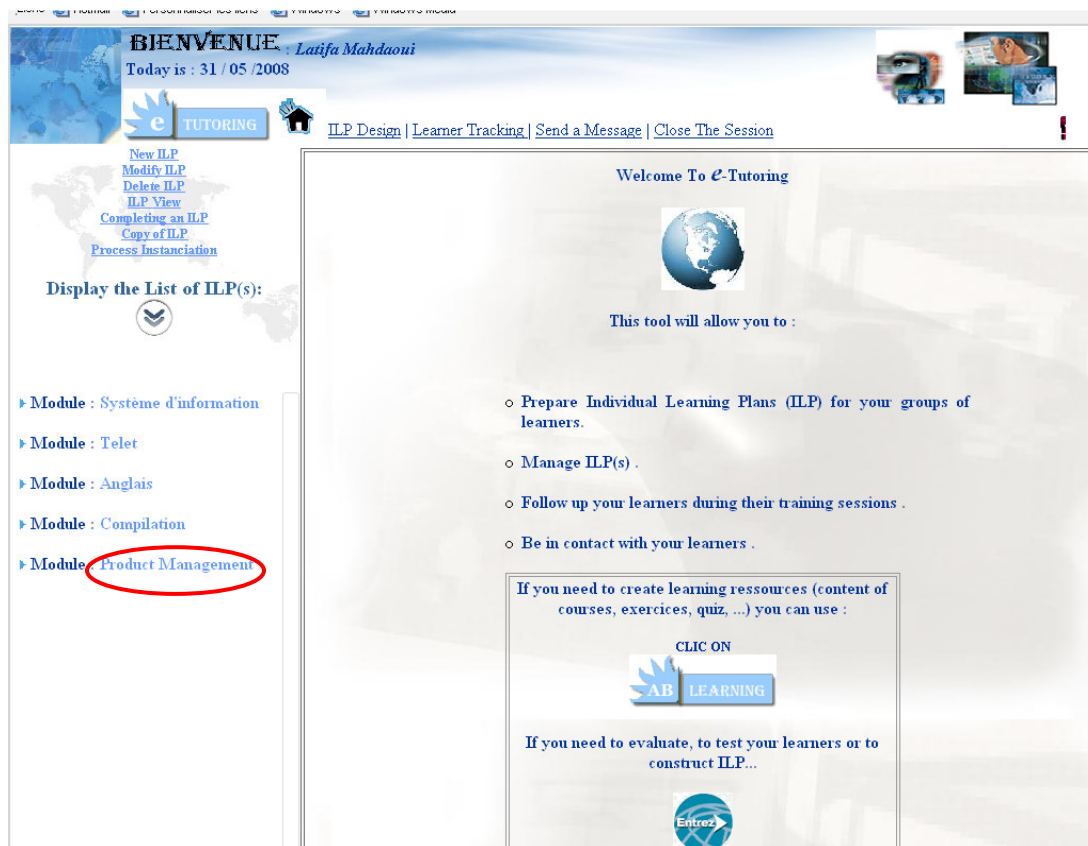


Figure V.5 : Page web principale de "e-tutoring 1.0".

Afin de préparer facilement le PAI, le tuteur doit suivre une série d'étapes proposées par le logiciel. Les étapes sont au nombre de quatre que les figures V.6 à V.11 décrivent brièvement.

L'étape 1 (figure V.6) permet au tuteur de décrire toutes les activités du processus aussi bien celle de l'apprenant que du tuteur.

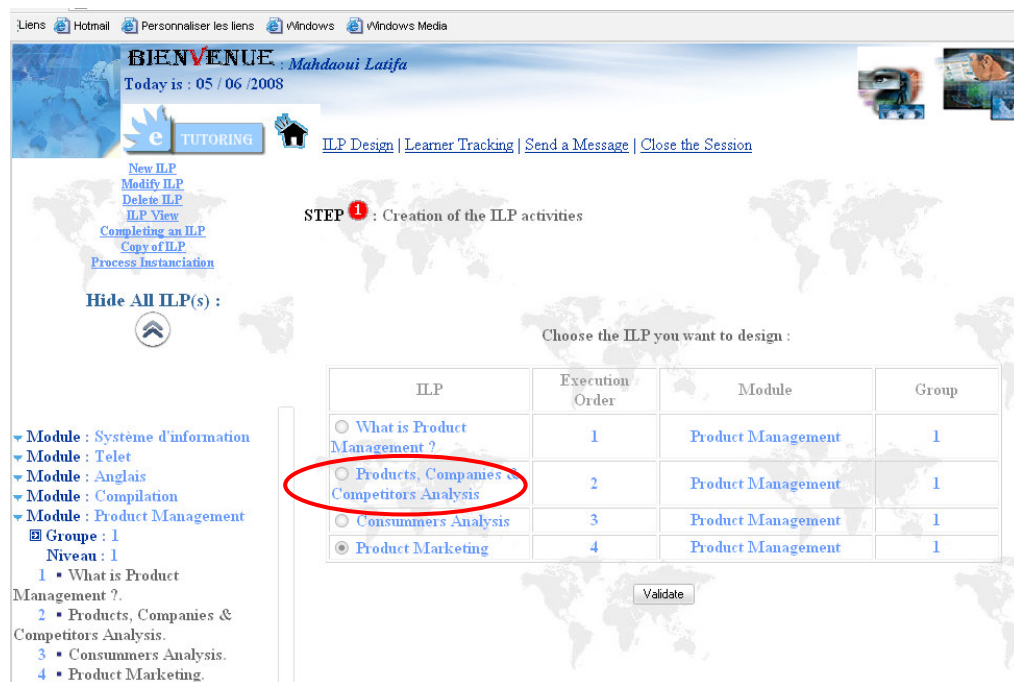


Figure V.6: Etape 1, création de la description du PAI

Suit alors la création des activités relatives au PAI comme le montre la figure V.7 :



Figure V.7: Etape 1, création des activités

Durant l'étape 2 (figure V.8), le tuteur pourra préciser les activités qui le concernent et le reste des activités sera automatiquement attribué à l'apprenant.



Figure V.8 : Etape 2, définir les activités du tuteur et de l'apprenant

L'étape 3 permet l'utilisation d'opérateurs de workflow pour organiser et connecter les activités (figure V.9):

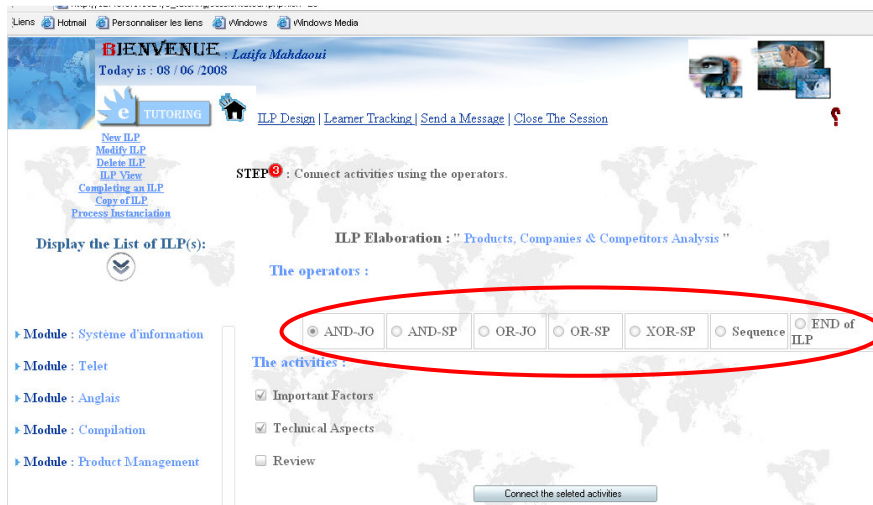


Figure V.9 : Etape 3, connecter les activités à l'aide des opérateurs

La figure V.10 relate la fin de l'étape 4 qui est pour décrire les tâches de chaque activité en détail. La fin de l'étape montre la liste des activités dont les tâches ont été spécifiées :

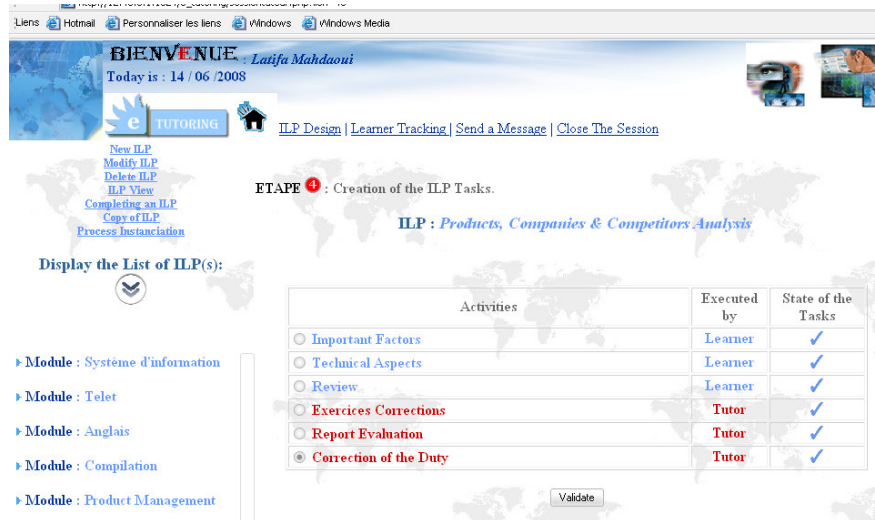


Figure V.10 : Etape 4, activités avec tâches spécifiées

Enfin, le modèle final peut être visualisé comme le montre la figure V.11 :

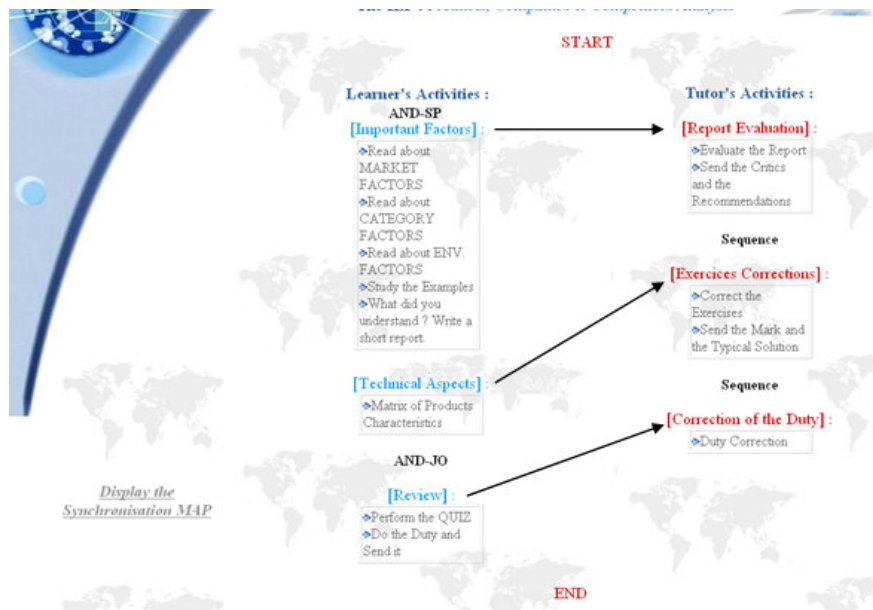


Figure V.11 : Visualisation du modèle de processus

A la fin, le modèle de processus pourra être généré en BPEL et stocké dans les fichiers appropriés. Le listing V.1 est une portion du code relatif au modèle de processus créé :

```
//-----  
//Process SP.1.2  
//-----  
<process name="SP12" ...>  
  
//-----  
//Actors/roles or Partner Links  
//-----  
<PartnerLinks>  
  //Learner  
  <partnerLink name="Learner"  
    partnerLinkType="client:SP12"  
    myRole="SP12Provider"  
    partnerRole="SP12Requester"/>  
  //Tutor  
  <partnerLink name="Tutor"  
    partnerLinkType="client:SP12"  
    myRole="SP12Provider"  
    partnerRole="SP12Requester"/>  
</PartnerLinks>  
  
//-----  
//Variables  
//-----  
<variables>  
...  
</variables>  
...  
//-----  
//Activities and control flow  
//-----  
<sequence name="SP.1.2">  
<flow name="AND-SP">  
  <sequence name="SSB1">  
    <scope name="A.1.2.1">  
      <sequence name="ImportantFactors">  
        <receive name="ReadAboutMarketFactors"  
          createInstance="yes" partnerLink="Learner"  
          portType="client:SP12"  
          operation="initiate"/>  
        <receive name="ReadAboutCategoryFactors"  
          createInstance="yes" partnerLink="Learner"  
          portType="client:SP12"  
          operation="initiate"/>  
        <receive name="ReadAboutEnvironmentFactors"  
          createInstance="yes" partnerLink="Learner"  
          PortType="client:SP12"  
          operation="initiate"/>  
        <receive name="StudyTheExamples"  
          createInstance="yes" partnerLink="Learner"  
          portType="client:SP12"  
          operation="initiate"/>  
        <invoke name="WriteAShortReport"  
          partnerLink="Learner"  
          PortType="client:SP12Callback"  
          operation="onResult"/>  
      </sequence> //End Of "ImportantFactors"  
    </scope> //End of "A.1.2.1"  
  
    //Make Some Variable Assignations  
    <assign name="Assign_1" ... />
```

```
<scope name="A.1.2.4">
  <sequence name="ReportEvaluation">
    <receive name="EvaluateTheReport "
      createInstance="no" partnerLink="Tutor"
      portType="client:SP12" operation="initiate"/>
    <invoke name="SendCriticsAndRecommendations"
      partnerLink="Tutor"
      portType="client:SP12Callback"
      operation="onResult"/>
  </sequence> //End of "EvaluateTheReport"
</scope> //End of "A.1.2.4"
</sequence> //End of SSB1

<sequence name="SSB2">
  <scope name="A.1.2.2">
    <sequence name="TechnicalAspects">
      <receive
name="ReadAboutMatrixOfProductsCharacteristics"
partnerLink="Learner" portType="client:SP12"
operation="initiate" createInstance="yes"/>
      <receive
name="ReadAboutGoalsStrategiesAndPositionning"
partnerLink="Learner" portType="client:SP12"
operation="initiate" createInstance="yes"/>
      <receive name="DiscoverTheAnalyticConcepts"
partnerLink="Learner"
      portType="client:SP12" operation="initiate"
      createInstance="no"/>
      <receive name="DoTheExercices"
partnerLink="Learner"
portType="client:SP12" operation="initiate"
createInstance="no"/>
      <invoke name="SendExercices" partnerLink="Learner"
      portType="client:SP12Callback"
      operation="onResult"/>
    </sequence> //End of "Technical Aspects"
  </scope> //End of "A.1.2.2"

  <assign name="Assign_2" ... />

  <scope name="A.1.2.5">
    <sequence name="ExercicesCorrection">
      <receive name="CorrectTheExercices" ... />
      <invoke name="SendMarkAndTypicalSolution" .../>
    </sequence> //End of "Exercices Correction"
  </scope> //End of "A.1.2.5"
</sequence> //End of SSB2

</flow> //End of "AND_SP"

//The rest of the activities and tasks for SSB3
<sequence ... </sequence>

</sequence> //End of PSB1 (Primary Sequential Bloc)
</process> //End of the Sub_Process 1.2
```

Listing V.1 : Code du processus en BPEL

Durant l'exécution d'un PAI, le tuteur a la possibilité de suivre l'évolution de ses apprenants et de connaître leur état d'avancement. Le tuteur a aussi la possibilité de consulter la liste des apprenants exécutant le même PAI.

V.4 Conclusion

Si la conception de l'infrastructure n'a pas été facile, la mise en oeuvre l'est beaucoup plus car elle nécessite différents styles de programmation tel que la programmation web, le développement d'application et la programmation orienté agent. C'est dire toute la complexité qui caractérise le système projeté, mais son utilité est à notre sens un facteur d'encouragement pour améliorer les outils développés jusque là.

Sur ce plan, nous comptons continuer à améliorer les produits en plus d'en ajouter d'autres :

- ▲ L'outil auteur pour une plus grande ouverture vers les formats de contenu existants notamment le multimédia.
- ▲ L'outil d'élaboration des modèles de PAI doit être étendu aux PAC(s) et comporter plus de fonctionnalités comme par exemple celles relatives à la flexibilité des modèles que e-Tutoring 1.0 ne prend pas en charge actuellement.
- ▲ Un autre outil qui a été développé et dont nous n'avons pas parlé dans ce travail car il sort de son cadre est l'outil easy-décision 1.0. C'est un outil décisionnel destiné aux tuteurs et qui leur fourni des analyses et des tableaux de bord pour les aider dans leur prise de décision pour le tutorat. e-Tutoring aurait tout à gagner à être jumelé avec easy-décision durant les phase d'instanciation et d'exécution des modèles.

Conclusion générale

Ce travail nous a permis d'investir trois domaines qui sont le travail coopératif assisté par ordinateur, le domaine du e-learning dont nous avons tiré la démarche proposée et les systèmes multi-agents ainsi que leurs applications dans l'enseignement à distance.

Dans le premier domaine, nous avons d'abord étudié les notions et principes relatifs aux systèmes d'information coopératifs SIC(s). Nous nous sommes particulièrement intéressés au domaine du travail collaboratif assisté par ordinateur TCAO dont le workflow est l'un des outils. Le modèle de référence des SGWF(s) établi par la WfMC a le mérite d'essayer d'unifier les efforts fournis pour la construction de système workflows conformes à certaines caractéristiques comme l'interopérabilité par exemple. Ce modèle est un schéma spécifiant un cadre référentiel pour les SGWF(s) en définissant leurs caractéristiques communes, leurs interfaces et leurs fonctions. La mise au point d'un SGWF nécessite impérativement celle d'un moteur de Workflow, composant principal de tout SGWF, qui fournit le "run-time", c'est-à-dire, l'environnement d'exécution des instances de processus Workflow.

Nous avons du aussi voir de près les techniques de modélisation de processus métier en particulier ceux susceptible à être modélisés par les workflows. Ont été aussi relatés les aspects concernant la flexibilité des modèles de workflows et des systèmes workflows. A cet effet, l'une des premières conceptualisations du workflow se trouve être le méta-modèle proposé par la *Wfmc* qui présente les principaux éléments de l'ontologie des processus. Ce modèle a constitué pour nous un bon cadre de départ pour le développement de notre méta-modèle prenant en comptant l'adaptabilité et l'adaptativité.

D'un autre côté, cela nous a permis de constater que l'usage des technologies workflows peut être étendu à toute organisation exerçant des métiers non forcément orientés vers le monde du business, pourvu que les processus visés répondent aux règles et critères de processus workflow. Nous avons donc étendu le méta-modèle de workflow vers le domaine du e-learning qui est une forme d'enseignement à distance, on s'est particulièrement intéressé au tutorat.

Pour ce faire, nous avons abordé le domaine du e-learning et étudié ses problèmes et ses spécificités. Notre objectif a été de bien prendre en compte les besoins spécifiques en matière de tutorat et la prise en compte de la terminologie adéquate pour le domaine d'application choisi.

Enfin, le troisième domaine concerne aussi une technologie favorisant le domaine de l'enseignement qui est celle des agents. Elle nous a permis de proposer un SMA d'assistance en introduisant des acteurs artificiels dans l'aspect organisationnel du méta-modèle proposé. L'architecture du SMA que nous avons proposé de greffer au système workflow s'inspire fortement des travaux liés aux domaines des systèmes tuteurs intelligents (STI) et des environnements interactifs pour l'apprentissage humain (EIAH).

Sur le plan des outils, nous avons réalisés quelques outils dans le cadre de projets de PFE, dont :

- Un outil « AB-Learning 1.0 » pour les auteurs de contenus et même les tuteurs, est destiné à l'élaboration de contenus pédagogiques comme les leçons, les Quiz et les exercices.
- Un second outil « e-Tutoring 1.0 » a été implémenté pour l'élaboration des plans d'apprentissage de type PAI prenant en charge les activités des apprenants et des tuteurs.
- Une tentative de développement d'un outil « SMA » avec *AgentTool* a été faite mais s'est avérée ne pas répondre à nos attentes. Heureusement, l'acquisition d'un nouvel outil *AgentBuilder* au sein de l'équipe de recherche nous permettra de continuer ce travail.

Ce projet continuera donc dans le sens de l'élaboration de l'ensemble des outils nécessaires pour couvrir tous les aspects abordés en conception. D'autres perspectives restent à développer, nous citons à titre non exhaustif :

- ▲ Etudier l'intégration de cette approche de tutorat dans un environnement de learning management system (LMS). En effet, les LMS(s) présentent beaucoup d'avantages susceptibles d'être exploités.
- ▲ Etendre l'assistance à base d'agents pour des équipes d'apprenants dans le cadre de plan d'apprentissage collectif (PAC). De plus, l'évaluation de l'apprentissage collectif dans le cadre de PAC est d'une grande importance. Un premier travail dans (Boukhedouma & al, 2006) propose des éléments constituant un point de départ dans cette perspective.
- ▲ Des premiers travaux nous ont permis d'engager une réflexion sur une manière plus industrielle d'implémenter nos agents en utilisant les systèmes holoniques qui sont un autre type d'organisation multi-agent (Adam, 2002) (Mahdaoui & al2, 2004), il serait intéressant d'exploiter cette voie.
- ▲ Enfin, La prise en compte de l'aspect décisionnel en tutorat permettra d'intégrer des outils décisionnels à même d'aider le tuteur pour une meilleure qualité de suivi des apprenants. (Selmoune & al, 2006) présente une première approche dans cette direction.

Bibliographie



Références bibliographiques

Articles

- (Aimeur & al, 1999) Aimeur, E., Frasson, C., Lalond, P., "Learning with positive aspects of conflicts". ACM Journal of Science, Vol 27(3), 19 pages, 1999.
- (Alimazighi & Hamdah, 2002) Alimazighi, Z., Hamdah, M. (2002) « Modélisation de processus workflow avec le langage EXPRESS ». CIP : Conférence Internationale sur la Productique, Octobre 2002, Algeria.
- (Bauer, 2001) Bernard Bauer, "UML Class Diagrams Revisited in the context of Agent-Based Systems". Siemens AG, Corporate Technology, Informations and Communications. Otto-Hahn-Ring 6 81539, Germany, 2001.
- (Becker, 1960) Becker, H.S., "Notes of the Concept of commitment". American Journal of Sociology, July 1960.
- (Ben Ali & al, 2007) Mohamed Ben Ali, Yassine Jamoussi, Henda Ben Ghezala (2007) : « Web Services Adaptivity Approach Based on a Multi Level Vision of the Learning Process ». In Proceedings of The 2nd International Conference on e-Learning (ICEL) Columbia University, New York, USA.
- (Bond, 1990) Bond, H.S, "Distributed Decision Making in Organization". IEEE Transactions on Systems, Man and Cybernetics Conference, November 1990.
- (Boukhedouma & al, 2006) Boukhedouma S., Mahdaoui L., Alimazighi Z.: « A Workflow Based Approach to Support Individual and Collaborative Work of Learners ». In proceedings of the 5th European Conference on e-Learning (ECEL'2006), UK..
- (Caballero & al, 2005) A., F., Caballero, J., M., Gascueña, E., Lazcoretta, F., Botella, "Distant Learning By Intelligent Tutoring System, Part I: Agent-based Architecture for User-centered Adaptivity". In proceedings of the 7th ICEIS 2005, May 24th - 29th, 2005. Miami – USA.
- (Cammarata & al, 1983) Cammarata, S., McArthur, D., Steeb, R., "Strategies of Cooperation in Distributed Problem Solving", in Karlsruhe editor Proceedings of the 8th International Joint Conference on Artificial Intelligence, Germany, August 1983.
- (Cesarini & al, 2004) Cesarini, M., Monga, M., Tedesco, R. (2004) "Carrying on the e-learning process with a Workflow Management Engine", SAC'04, Nicosia – Cyprus. Copyright ACM 1-58113-812-1/03/04.
- (Chaib-draa, 1993) Chaib-draa, B., "Architecture d'un Systeme Intelligent pour les environnements multi-agents". In Actes des journées Volcants-IA 93, Clermont-Ferand, Mars 1993.
- (Dejours, 1993) Dejours, C., "Coopération et construction de l'identité en situation de travail". Revue "Future antérieur", no.2, 1993.
- (Deloach, 1999) Scott Deloach, "Multiagent Systems Engineering: A Methodology And Language for Designing Agent Systems". Afit Agent Lab, 1999.

- (DeLoach, 2001) Scott Delaoch, "Multiagent System Construction: Developing Multiagent Systems with agentTool". Afit Agent Lab, 2001.
- (Deloach, 2003) Scott Deloach, "Specifying Agent Behavior as Concurrent Tasks: Defining the Behavior of Social Agents". Laboratoire AFIT, 2003.
- (Florea, 2001) Florea Adina Magda, "Introduction to Multi-Agent Systems". In Proc. of Continuous Education Program on Intelligent Agents Technology and Knowledge Processing, Bucharest, 24 June - 4 July 2001, p.49-60.
- (Frasson & al, 1996) Fransson C., Mengelle T., Aimeur E., Gouardères G., "An Actor-based architecture for intelligent tutoring systems". The Third International Conference ITS96, Montreal, Canada, June 1996.
- (Gardner, 2003) Gardner, T. (2003) « Mapping from UML to the Business Process Execution Language for Web-Services (BPEL4WS) ». OMG MDA Implementer's Workshop, May-2003, Orlando - USA.
- (Garro & al, 2003) Garro, A., Palopoli, L., "MASel : A Multi-Agents System for e-learning and skill management". Calabria University (exp.telecomitalialab.com), Vol. 3 n° 3. Italy, 2003.
- (Gascueña & al, 2005) J., M., Gascueña, A., F., Caballero, F., Botella, E., Lazcoretta, "Distant Learning By Intelligent Tutoring System, Part II: Student/Teacher Adaptivity in an Engineering Course". In proceedings of the 7th ICEIS 2005, May 24th - 29th, 2005. Miami – USA.
- (Greenwood & al., 1995) R.M. Greenwood, I. Robertson, R.A. Snowdon, B.C. Warboys, "Active Models in Business". Proceedings of the Business IT Conference, Manchester, 1995.
- (Huguet, 2002) Mark phillipe Huguet, "Une Application d'Agent UML au Supply Chain Management". Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents, "JFIADSM", 2002.
- (Jarras & al, 2002) Imed Jarras, Brahim Chaib-draa, « Aperçu sur les systèmes multi-agents ». Montréal, Juillet 2002
- (Jennings & al, 1998) Jennings N.R., Sycara K. and Wooldrige M.A, "Roadmap for Agent Research and Development". Autonomous Agents and Multi- Agents Systems, Vol.1.1998.
- (LeMoigne, 1986) LeMoigne J.-L., "Vers un Système d'Information Organisationnel ?", Revue Française de Gestion, Nov-Dec, 1986.
- (Lewis, 1998) Lewis, R (1998) "Apprendre conjointement : une analyse, quelques expériences et un cadre de travail", Quatrième Colloque Hypermédia et Apprentissages – October 15th -17th, Poitiers - France (Invited Conference).
- (Leyking & al, 2007) Katrina Leyking, Pavlina Chikova and Peter Loos (2007) : « Competency- And Process-Driven e-Learning – A Model-Based Approach ». In Proceedings of The 2nd International Conference on e-Learning (ICEL) Columbia University, New York, USA.
- (Lin & all, 2002) Joe Lin, Charley Ho, Wassim Sadiq, Maria E.Orlowska (2002) « Using Workflow Technology to Manage Flexible e-Learning Services ». Educational Technology & Society ISSN 1436-4522, pages 116-123, 2002.

- (Mahdaoui & al1, 2004) Mahdaoui Latifa, Alimazighi Zaia : "Conception d'une plate forme de E-learning avec UML/XML et implémentation d'un outil d'aide à la construction de cours en ligne". In proceedings of SETIT'2004, March 15th – 20th, Sousse - Tunisia.
- (Mahdaoui & al2, 2004) Mahdaoui Latifa, Boukhedouma Saida & Alimazighi Zaia : "designing e-learning process with workflow: an approach based on UML and Petri nets". In proceedings of ACIT2K'2004 constantine - Algeria.
- (Mahdaoui & al3, 2004) Mahdaoui L, Alimazighi Z : « Holonic Agent System for workflow based e-learning ». In proceedings of the 3rd ECEL 2004, November 25th - 26th, 2004. Université Paris-Dauphine – France.
- (Mahdaoui & al4, 2005) Mahdaoui L., Alimazighi Z. (2005): « A cooperative Information system for e-learning : a system based on workflow and agents ». In proceedings of the 7th ICEIS 2005, May 24th - 29th, 2005. Miami – USA.
- (Mahdaoui, 2002) Mahdaoui Latifa: "L'EAO, entre théories psychologiques et nouvelles technologies ». 2^{ème} rencontre internationale sur l'université virtuelle, Alger 2002.
- (Mahdaoui & al, 2008) Mahdaoui, L. and Alimazighi, Z. (xxxx) 'Distant tutoring using flexible workflow design: an application to product management courses', *Int. J. Product Lifecycle Management*, Vol. x, No. x, pp.xx–xx. (en cours d'édition)
- (Medina-Mora & al, 1992) Medina-Mora, Raul, Winograd, Terry, Flores, Rodrigo and Flores, Fernando (1992): "The Action Workflow Approach to Workflow Management Technology". In: Mantel, Marilyn and Baecker, Ronald M. (eds.) Proceedings of the 1992 ACM conference on Computer-supported cooperative work November 01 - 04, 1992, Toronto, Ontario, Canada. pp. 281-288.
- (Odell, 2000) James Odell, H. Van Dyke Parunak, Bernhard Bauer, "Extending UML for Agents". In *Proc. of the Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligence*, Gerd Wagner, Yves Lesperance, and Eric Yu eds., Austin, TX, pp. 3-17 accepted paper, AOIS Workshop at AAAI 2000.
- (Ole & al, 2003) Ole E.M.Borch, Jan Helbe, Morten Knudsen, Ole Rokkjaer (2003) « UniFlex : A www environnement for project-based collaborative learning ». Aalborg University, Danemark. In proceedings of ITHET'03 July 7-9 2003, Morocco.
- (Ouahrani & al, 2003) Ouahrani, L., Alimzighi, Z., "Agents pour la cooperation de systèmes d'information". Conférence Maghrébine, Tunisie, 2003.
- (Peraya & al1, 2004) Peraya, D., Jaccaz, B. (2004): "Analyser, Soutenir, et Piloter l'innovation : un modèle « ASPI »". In *Actes du Colloque TICE 2004, Technologies de l'information et de la connaissance dans l'enseignement supérieur et l'industrie* (pp. 283-289). Université de technologie. Compiègne (19 au 21 octobre).
- (Peraya & al2, 2006) Peraya, D., Charlier, B. & Deschryver, N. (2006): "L'analyse des dispositifs technopédagogiques : comment rendre compte des effets de l'introduction des innovations ?" Actes du colloque CEMAFORAD 3, Colloque Euro Méditerranéen et Africain sur la Formation à Distance « Intégration des TIC à l'enseignement et à la formation. Réalités et Perspectives

- » (Conférence inaugurale) (publication électronique sur CD-Rom). Université de Sousse (12-14 novembre).
- (Pesty & al, 2001) Pesty, S., Webber, C., Balacheff, N. (2001) "Baghera : une architecture multi-agents pour l'apprentissage humain. Leibniz Laboratory, IMAG, France (www-leibniz.imag.fr)
- (Sadiq & al, 2002) Sadiq, S., Sadiq, W., Orłowska, M. (2002) «Workflow Driven e-learning beyond collaborative environments». School of Computer Science and Electrical Engineering, University of Queensland, Australia.
- (Salomon, 1993) Salomon, G (Ed), (1993) "Distributed Cognitions : Psychological and Educational Considerations", Cambridge University Press – New York.
- (Schmidt, 1994) Schmidt, K., "The organization of cooperative work : Beyond the Leviathan conception of organization of cooperative work". ACM Press. 1994.
- (Sébastien & al, 2001) Sébastien, G. and Leroux, P., "Un environnement support de projets collectives entre apprenants : SPLASH" Actes des sixièmes journées francophones Environnement Interactifs d'apprentissage avec Ordinateur (EIAO'2001), pp.49–60, France, 2001.
- (Tchounikine, 2002) Tchounikine P., 2002, "Pour une ingénierie des EIAH", Revue I3, information – interaction – intelligence 2(1) - www.revue-i3.org.
- (Van der Aalst & al, 1997) Van Der Aalst W.M.P., "Verification of workflow nets". In P. Azema and G.Balbo editors, "application and theory of Petri Nets", Springer Verlag, Berlin 1997.
- (Vantroys & al, 2002) Thomas VanTroys, Yvan Peter (2002) "A WMF-based workflow for e-learning". Université de Lille, Laboratoire Trigone, Equipe NOCE, France.
- (Verbeek & al, 2001) Verbeek H.M.W, T., Basten, Van Der Aalst W.M.P. , "Diagnosing workflow processes using WOFLAN", *the computer journal*, vol. 44 n°4, 2001.
- (Wooldridge & al, 1995) M. Wooldridge, N. R. Jennings, "Intelligent Agents: Theory and Practice". De la revue "Knowledge Engineering Review 10 (2)", 1995. Disponible sur <http://www.csc.liv.ac.uk/~mjw/pubs/ker95/ker95-html.html>.

Ouvrages

- (Alliot & al, 1994) Alliot Jean-Marc, Schiex Thomas, "Intelligence Artificielle et Informatique Théorique". CEPAD, Mars 1994.
- (Alquier & al, 2000) Alquier, A.-M, et Tignol, M.-H., "Management des connaissances et management par le projet". In Charlet, J., Zcklad, M., Kassel, G., Bourigaut, D. (EDs), ingénierie des connaissances. Evolutions récentes et nouveaux défis, Editions Eyrolles 2000.
- (Baecker, 1993) Baecker, R.M. Reading in Groupware and Computer-Supporter Cooperative Work, Morgan Kaufmann. 1993
- (Conallen, 2000) Conallen, J., "Concevoir des applications Web avec UML", Editions Eyrolles, 2000.

- (Courbon & al., 1999) Jean-Claude COURBON, Silvère TAJAN, "Groupware et intranet – Vers le partage des connaissances", 2^{ème} Edition, Editions Dunod, Paris 1999.
- (Courbon, 1993) Courbon J.-C., "Systèmes d'Information : Structuration, Modélisation et Communication". Editions InterEdition, 1993.
- (Délfard, 2000) Délfard, H., "Une explication non réductionniste de la coopération inter-firmes". Dans : Voisin, C., Plunket, A. et Betton, B., "La coopération industrielle". Editions "Economica", 2000.
- (Dufree & al, 1989) Dufree E.H., Lesser V.R. and Corkill D.D., "Cooperative Distributed Problem Solving". Vol 4, Editions Addison Wesley, 1989.
- (EncycMan, 1991) Avenir M.J., "De la Planification au Pilotage Stratégique", Encyclopédie du Management, Editions Vuibert 1991.
- (Ferber, 1995) Ferber J., "Les Systemes Multi-Agents : Vers une Intelligence Collective". Editions InterEditions, Paris 1995.
- (Hollingsworth, 2004) David Hollingsworth, "The Workflow Reference Model 10 Years On". In: Workflow Handbook, 2004, p 295-312.
- (Jacobson & al, 2000) JACOBSON, I., BOOCH, G., RUMBAUGH, J. (2000) « Le processus unifié de développement logiciel ». Editions Eyrolles, Juin 2000, ISBN 2212091427.
- (Khoshafian & al, 1998) Khoshafian, S., Buckiewics, M. (1998) «Groupware & Workflow», MASSON editions.
- (LeMoigne, 1990) Le Moigne J.-L. : "La Modélisation des Systèmes Complexes", Editions Dunod–AFCET, 1990.
- (Levan, 2000) Levan S.K., "Le projet Workflow". Paris : Hermès, 2000.
- (Lucas, 2001) Jean-Luc LUCAS, "Une architecture Internet pour le système d'information de France Télécom", Editions Eyrolles, Septembre 2001.
- (Mager, 1977) Mager, F., "*Comment définir des objectifs pédagogiques*". Paris, France: BORDAS Editions, 1977.
- (Mintzberg, 1986) Mintzberg, H., "Structure et Dynamique des Organisations". Les éditions d'organisation (Paris) et éditions agence d'ARC Inc (Montreal) 1986.
- (Muller, 1997) Pierre-Alain Muller, "Modélisation objet avec UML". Editions Eyrolles 1997.
- (Newell, 1982) Newell, A., "the Knowledge Level", Artificial Intelligence, 1982.
- (Reusser, 1993) Reusser, K., "Tutoring systems and Pedagogical Theory: representational Tools for understanding, planning and reflection in problem solving". In S. Lajoie & S. Derry (eds) Computers as cognitive tools. Hillsdale, N.J.: Lawrence Erlbaum, 1993.
- (Russell & al, 2002) S. Russell, P. Norvig, "Artificial Intelligence - A Modern Approach". Prentice Hall, 2002, Chapitre 2, p. 31-52.
- (Saadoun & al, 1996) Saadoun, M., Levan, K.,S. (1996) «Le projet Groupware». Eyrolles Editions 1996 – France.

- (Saadoune, 1996) Saadoune, M., S., K., Levan, "Le projet groupware". Paris : Editions Eyrolles 1996.
- (Searle, 1990) Searle, J.R., "Intentions in Communication : Collectives Intentions and Actions". MIT-Press, London, 1990.
- (Steele, 1990) Steele, Guy, "Common Lisp: The Language". Editions Digital Press, 1990.
- (Tardieu & al, 1994) Tardieu H., Rochfled A. et Coletti R., "La Méthodologie Merise", Les éditions d'Organisation, Paris, Juillet 1994.
- (Tuomeal & al, 1988) Tuomela, Raimo, Kaarl, and Miller., "We intentions". Philosophical Studies, 1988.
- (Van Der Aalst) Van Der Aalst, W.M.P. and Van Hee, K., "*Workflow Management: Models, Methods and Systems*". Eindhoven University of Technology. The Netherlands, 2000.
- (Weske & al, 1996) M. Weske, G. Vossen, C. Medeiros. "Scientific Workflow Management : Architecture and Applications". Fachbericht Angewandte Mathematik und Informatik.Universität Münster. Mars 1996.

Thèses, Rapports & Mémoires

- (Adam, 2000) Adam, E., "Modèle d'organisation multi-agent pour l'aide au travail coopératif dans les processus d'entreprise : application aux systèmes administratifs complexes". Thèse de l'université de Valenciennes et du Hainaut-Cambrésis, France 2000.
- (Alquier, 1993) Alquier, A.M., "MUSIC : Management et Utilisation des Systèmes d'Information Coopératifs". Habilitation à diriger des recherches, Toulouse, 1993.
- (Babin & al, 2003) Gilbert Babin., Michel Leblanc, "Les web services et leur impact sur le commerce B2B". Rapport Bourgogne : Centre Interuniversitaire de Recherche en Analyse des Organisations (CIRANO), août 2003, 38 p.
- (Baghdali, 1997) Baghdadi Y., "Contribution Méthodologique à la Conception des Systèmes d'Information Coopératifs". Thèse de l'université de Toulouse I, 11 Mars 1997.
- (Barreiro, 2006) Daniela BARREIRO CLARO, "SPOC - Un canevas pour la composition automatique de services web dédiée à la réalisation de devis". Thèse doctorat, LÉRI d'Angers : Université d'Angers, 2006, 170p.
- (Gasser, 1990) Gasser, L., "Social Conception of Knowledge and Action". Rapport technique ACT-AI-355-90, MCC, Octobre 1990.
- (Gauthier, 2004) Gauthier P., "Méthodologie de développement de systèmes multi-agents adaptatifs et conception de logiciels à fonctionnalité émergente". Université Paul Sabatier de Toulouse III. Disponible sur : <http://citeseer.ist.psu.edu/wooldridge00gaia.html>
- (Gutierrez, 2006) Bernard Gutierrez, « Aide à la Conduite de Processus Socio-Techniques ». Thèse de doctorat. Laboratoire d'accueil: LAAS – CNRS 7, avenue du Colonel Roche 31077 TOULOUSE Cedex 4 .bgutierr@laas.fr

- (Mbala, 2003) Aloys Mbala Hikolo, "Analyse, conception, spécification et développement d'un système multi-agents pour le soutien des activités en formation à distance". Thèse de doctorat de l'université de Franche-Comté (n° 977), France, 2003.
- (Morand, 2001) Frédéric Morand, « Workflow, groupware et systèmes à base de données, études des couvertures fonctionnelles, des critères de choix et des possibilités d'intégrations. Illustration à l'aide de la modélisation du Foyer André ». Mémoire de post graduation en informatique et organisation, Institut d'informatique et organisation, Université de Lausanne école des hautes études commerciales, Lausanne, Suisse, 2001.
- (Rampacek, 2006) Sylvain RAMPACEK, "Sémantique, interactions et langages de description des services web complexes". Thèse de doctorat, Laboratoire CRESTIC : l'Université de Reims Champagne-Ardenne, Novembre 2006, 198 pages.
- (Ricordel, 2001) Pierre-Michel Ricordel, "Programmation Orientée Multi-Agents". Thèse de doctorat de l'Institut National Polytechnique de Grenoble, préparée au laboratoire LEBNIZ, France, 2001.
- (Sabas, 2001) Arsène SABAS, "Systèmes Multi-Agents : Une analyse comparative des méthodologies de développement". Mémoire de maîtrise, Département de mathématiques et d'informatique, Université du Québec à Trois-Rivières, Canada, 2001.
- (Saikali, 2001) Saikali, K. (2001) "Flexibilité des workflows par l'approche objet : 2FLOW, un framework pour workflow flexibles", Thèse de doctorat. Ecole Centrale de Lyon – France.
- (Salatgé, 2006) Nicolas Salatgé, "Conception et mise en œuvre d'une plate-forme pour la sûreté de fonctionnement des Services Web". Thèse doctorat, LAAS-CNRS : Institut National Polytechnique de Toulouse, 2006, 188p.

Webographie

- (Alves & al, 2007) Alexandre Alves, Assaf Arkin., and all, "Web Services Business Process Execution Language Version 2.0". Publié le 11 April 2007. Disponible sur : <<http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>>.
- (Bellwood & al, 2002) Tom Bellwood., Luc Clément., "UDDI Spec Technical Committee Specification". Publié le 19 juillet 2002. Disponible sur : <<http://www.uddi.org/pubs/uddi-v3.00-published-20020719.htm>>.
- (Booth & al, 2007) David Booth., Canyang Kevin Liu, "Web Services Description Language (WSDL) Version 2.0 Part 0: Primer", publié le 26 Juin 2007. Disponible sur : <<http://www.w3.org/TR/wsdl20-primer>>.
- (Bpml, 2002) Business Process Management Initiative, "BPML/BPELWS". Publié le 15 août 2002. Disponible sur : < <http://www.bpmi.org/downloads/BPML-BPEL4WS.pdf> >.
- (Bucharest, 2002) "Agent Intelligent : Cours WEB interactif". Cours de l'université de Bucharest sur les SMA. Politechnica University of Bucharest, (2002).

- (Farrell & al, 2007) Joel Farrell., Holger Lausen, "Semantic Annotations for WSDL and XML Schema". Publié le 28 août 2007. Disponible sur : <<http://www.w3.org/TR/sawsdl.htm>>.
- (Gruselle, 2004) Gruselle, "La Société des agents". Source : <http://www.mi2s.u-bordeaux2.fr/~gruselle/dea/essais/TC1/Berry/TC1/main/smaV01.htm#socB>
- (Lonjon, 2004) Antoine LONJON, "Modélisation des processus métiers et standardisation". Disponible sur : <http://www.bpms.info/article.asp?ref=167>.
- (Mantell, 2005) Mantell K. (IT Architect IBM), (2005): «From UML to BPEL : Model Driven Architecture in a Web Services World». IBM Library <http://www.ibm.com/developerworks/webservices/library/ws-uml2bpel>, septembre 2005.
- (Mitra & al, 2007) Nilo Mitra., Yves Lafon, "SOAP Version 1.2 Part 0: Primer (second edition)", publié le 27 avril 2007. Disponible sur: <<http://www.w3.org/TR/soap12-part0/>>.
- (Snell, 2001) James Snell, "The Web services insider, Part 4: Introducing the Web Services Flow Language". Publié en juin 2001. Disponible sur : <<http://www.ibm.com/developerworks/library/ws-ref4/>>.
- (WfMC, 1996) Workflow Management Coalition (WfMC), "WfMC – TC – 1011", Mai 1996, <http://www.wfmc.org>.

- Index des illustrations -

Figures du Chapitre I

Figure I.1 - Représentation d'un SIO -----	2
Figure I.2 - Les applications du groupware selon (Morand, 2001) -----	13
Figure I.3 - Constituants de base d'un système de GED -----	14
Figure I.4 - Cycle de vie d'un processus métier -----	16
Figure I.5 - L'environnement d'exécution et de modélisation workflow -----	19
Figure I.6 - Classification de systèmes Workflow par domaines d'application ---	21
Figure I.7 - Modèle de référence du Workflow -----	23
Figure I.8 - Les différents composants d'une architecture de Workflow -----	28
Figure I.9 - Méta-modèle Workflow pour la définition de processus de la WfMC -----	29
Figure I.10 - Les différents cas de flexibilité du workflow d'après (Saikali, 2001) -----	31
Figure I.11 - Modèle fonctionnel des web service -----	34
Figure I.12 - Schéma d'un message SOAP -----	35
Figure I.13 - Structure générale d'un fichier WSDL -----	35
Figure I.14 - Orchestration des web services -----	37
Figure I.15 - chorégraphie des web services -----	38
Figure I.16 - Modélisation d'un processus workflow avec Oracle Workflow Builder -----	42

Figures du Chapitre II

Figure II.1 - Intersection de plusieurs domaines pour le paradigme « Agent » -----	48
Figure II.2 - Evolution vers une programmation « Agent » -----	49
Figure II.3 - Structure générale d'un agent -----	51
Figure II.4 - La relation qui existe entre les trois principales caractéristiques d'agent -----	53
Figure II.5 - Fonctionnement d'un agent réactif -----	53
Figure II.6 - Représentation de la structure interne d'agent cognitif -----	54
Figure II.7 - Fonctionnement d'un agent cognitif (a) -----	56
Figure II.8 - Fonctionnement d'un agent cognitif (b) -----	56
Figure II.9 - Modèle d'agencement des modules d'un agent permettant de caractériser son intelligence -----	57
Figure II.10 - Positionnement d'un agent dans le monde -----	59
Figure II.11 - Structures organisationnelles d'agents -----	60
Figure II.12 - Architecture d'un réseau hybride -----	62
Figure II.13 - Communication par envoi de messages -----	63
Figure II.14 - Communication par partage d'information -----	63
Figure II.15 - Mode de coopération par commande -----	65
Figure II.16 - Mode de coopération par appel d'offre -----	65
Figure II.17 - Structure générale d'un agent -----	67
Figure II.18 - Composants d'une architecture réactive -----	68
Figure II.19 - Architecture hybride en couches verticales -----	70
Figure II.20 - Architecture BDI d'un agent -----	70
Figure II.21 - L'architecture JAM selon (Mbala, 2003) -----	74
Figure II.22 - les différentes phases de MaSE -----	79
Figure II.23 - Méthodologie MaSE -----	80

Figure II.24 - Modèle de classe Agent dans AUML -----	82
Figure II.25 – Classes et instances d'agents -----	83
Figure II.26 - Communication élémentaire entre deux agents -----	83
Figure II.27 - Connecteurs AUML -----	83
Figure II.28 - Quelques techniques pour exprimer la communication avec un agent jouant plusieurs rôles ou répondant à différents cas -----	84
Figure II.29 - Diagramme de collaboration entre agents jouant plusieurs rôles ----	85
Figure II.30 : Architecture de base d'un STI -----	86

Figures du Chapitre III

Figure III.1 - Communautés du e-learning -----	93
Figure III.2 - Exemple de découpage d'un cours selon des objectifs pédagogiques	95
Figure III.3 - Phases d'un PAC -----	96
Figure III.4 - Etapes de développement d'un processus de tutorat distant -----	98
Figure III.5 - Détails de l'aspect organisationnel -----	101
Figure III.6 - Détails de l'aspect fonctionnel -----	103
Figure III.7.a - Aspect Comportemental -----	107
Figure III.7.b - Aspect Comportemental étendu aux Wf-Nets -----	108
Figure III.8 - Diagramme d'état/transition relatif aux états possibles d'une activité	109
Figure III.9.a - Diagramme d'activité décrivant un processus workflow -----	111
Figure III.9.b - Diagramme d'activité décrivant un processus workflow -----	112
Figure III.10 - Détails de l'aspect Informationnel -----	113
Figure III.11 - Association des rôles aux activités -----	116
Figure III.12 - Scénario d'interaction durant les phases d'apprentissage -----	117
Figure III.13 - Sous-processus "SP1" -----	125
Figure III.14 - Sous-processus "SP1" détaillé en tâches élémentaires -----	126
Figure III.15 - Sous-processus "SP2" -----	127
Figure III.16 - Sous-processus "SP3" -----	128
Figure III.17 - Demande d'assistance par un apprenant -----	129
Figure III.18 - Les blocs d'exécution de "SP1" -----	129
Figure III.19 - Décomposition de "SP1" par blocs d'exécution -----	130
Figure III.20 : Les blocs d'exécution de "SP2" -----	130
Figure III.21 : Décomposition de "SP2" par blocs -----	130
Figure III.22 - Les blocs d'exécution de "SP3" -----	131
Figure III.23 - Décomposition de "SP3" par blocs -----	131
Figure III.24 - Fonctionnalités du système de tutorat distant -----	135
Figure III.25 - Diagramme d'activités du scénario « Définir un processus » -----	136
Figure III.26 - Diagramme de séquence illustrant les interactions du scénario « Définir un processus » -----	137
Figure III.27 - Diagramme d'activité du cas « Instancier un processus » -----	138
Figure III.28 - Diagramme de séquence de « Instancier un processus » -----	139
Figure III.29 - Diagramme d'activité du scénario « Superviser un processus » ----	140

Figures du Chapitre IV

Figure IV.1 - Assistance virtuelle à l'aide d'un SMA -----	142
Figure IV.2 - Les acteurs artificiels du Méta-modèle de workflow pour le e-learning -----	143
Figure IV.3 - Interactions possibles dans le système d'assistance -----	153

Figure IV.4 - Fonctionnalités associées aux agents -----	155
Figure IV.5 - Connexion d'un apprenant -----	156
Figure IV.6 - Assistance à l'exécution d'un plan d'apprentissage de l'apprenant ---	157
Figure IV.7 - Assistance à l'exécution d'une Activité par l'apprenant -----	158
Figure IV.8 - Sollicitation d'assistance par l'apprenant -----	160
Figure IV.9 - Etats des agents cognitifs -----	161
Figure IV.10 - Diagramme de classe des agents (niveau conceptuel) -----	162
Figure IV.11 - Classe AUML de l'agent LAA -----	164
Figure IV.12 - Classe AUML de l'agent PTA -----	167
Figure IV.13 - Classe AUML de l'agent PA -----	169
Figure IV.14 - Classe AUML de l'agent TAA -----	170
Figure IV.15 - Classe AUML de l'agent GTA -----	174
Figure IV.16 - Classe AUML de l'agent CMA -----	176

Figures du Chapitre V

Figure V.1- Architecture Générale sans le SMA -----	178
Figure V.2 - Architecture avec le SMA -----	179
Figure V.3 - Interface de création d'un cours -----	180
Figure V.4 - Page d'accueil du tuteur -----	181
Figure V.5 - Page web principale de "e-tutoring 1.0" -----	182
Figure V.6 - Etape 1, création de la description du PAI -----	183
Figure V.7 - Etape 1, création des activités -----	183
Figure V.8 - Etape 2, définir les activités du tuteur et de l'apprenant -----	184
Figure V.9 - Etape 3, connecter les activités à l'aide des opérateurs -----	184
Figure V.10 - Etape 4, activités avec tâches spécifiées -----	185
Figure V.11 - Visualisation du modèle de processus -----	185

- Index des Tables -

Tables du Chapitre I

Table I.1 : Tableau représentant le passage de la " <i>culture verticale</i> " à la " <i>culture horizontale</i> " en entreprise -----	6
Table I.2 : Matrice « Espace \Temps » de Johannsen -----	9

Tables du Chapitre II

Table II.1 : Comparaison Agent Cognitif / Agent Réactif -----	58
---	----

Tables du Chapitre III

Table III.1 - Contexte d'insertion d'un dispositif de e-learning -----	92
Table III.2 - Comparaison entre les deux approches d'implémentation de PAC ---	104
Table III.3 - Association des objectifs pédagogiques aux parties d'un processus de e-learning -----	105
Table III.4 - Types des éléments d'un processus de e-learning -----	105
Table III.5 - Les opérateurs de contrôle de flux -----	110
Table III.6 - Répartition des activités d'apprentissage par période -----	115
Table III.7 - Répartition des activités de tutorat par période -----	116
Table III.8 - Ressource pédagogiques nécessaires par activité et par période -----	117
Table III.9 - Découpage du processus relatif au dispositif "TP-SI-01" -----	118
Table III.10 - Tâches d'apprentissage du processus "TP-SI-01" -----	119
Table III.11 - Tâches de tutorat dans le processus "TP-SI-01" -----	120
Table III.12 - Détails du processus global P.1 -----	120
Table III.13 - Détails des sous processus du processus P.1 -----	120
Table III.14 - Détails des activités du processus P.1 -----	121
Table III.15 - Détails des tâches du processus P.1 -----	121
Table III.16 - Ressources nécessaires pour l'apprentissage -----	123
Table III.17 - Ressources nécessaires pour le tutorat -----	124
Table III.18 - Ressources documentaires pour l'apprentissage et le tutorat -----	124
Table III.19 - Mode d'exécution des activités collectives -----	127
Table III.20 - Mapping de UML vers BPEL -----	131
Table III.21 - Transformation des concepts du méta-modèle vers BPEL -----	132

Tables du Chapitre IV

Table IV.1 - Description des agents du système d'assistance -----	144
Table IV.2 - Grille d'analyse fonctionnelle des propriétés d'agents -----	149
Table IV.3 - Connaissances des agents -----	150
Table IV.4 - Organization des agents -----	150
Table IV.5 - Interactions des agents -----	151

Annexes



Annexe A

Le e-learning

Le e-learning : Un enseignement à distance basé sur les TICE(s)

Depuis que le monde est monde, la nécessité d'apprendre s'est toujours faite ressentir, l'homme a en effet toujours voulu en savoir plus sur ce qui l'entourait, et ce en cherchant dans tous les sens et par tous les moyens ce qui pouvait l'intéresser. Ce n'est d'ailleurs que de cette manière qu'il a pu évoluer et développer tant de méthodes, c'est même dans cet esprit la que la méthode d'apprendre elle-même a évolué.

Quand on parle de méthodes d'apprentissage, l'accent est mis sur la transmission du savoir et comment le transmettre ? Telle est la question qui se pose. L'histoire a eu plusieurs réponses à cette question. En effet, on est passé du simple bouche à oreille, à la rédaction de livre pour y sauvegarder le savoir aux éventuels apprenants à venir, jusqu'à ce que l'on aboutisse aux écoles.

L'idée de l'école a vraiment révolutionné le monde, ses résultats ont été très vite observés et appréciés, l'école continue toujours de prouver son efficacité, néanmoins des limites commencent à apparaître. Cela est dû au développement des technologies de l'information et de la communication (TIC) pour l'éducation appelé souvent (TICE). Nous avons assisté à de profonds changements dans la communication dans le milieu scolaire. Le développement rapide des TICE(s), ainsi que le fait qu'elle constituent un réservoir quasi illimité d'informations, une sorte de gigantesque bibliothèque, et les possibilités diversifiées qu'elles offrent en matière de partage et de diffusion des informations contraignent l'école à déterminer la place qu'elle devrait leur accorder.

Donc force est de constater que ce n'est que grâce à tous ces bouleversements qu'à connu la transmission du savoir, que la technologie du e-learning a pu voir le jour, c'était en quelque sorte une conséquence inévitable de l'évolution de la technologie dans ce sens.

I.1. Quelques Définitions de l'EAD

Définition 1 :

« L'enseignement à distance n'est rien d'autre que l'enseignement classique qui utilise les nouvelles technologies telles que le multimédia et les réseaux. Il offre une grande flexibilité aux apprenants dans la mesure où leur formation ne sera pas liée au temps et à l'emplacement, de plus ils ne seront pas isolés géographiquement des grandes universités et instituts. » [KOZ91].

Définition 2 :

Définition française officielle (loi du 12 juillet 1971) [PER99] :

« La formation à distance (FAD) constitue un enseignement à distance, l'enseignement ne comportant pas dans les lieux où il est reçu la présence physique du maître chargé de le dispenser, ou ne comportant une telle présence que de manière occasionnelle ou pour certains exercices. »

I.2. l'EAD et la technologie

Comme nous l'avons constaté précédemment, l'EAD a toujours été dépendant de la technologie, sa naissance même est une conséquence du développement technologique, son développement aussi, ceci nous ouvre des perspectives futuristes libres à toutes imaginations sur les formes de l'EAD qu'on aura dans peut être 20, 50 ou même dans 100 ans. Pour l'instant on se contente du présent, dans lequel on peut retenir quatre grandes étapes dans le développement de l'EAD [INC]:

- La première génération des cours à distance offerte sur Internet avait un contenu principalement textuel.

- Une deuxième génération de cours présentait un design évolué, avec l'utilisation de métaphores.
- La troisième génération de cours permet aux apprenants de télécharger les contenus sur leurs disques et de travailler hors ligne.
- La quatrième génération fait place aux cours multimédias qui emploient des outils de pointe performants, autant audio que vidéo.

I.3. Avantage de la FAD

- ▲ Individualisation de formation en considérant que l'apprenant est unique en lui proposant un environnement et un contexte de formation (outils, contenu, mode d'apprentissage, calendrier) qui s'adapte à son niveau, ses besoins et ses préférences.
- ▲ La formation à distance est le produit de l'organisation d'activités et de ressources pédagogiques dont se sert l'apprenant de façon autonome et selon ses propres désirs, sans qu'il lui soit imposé de se soumettre aux contraintes spatio-temporelles, ni aux relations d'autorités de la formation traditionnelle.
- ▲ Le rythme d'apprentissage n'est plus un problème, chaque apprenant suit le rythme qui lui convient et non pas le rythme imposé par un groupe.
- ▲ Elimination du problème de distance : la notion d'élimination de distance à elle seule élimine beaucoup de problèmes, le fait de ne pas se déplacer permet d'économiser temps et coût. De plus, cela arrange les personnes qui ne peuvent assister physiquement aux cours à cause de plusieurs raisons légitimes ; nous avons donc un avantage social, psychologique et financier.
- ▲ La familiarisation avec les nouvelles technologies : dans l'EAD, l'utilisateur travaille souvent avec des outils issus de la technologie du Web comme les multimédias, ainsi il acquerra une maîtrise de l'outil informatique en plus des contenus de cours.
- ▲ La possibilité de la collaboration inter institutionnelle : grâce aux normes imposées aux concepteurs de plateformes d'EAD et à la compatibilité entre la plupart des architectures réseaux des plateformes, on pourrait créer des partenariats entre les différentes plateformes, ceci permet d'offrir des formations plus complètes et plus riches, et surtout, à coût moindre [TEC01].
- ▲ Augmentation du nombre de personnes qualifiées (diplômées), car dorénavant même les gens pour lesquels les distances géographiques ou d'autres contraintes étaient un rempart, peuvent étudier et avoir des diplômes.

I.4. Inconvénient de la FAD

- ▼ L'évaluation de l'apprenant ne permet pas de s'assurer que la formation a été réellement réalisée par lui-même
- ▼ En ce qui concerne l'apprentissage, l'apprenant seul peut facilement se perdre dans le flux important d'informations rencontrées, qui le conduit à cliquer dans tous les sens.
- ▼ Incertitude dans le processus d'apprentissage dans le sens où toute la responsabilité d'apprendre est attribuée à l'apprenant seul, cela nous ramène par analogie à l'autre inconvénient majeur qui est l'absence physique de tuteur, en vertu duquel cette incertitude et plusieurs autres inconvénients sont nés.

I.5. Normalisation des technologies de l'information

Si on a été amené à parler de normalisation cela veut dire qu'il existe plusieurs types de formations, donc une normalisation est de rigueur, cela en vue d'assurer une portabilité du matériel pédagogique et une interopérabilité, des systèmes qui distribuent ce matériel par des voies électroniques. Plusieurs recherches se sont dirigées dans cette voie mais la plupart se

tourment vers des stratégies d'échange du matériel pédagogique numérisé, adoptant pour ce faire une sorte de langage descriptif commun visant à faciliter l'évolution, la gestion et le libre échange des objets pédagogiques. Et puisqu'on a parlé des différents types de formations, il est alors impératif qu'on les énumère :

- La formation ouverte à distance (FOAD).
- Enseignement en ligne.
- Enseignement par visioconférence.
- Le e-learning.

Le modèle que nous allons étudier est le e-learning, mais avant de plonger dans l'étude de ce modèle, quelques éclaircissements sur le terme lui même s'imposent. En effet ce terme semble le plus adapté, car si on le voit de plus près on s'aperçoit que le terme learning révèle une pédagogie centré sur l'apprentissage, et le « e » est relatif aux technologies utilisées.

I.6. Le e-learning

Définition 1 : [KOZ91].

Le e-learning est un mode d'apprentissage basé sur l'utilisation des nouvelles technologies, permettant l'accès à des formations en ligne, interactive et parfois personnalisées, diffusés par l'intermédiaire d'Internet, d'un Intranet ou toute autre média électronique, afin de développer les compétences, tout en rendant le processus d'apprentissage indépendant de l'heure et de l'endroit.

Définition 2 : [PER96]

*Le e-learning c'est la possibilité de suivre un programme de formation à distance, en autoformation ou accompagné, de manière individuelle ou collective.
Le e-learning s'appuie sur Internet et les outils multimédias pour offrir des modules de formation court, progressif, adapté aux niveaux et besoins des apprenants.*

I.6.1. Les raisons de la naissance du e-learning

La formation devient un enjeu essentiel pour les domaines de l'éducation et de l'économie entre autres. Or, chaque jour, les technologies progressent, les métiers évoluent, l'organisation change, les méthodes de management se transforment, les besoins augmentent tant pour la formation initiale que continue. D'autre part, les budgets disponibles et surtout le temps qu'il est possible de dégager ne sont pas extensibles à l'infini. C'est la raison pour laquelle les outils construits sur l'Internet, qui offrent d'immenses atouts (autre l'économie considérable en temps et en déplacement) émergent à très grande vitesse dans les pays qui ont pris de l'avance (notamment en Amérique du Nord et en Europe). C'est pourquoi on a vu apparaître le e-learning, qui est une méthode d'apprentissage reposant sur la mise à disposition de contenus pédagogiques via un support électronique (Cd-Rom, Internet, Intranet, TV interactive, ... etc).

Le e-learning consiste à utiliser les ressources de l'informatique et de l'Internet dans l'enseignement à distance. En Allemagne par exemple, 35000 écoles sont connectées à Internet et l'ensemble des filières, modules et cours disponibles sont accessibles sur des sites. En Grande-Bretagne, 96% des lycées sont connectés à Internet.

I.6.2. Le e-learning : une méthode d'enseignement aux nombreux atouts

- *Pour les étudiants*

Le e-learning offre de nombreux avantages pour les étudiants. D'abord, il faut noter qu'il permet l'accès à de nouvelles compétences qui sont plus que jamais indispensables aux exigences de la vie moderne. Chacun peut se familiariser avec les nouvelles technologies. Ensuite, il s'agit d'un outil flexible dans le temps. Ainsi, le e-learning constitue un moyen d'aller en cours régulièrement pour ceux qui n'ont pas la possibilité de se déplacer facilement comme par exemple les handicapés. L'enseignement à distance établit des liens entre les personnes malgré l'éloignement. De plus, la situation traditionnelle des écoles en milieu rural par exemple peut se trouver renversée. Les écoles ne sont plus désenclavées.

Enfin, l'étudiant est plus autonome et devient l'acteur de sa formation. Le e-learning crée un renouveau intellectuel et transforme le mode de diffusion et de propagation des idées. L'outil informatique sert à de nouveaux usages éducatifs et il permet d'adapter la formation au tempérament, au rythme et à l'intelligence de chacun.

➤ **Les livres électroniques.**

Ils permettent à chaque élève de disposer de supports de cours plus riches, plus interactifs et au final plus légers. Cartable électronique, e-manuel, bureau virtuel de l'enseignant sortent aujourd'hui du domaine de la science fiction pour entrer dans les projets des éditeurs et des constructeurs de matériel informatique.

➤ **Les logiciels de simulation.**

Ils permettent de modéliser des phénomènes complexes, de mieux les comprendre et de procéder à des expérimentations virtuelles en modifiant certains de leurs paramètres. Un exemple, la tectonique des plaques ; aujourd'hui nous pouvons simuler le déplacement des continents à la surface du globe et les forces qui interviennent dans le processus.

Les grands établissements publics pourront bientôt mettre leurs données à la disposition du plus grand nombre de personnes, dans les meilleures conditions, tirant ainsi parti de l'augmentation des débits.

• **Intérêts du changement de méthode :**

L'éducation à distance représente un véritable marché. Le e-learning offre la possibilité d'amortir les investissements pédagogiques des cours sur un plus grand nombre d'étudiants. Cette méthode est donc plus productive. L'audience est élargie et il n'y a plus de limitation du nombre des participants. Aussi, il permet de nouvelles perspectives pour les peuples exclus des plans d'éducation à long terme. Le but de l'éducation à distance est d'universaliser l'éducation en désenclavant les populations isolées et en palliant à la pénurie d'enseignement.

I.6.3. Plate-forme de e-learning

Définition : [FAE].

Une plate-forme pour la formation à distance (FAD) est un logiciel qui assiste la conduite de la formation lorsque les protagonistes sont à distance. Ce type de logiciel regroupe les outils nécessaires aux trois principaux utilisateurs (enseignant, étudiant, coordinateur) d'un dispositif qui a pour finalité l'apprentissage à distance en fonction d'un style pédagogique.

Dans ce système, les enseignants prescrivent les contenus des apprentissages, éventuellement les parcours pédagogiques types et individualisés, incorporent les ressources pédagogiques multimédias et effectuent un suivi des activités des étudiants. On peut distinguer les enseignants qui conçoivent les contenus de ceux qui facilitent le travail des apprenants. Une plate-forme e-learning est animée par trois principales catégories d'acteurs que sont :

• **L'étudiant :**

L'étudiant, qui peut visualiser les cours en ligne ou bien les télécharger. Il pourra accéder à un système de news dans lequel il pourra consulter les messages laissés par ses enseignants par exemple.

- **L'enseignant :**

Pouvant être investi de différents rôles et fonctions, l'enseignant prend en charge par exemple la notation des tests (i.e. des devoirs). Il peut entretenir un dialogue avec les étudiants par l'intermédiaire des « news ». Il peut se charger aussi de vérifier si ses étudiants consultent régulièrement cours et exercices. Il peut également avoir le statut d'auteur, ce qui lui permet de créer les cours et tests qu'il peut ensuite soumettre aux apprenants.

- **L'administrateur :**

L'administrateur possède tous les droits de gestion du site. Ainsi, il se voit confier la création des groupes, des classes et des comptes utilisateur. De plus, il affecte les personnes aux groupes et classes.

I.6.4. Les entités et les objets gérés par une plate-forme

Pour illustrer cela nous proposons de citer les entités et objets gérés par un campus virtuel qui fait partie des différents modes d'enseignement à distance. Dans un campus virtuel on peut trouver :

1- Un apprenant :

Un apprenant est une personne qui utilise un campus virtuel dans le but de suivre une formation. Son rôle est de :

- Naviguer dans le scénario d'apprentissage.
- Explorer les ressources documentaires.
- Résoudre les problèmes.
- Réaliser des activités servant à son évolution.
- Auto évaluer ses activités.

2- Un tuteur :

Un tuteur est une personne qui exerce à distance les fonctions de conseil traditionnellement assurées par un enseignant ; les tâches d'un tuteur consistent typiquement à :

- Suivre l'évolution de la formation.
- Répondre à des questions.
- Donner un conseil, une piste à suivre.
- Déceler les difficultés d'apprentissage.

3- Les évaluateurs :

Le rôle de l'évaluateur est de définir des critères de notation de tâches didactiques effectuées par les apprenants et d'évaluer la qualité de l'apprentissage (attribution des notes). Il peut avoir un rôle dissocié de celui de tuteur lorsque son rôle n'est pas d'apporter une guidance mais d'attribuer des notes aux apprenants.

4- Le gestionnaire :

Un gestionnaire est une personne capable de configurer et de dimensionner une architecture matérielle et logicielle informatique, d'installer un système d'exploitation, d'installer des logiciels de base et d'administrer le tout.

5- Les auteurs ou concepteurs de cours :

Le rôle des auteurs (des pédagogues et des experts du domaine) et de concevoir les éléments constitutifs des cours.

6- Les réalisateurs de cours :

Cette catégorie de personnes a pour rôle de préparer et d'injecter le matériel didactique conçu par les auteurs dans le système informatique constituant la plate-forme.

7- Objets constituant le matériel pédagogique :

Les campus virtuels manipulent divers types d'objets en vue de constituer le matériel pédagogique mis à la disposition des apprenants. Ces objets pédagogiques sont donc la raison d'être des campus virtuels. Parmi les objets pédagogiques en général manipulés par ces plates-formes on trouve :

- Des cours, des unités d'apprentissage (des modules, des leçons, des blocs, des étapes, des chapitres, des sections), des unités de matière (des concepts, des thèmes, des notions), des travaux pratiques, des pré-requis, des acquis, des ressources, des travaux d'approfondissement, des synthèses, ...etc.
- Egalement des objets complémentaires à la formation (un glossaire, un index, des aides, une bibliothèque).

I.6.5. Quelques interrogations sur le e-learning

Le e-learning est-il adapté à tous les contenus et à tous les publics ? [BEL01]

A priori, oui. Le e-learning peut s'adapter à tous les contenus de formation et à des publics de niveaux très variés. Néanmoins il est davantage présent dans les domaines de la Bureautique, de l'informatique et de l'apprentissage des langues. S'il s'avère encore peu développé pour les contenus complexes (communication, management) et les contenus techniques (plomberie, nettoyage.), on note une nette progression de cette forme d'apprentissage en entreprise. Le e-learning implique, en amont, la conception d'objectifs déterminés adaptés au public visé dans l'action de formation. Il requiert en outre une capacité d'autonomie de l'apprenant pour bien progresser face à l'outil et au cheminement pédagogique.

Et le problème de l'évaluation des acquis ? [BEL01]

L'évaluation des acquis est essentielle dans les dispositifs de e-learning. Elle doit être non seulement une façon d'apprendre mais aussi une façon de prendre conscience de ce que l'on a appris. Un système d'évaluation idéal devrait suivre le parcours de l'apprenant au fur et à mesure de la progression pédagogique.

Aujourd'hui, l'évaluation des acquis reste un problème dans les dispositifs existants. Les Quiz, systèmes d'évaluation les plus courants, permettent surtout de vérifier que les contenus des formations ont été mémorisés. En revanche, on est loin de pouvoir valider le fait qu'il y ait eu compréhension. En fait, ce n'est pas tant la forme des Quiz qui est en jeu, mais leur conception et construction. De plus, le problème de la délivrance des diplômes reste entier. Seules quelques universités ont mis en place, des formations mixant présentiel et cours en ligne mais avec une évaluation traditionnelle.

Que devient la relation formateur-apprenant dans l'e-learning ? [HAR01]

Avec le e-learning, au contact physique et au temps réel se substituent **les forums, les chats, les mails et les séances de regroupements** qui doivent être conçus, préparés et annoncés à l'avance afin qu'un nombre suffisant d'apprenants y participent pour instaurer une dynamique de groupe. Les différentes méthodes et/ou temps de regroupement sont complémentaires et sont des moments forts pour repérer les besoins, identifier et résoudre les difficultés. La mise en réseau des individus constitue une communauté d'apprentissage. Effectivement la notion de

groupe est importante dans une session de formation car ce que l'on nomme « les conflits cognitifs » provoquent l'apprentissage.

Un rythme régulier permet de maintenir la motivation de l'apprenant. Celle-ci résulte également du fait qu'il est le véritable acteur de son parcours ; responsable de son rythme de formation et de ses résultats. L'adhésion à son projet a beaucoup plus d'importance que lors d'une formation classique et on peut dire que sans capacité d'autonomie, le projet a peu de chances de succès. La richesse des données trouvées sur Internet représentent des outils de formation (simulation, communication, son) qui entraînent une autre façon d'agir et d'apprendre, l'esprit critique augmente.

La relation formateur-apprenant est aussi efficace mais différente, le formateur doit dissocier son activité d'enseignement de celle d'accompagnement. Il doit beaucoup plus structurer ses séquences de formation afin de donner à l'apprenant une idée du chemin à parcourir; il doit anticiper les réactions et les stratégies d'acquisition, une phase de positionnement à l'entrée dans le parcours permet une individualisation. L'apprenant se fait une représentation imaginaire (pas de visuel), d'où l'importance de la voix pour les formations synchrones. Le contact peut aussi être plus individualisé car il se fait en dehors du groupe, le phénomène de déconcentration diminue ainsi que celui de rejet ou d'assistanat. On peut donc dire que :

« Les enseignements 'présentiel' et 'à distance' sont complémentaires ; que la qualité des données ne suffit pas, l'accompagnement est fondamental ».

Quelles modalités de construction d'une e-formation ? [KUP93]

La mise en place d'une e-formation peut s'appliquer à tous types de formations, tous types de publics et tous types de structures. Malgré cette diversité, on peut dégager une problématique commune à ces nouveaux dispositifs. La e-formation se pose soit en alternative soit en complément des formations conventionnelles. C'est un nouveau mode de formation qui articule des moyens technologiques, humains, logistiques plus complexes que dans des configurations classiques.

Dans le e-learning, c'est d'abord le processus pédagogique qui compte, même si les critères technologiques tiennent une place importante. Il s'agit de construire des scénarios, supportés par des outils, qui mettent l'apprenant au centre du projet.

Les formations ouvertes et à distance :

- Utilisent les nouvelles technologies de l'information.
- Font appel à l'autoformation individuelle ou collective.
- Permettent l'augmentation de la productivité pédagogique.
- Impliquent la collaboration d'acteurs multiples.
- Mettent une grande masse d'information à disposition des stagiaires.

I.7. L'aspect collaboratif de la FAD

Collaborer pour apprendre ou faire apprendre n'est pas une idée nouvelle, mais c'est une idée pédagogique dont la pertinence et la faisabilité s'accroissent avec l'intégration des TIC aux différents ordres d'enseignement ainsi qu'en formation continue. Il s'ensuit actuellement une grande floraison d'articles et d'ouvrages consacrés à l'apprentissage dit collaboratif. Mais tous les témoignages sérieux prouvent qu'il y a un grand chemin à faire de la théorie à la pratique.

La collaboration entre apprenants n'est pas spontanée. Elle doit être provoquée et soutenue. Le rôle des tuteurs devient alors prépondérant, certains disent prééminent sur le rôle

d'enseignant traditionnel de transmetteur de savoir. Les tuteurs sont les nouveaux médiateurs du savoir dans l'enseignement à distance. Du reste, les spécialistes distinguent au moins trois catégories de formes collaboratives : l'apprentissage collectif, l'apprentissage coopératif et l'apprentissage collaboratif.

- **Apprentissage collectif :**

Lorsqu'un individu partage ce qu'il connaît avec les autres membres de son groupe et que ceux-ci approfondissent ensemble l'objet soumis à leur investigation, il est opportun de parler d'apprentissage collectif.

- **Apprentissage coopératif :**

L'apprentissage coopératif se reconnaît à travers la répartition à l'avance des rôles et des responsabilités en vue d'accomplir une tâche d'apprentissage déterminée.

- **Apprentissage collaboratif :**

L'apprentissage collaboratif se structure autour de l'objectif poursuivi par les participants en fonction d'une suite d'actions susceptibles d'en permettre l'atteinte. Cette énumération tend à prouver que ces formes d'apprentissage, que recommandent les pédagogues et que permet la technologie numérique actuelle, ne relèvent pas de la pédagogie-fiction, mais sont réellement à portée de main si l'on veut les intégrer dans un dessein pédagogique nouveau.

Lorsqu'un étudiant est confronté à une situation pédagogique complexe (principalement la réalisation d'un projet) et réalise qu'il lui manque certaines compétences, il va appeler ses confrères et consœurs à l'aide par un message sur le forum. La philosophie qui sous-tend cet appel public est que tout le monde ne peut pas tout connaître parfaitement, surtout dans des équipes hétérogènes où les curriculums vont du technicien informatique au diplômé d'esthétique.

Dans un travail réalisé de façon **collaborative**, il n'y aura aucune répartition des tâches à réaliser entre les participants. En effet, ces derniers travailleront tous ensemble à chaque étape de l'élaboration du travail. Il sera donc impossible, une fois le travail réalisé, d'identifier le travail fourni par chaque participant. Ce genre de travail se base sur les capacités de communication et d'interaction de chaque membre participant à la constitution du groupe.

Dans le cas d'un travail **coopératif**, par contre, les tâches sont réparties à l'avance, chaque membre du groupe sait ce qu'il a à faire. Le travail final sera le montage de tout ce qui a été fait individuellement.

I.7.1. Les quatre principaux acteurs de l'interaction cognitive

La dynamique cognitive de l'apprentissage met en présence quatre acteurs que nous allons tenter ici de situer. Ce sont :

- **Le modèle de connaissance de l'expert**

Dans un environnement d'apprentissage, le modèle de connaissances est un corpus qui représente le savoir socialement reconnu ; c'est l'objet à connaître, l'objet d'étude, la matière, le contenu d'apprentissage. Il est réputé être le fait de l'expert. Il est proposé comme acteur du processus d'apprentissage collaboratif parce qu'à travers lui, c'est la société qui s'exprime ; parce qu'il influence la société et qu'il peut être influencé par elle.

- **L'apprenant**

Au départ, les connaissances de l'expert ne sont encore pour l'apprenant que des informations avec lesquelles il interagit afin d'en appréhender la structure, l'étendue, le sens et l'utilité. Dans un dialogue intérieur (qui mobilise ses capacités d'analyse, de synthèse et de penseur), il élabore ses propres représentations. Cette première phase, qui en est une exploration, se poursuit collectivement avec les membres du groupe.

- **Le groupe**

Des discussions et des débats qui ont lieu entre les apprenants au cours de la collaboration se dégagent une intelligence collective, une entité qui a son existence propre. Le groupe est une entité qui naît, croît et se développe par le jeu des rapports et ces interactions entre ses membres. Ils se donnent des règles, inventent leur propre organisation, distribuent les rôles et travaillent de manière collaborative pour atteindre un but qui est bien sûr commun.

- **Le formateur**

De nos temps, la conception qui fait du formateur un « maître » est ébranlée. En effet le formateur n'a pas comme rôle principal de transmettre les connaissances ; il n'est plus confondu avec le corpus de connaissances à acquérir. Il est plutôt vu comme un facilitateur et un animateur de l'interaction cognitive et sociale. Son rôle consiste, en éclairé avisé, à structurer la démarche de recherche et de traitement de l'information par l'apprenant et par le groupe.

Il les accompagne, les aide à donner un sens à leur travail, les encourage, les stimule et les motive. Il veille sur le groupe, recueille des informations sur son fonctionnement, pose des diagnostics, formule des hypothèses, élabore des stratégies d'interventions et les met en œuvre auprès des apprenants et du groupe. Il intervient comme médiateur entre les parties, préoccupé de la productivité cognitive du groupe mais aussi du climat social qui y règne.

I.7.2 Les outils de l'apprentissage collectif

Ces outils sont nombreux, (cela va du "simple" groupe de travail dans une classe au campus virtuel), pour notre cas, nous citerons un outil de référence depuis quelques années : **Internet**.

Internet est un outil très important et très intéressant actuellement, car il permet une collaboration entre des individus qui sont distancés géographiquement. Grâce à cette technologie nous avons vu naître de réelles communautés virtuelles, qui au-delà des frontières géographiques et culturelles réalisent des projets communs.

L'accès à une formation à distance n'est pas nouveau, car, la formation à distance est issue des cours de correspondance nés en Angleterre à l'avant dernier siècle grâce au développement du timbre poste, d'un service postal fiable et de l'amélioration de la production du papier. On présente trois étapes dans le développement de la formation à distance :

1. L'usage de l'imprimé et de la radio.
2. Dès les années 60, le multimédia, soit l'utilisation de différents médias (imprimé, radio, télévision, vidéo).
3. Dès les années 80, la micro-informatique, la télématique, puis Internet, les hypermédias et le multimédias distribués.
4. A partir des années 90, les plateformes dédiées à l'enseignement à distance comme les LMS (Learning Management System) et les LCMS (Learning Content Management System).

En conclusion, nous pouvons dire que la formation en présentiel et la formation à distance sont complémentaires, et que leur développement a été influencé par l'évolution de trois facteurs que sont :

- Celui des conceptions des modèles pédagogiques ;
- Celui des dispositifs technologiques et des médias ;
- Celui des technologies, Internet et le Web.

Annexe B

La démarche ASPI



La démarche "ASPI"¹ : Accompagner, Soutenir et Piloter un projet d'Innovation techno-pédagogique

La démarche que nous décrivons ici propose un modèle, mais aussi une vision, de ce que devrait être une démarche de pilotage de l'innovation, s'adressant aux différents acteurs impliqués dans la dynamique du changement et de l'innovation techno-pédagogique. La première version de ce modèle, appelé ASPI Model (*Analyzing, Sustaining, and Piloting Innovation*), a été développée dans le cadre du projet européen "Equel"².

Il s'agissait de réinvestir dans une vision globale les acquis des précédents projets européens dont TECFA³ avait été le partenaire, principalement "LearnNett"⁴ et "Récré@sup", mais aussi les résultats plus récents apportés d'une part, par le projet "Equel", et d'autre part par le travail de soutien mené par le groupe "IntersTICES"⁵ à seize projets de mise à distance de cours universitaires dans le cadre du programme national « Campus Virtuel Suisse ».

1. Qu'est-ce que l'innovation ?

Le concept d'innovation est vaste et défini de façon différente selon les auteurs, leurs cadres de référence et leurs centres d'intérêt. Cependant, la littérature fait ressortir quelques caractéristiques générales qui permettent d'en saisir l'essence.

Globalement, l'innovation est un changement qui, dans le but d'améliorer une situation, peut porter sur une pratique, une méthode, une façon d'enseigner certains contenus disciplinaires, une procédure, un outil ou de nouvelles clientèles, etc. Cette amélioration peut toucher un produit, un processus (en le rendant plus productif ou plus facile), elle peut également permettre d'atteindre de nouveaux objectifs ou objets qui n'auraient pu être abordés sans un changement de la situation.

Considérée comme une action, l'innovation s'identifie à un processus bien plus qu'à un produit. Elle est « centrée sur la proposition d'introduction d'une façon volontaire d'une pratique nouvelle au sein d'un établissement scolaire en vue d'une meilleure efficacité dans la réponse à un problème perçu dans l'environnement ou en vue d'une utilisation plus efficace des ressources ». L'innovation se rapproche également de la réforme par certains aspects, notamment son intentionnalité, cette sorte de volontarisme mais aussi en ce qu'elle est « une stratégie de changement planifié ».

Le rôle d'une instance externe, d'une « expertise externe » est fondamental dans le processus d'innovation. L'innovation, relève moins d'une vision sociétale que d'une conception réparatrice. Il s'agit toujours d'« une action intentionnelle développée pour faire face à une difficulté ». Cette façon de répondre à ce qui semble identifié comme un dysfonctionnement ou éventuellement à un besoin ramène l'innovation à un contexte plus local, celui de l'établissement scolaire où souvent naît l'innovation même si dans certains cas elle peut avoir été impulsée par un organisme central.

¹ Cette annexe est principalement résumée à partir des travaux des auteurs de la démarche, principalement (Peraya & al, 2004).

² E-Quality Network, 2002, "E-quality in e-learning Manifesto" présenté à la "Networked Learning 2002 conference", Sheffield, disponible sur <http://csalt.lancs.ac.uk/esrc/>.

³ TECFA : TEChnologies de Formation et d'Apprentissage. TECFA est une unité de la Faculté de Psychologie et des Sciences de l'Education, Université de Genève.

⁴ Site : <http://tecfa.unige.ch/proj/learnnett/>

⁵ IntersTICES est un groupe faisant partie de TECFA . Le projet IntersTICES (<http://tecfa.unige.ch/proj/cvs/>) s'est achevé en avril 2004 et le réseau GIRAFE destiné à lui succéder a été officiellement lancé à cette même date.

Le déplacement opéré vers le local et le contextuel est encore bien plus marqué chez un auteur comme De Ketele (Peraya & al, 2004), qui définit l'innovation comme le « surgissement d'un inédit souhaitable ou possible » tout en soulignant la relativité de cette notion d'inédit par rapport au contexte et aux acteurs. On comprend que radicalisée, cette conception amène certains à soutenir que l'innovation ne se transfère pas, mais qu'elle doit à chaque fois être réinventée. En somme, l'innovation est un processus de changement complexe, dynamique, qui s'inscrit dans la durée : il se développe entre des tensions et des enjeux liés à deux pôles souvent antagonistes que sont l'institutionnel et le local, chacun possédant ses objectifs, ses motivations et ses intérêts, sa culture, ses temporalités, et ses contraintes propres.

Dans ces conditions, piloter l'innovation c'est donc réguler un dispositif complexe. Aussi les choix épistémologiques⁶ et méthodologiques des acteurs en charge de cette régulation sont-ils essentiels.

2. Structuration générale du modèle

Le modèle que nous proposons cherche à aider tous les différents acteurs - des responsables institutionnels aux professeurs et aux étudiants - à rendre leurs projets d'innovation techno-pédagogique durables et pérennes. Or, ceci n'est pas chose aisée dans la mesure où tout système, par définition, recherche la stabilité pour exister. Dans ce contexte, l'innovation apparaît toujours comme un processus créateur de désordres, de tensions, d'inconforts pour les acteurs à qui l'on montre souvent les avantages et la valeur ajoutée de l'innovation, sans parler de ce qu'ils risquent d'y perdre, comme par exemple la maîtrise de procédures routinières, le sentiment de compétence et d'auto-efficacité, etc.

Aussi, la véritable question à laquelle doit répondre le modèle est bien celle-ci : *comment rendre l'innovation confortable pour les différents acteurs ?* Ce qui peut être formulé de façon plus opérationnelle comme suit : *quelle démarche mettre en place ? Quel rôle vais-je jouer moi-même et quelles fonctions vais-je m'attribuer en tant qu'intervenant (pilote) ? Quels rôles attribuer aux acteurs et comment les intégrer au processus ? Comment transformer les acteurs en agents de changement ? De quel type d'information les acteurs ont-ils besoin à chaque étape du processus d'innovation pour prendre des décisions allant dans le sens des objectifs du projet ? Comment exploiter les données récoltées ? Comment intégrer les résultats dans le processus ?* Le modèle de pilotage dont nous esquisserons les grandes lignes dans cette annexe repose sur l'analyse de trois axes particuliers qu'il tentera d'articuler et d'intégrer dans une modélisation cohérente :

a) **Les choix épistémologiques** dont découlent les cadres théoriques de référence et les méthodologies constituent le point central de toute démarche de pilotage de l'innovation. Ceci suppose la prise en compte d'aspects subjectifs (anthropologiques, psychosociologiques, culturels, etc.) propres aux acteurs, et porte à considérer qu'un « pilote » est avant tout un acteur réellement engagé dans le processus qu'il pilote, et enfin à reconsidérer les modes d'intervention, de soutien et de régulation du processus d'innovation.

b) **La description du dispositif** de formation dans toute sa complexité, autrement dit son « écologie d'implémentation ». Il est en effet essentiel de pouvoir donner une représentation du fonctionnement du dispositif aussi complète que possible afin de prendre en compte les différentes dimensions et composantes qui affectent le processus d'innovation. Le pilotage suppose en effet une prise d'informations régulière sur l'état du système pour pouvoir prendre les décisions nécessaires à sa régulation. Seule une conception systémique de l'analyse du dispositif peut répondre à cette exigence.

c) **La dimension temporelle** : l'innovation, est un processus qui se déploie dans le temps et la prise en compte de cette dimension est donc essentielle. Pourtant, dans le domaine de

⁶ Terme qui se confond avec la théorie de la connaissance. Utilisé aussi pour désigner la philosophie des sciences.

l'innovation technologique, la temporalité est trop souvent interprétée et analysée en référence à l'ingénierie pédagogique et aux différentes étapes que cette discipline identifie classiquement dans le développement d'un dispositif technologique ou encore pour les environnements d'apprentissage en réseau. En fait, il s'agit là d'une vision réductrice car elle ne tient pas compte de la dynamique du processus qu'il faudra intégrer au modèle.

1^{er} Axe : Le modèle descriptif du dispositif techno-pédagogique

Commençons par l'analyse descriptive du « dispositif » techno-pédagogique qui constitue d'après les auteurs de la démarche le « lieu », l'espace où se construit et se développe l'innovation. Comme tout dispositif, il peut être défini comme « une instance, un lieu social d'interaction et de coopération possédant ses intentions, son fonctionnement matériel et symbolique, enfin, ses modes d'interactions propres.

L'économie d'un dispositif - son fonctionnement - déterminée par les intentions, s'appuie sur l'organisation structurée de moyens matériels, technologiques, symboliques et relationnels qui modélisent, à partir de leurs caractéristiques propres, les comportements et les conduites sociales (affectives et relationnelles), cognitives, communicatives des sujets ». Cette définition bien qu'elle rende compte de la plupart des environnements virtuels de formation actuels a été l'objet de critique parce qu'elle ne laissait aucune place à la démarche de pilotage. La place qu'occupe le modèle descriptif du dispositif dans le modèle, répond à cette critique tout en donnant à la démarche de monitoring un outil d'analyse du dispositif innovant lui-même.

La régulation demande en effet une analyse, dans une perspective systémique, des différentes composantes du dispositif : une modélisation de ce dernier permet donc d'identifier, aux différentes composantes, aux différentes étapes du processus, les différents aspects qui peuvent fournir les données indispensables aux prises de décision.

Quatre dimensions générales ou quatre familles de variables, définissant le dispositif innovant ont été identifiées: a) les variables structurelles ; b) les variables actanciennes relatives aux acteurs du dispositif ; c) les variables individuelles ; d) les variables des domaines.

a) Les variables structurelles

L'analyse systémique a classiquement proposé une hiérarchie de niveaux d'analyse articulés, de granularité croissante, afin de rendre compte de la complexité des phénomènes observés. La théorie de l'innovation d'inspiration systémique s'est largement référée à ces trois niveaux d'analyse : micro, méso et macro. L'affectation de ces niveaux peut cependant différer selon les auteurs.

Par exemple, certains auteurs décrivent le micro-niveau comme celui qui renvoie aux aspects motivationnels cognitifs et affectifs des acteurs. Le méso-niveau serait celui du micro social, autrement dit des relations entre les acteurs et les normes et valeurs des collectifs impliqués. Le macro-niveau relève du monde social dans lequel s'insère l'innovation.

Pour d'autres par contre, le micro niveau est celui du dispositif de formation, le méso, celui de l'institution de formation dans lequel s'insère le dispositif innovant, enfin le macro-niveau concerne le niveau sociétal au sens large.

Les auteurs d'ASPI pensent qu'il est préférable de considérer qu'il s'agit de trois niveaux articulés et de granularité différente qui peuvent être projetés sur la réalité observée à partir de points de référence différents selon les questions posées, la situation analysée, etc. Il s'agirait donc plutôt d'un principe méthodologique réglant la description et l'observation des phénomènes étudiés. Cela dit, il est possible, d'identifier, par exemple, les niveaux suivants : ministère, agences gouvernementales, différentes organisations nationales, institutions, facultés, départements, unités d'enseignement, cours, classe, individu. Le choix des niveaux observés et leur importance relative pourra évoluer en fonction du projet lui-même, des objectifs qui sont les siens, de sa dynamique, de la phase à laquelle il se trouve, etc.

b) Les variables actanciennes

Les variables actanciennes concernent les acteurs du dispositif qui occupent une place centrale. La démarche du groupe "InterTICES" (*op.cit.*) constitue un bon exemple de cet intérêt central pour les acteurs et pour leur culture qui se réfère ici : « aux représentations, aux habiletés (et ressources à disposition), aux attitudes et aux pratiques des acteurs. Ce sont là des éléments qui influencent fortement la mise en place de l'innovation et qui, par ailleurs, sont aussi transformés par la mise en place de l'innovation. En effet, l'innovation changera les représentations et pratiques des acteurs, et dans ce sens, l'innovation devient un terrain de formation continue dont l'objet devient la culture des acteurs».

Cependant, il est important de faire une distinction entre les variables actanciennes qui concernent plus directement leur fonction, leurs tâches et leur rôle d'une part, et les variables individuelles qui concernent plus directement les individus concrètement engagés dans le projet. En effet, la même personne peut, dans de nombreux cas, assumer plusieurs fonctions et plusieurs rôles.

Il faut encore préciser la différence à faire entre la fonction et les rôles. Le premier terme constituerait la facette strictement professionnelle de l'acteur (ce qu'il doit faire, ses tâches, son cahier des charges), le second désignerait des aspects psychosociaux et comportementaux qui relèvent de la place et de la posture de l'acteur dans le dispositif. Le rôle « s'interprète » selon les variables individuelles des acteurs.

Par ailleurs, les rôles et les fonctions peuvent être différents selon le contexte. Mais ce qui semble important est que chaque acteur ait une perception claire de son rôle, et de ses tâches. Autrement dit que ceux-ci fassent l'objet d'un travail d'explicitation, de clarification et de négociation entre les acteurs concernés.

Il est bien sûr difficile de donner une liste exhaustive des différents rôles indispensables à la réalisation d'un projet d'innovation, leur nombre et leur diversité dépendant en effet du contexte de chacun des projets : ressources disponibles, taille du projet, culture de ses acteurs, etc. Aussi les classifications proposées par les auteurs sont-elles très différentes et parfois peu comparables.

A titre d'exemple, on rappellera deux familles principales de fonctions dans un dispositif de formation à distance que sont : les fonctions pédagogiques et non-pédagogiques, i.e., administratives, technologiques et logistiques. Paquette, quant à lui, identifie cinq acteurs: l'apprenant, le présentateur, le concepteur, le formateur et le gestionnaire, ayant chacun un rôle générique et des rôles secondaires.

Dans le projet "LearnNett" qui rassemble en une communauté de pratique plusieurs universités européennes (belges, suisses et française), les différents acteurs et rôles ont été strictement définis dans le Guide pédagogique LearnNett de 2004 en fonction du scénario pédagogique. Parmi les rôles importants (pour LearnNett), celui d' « animateur local » qui accompagne les étudiants de son université. Il assure une formation pédagogique préalable à l'expérience (représentations, concepts liés à la collaboration, à l'usage des TIC en éducation...).

Enfin, il organise à la fin de l'expérience une réunion d'évaluation au niveau local avec le professeur et les étudiants de son université. Le tuteur est un acteur central dans le projet : c'est la personne qui interagit directement avec les étudiants et les soutient dans leur apprentissage. Son rôle est pédagogique, il est davantage centré sur le processus que sur la production de contenu. Selon les groupes, ses interventions peuvent varier. Son rôle évolue également en cours de projet d'une intervention plus proactive au début vers une intervention plus réactive.

Voici, à titre d'exemple quelles sont les différentes fonctions **tutorales** identifiées dans ce projet: permettre le contact et la prise de connaissance entre les partenaires; *organiser le groupe; apporter un soutien technologique; socio-affectif et motivationnel; être un référent théorique; un appui méthodologique; organiser le suivi des apprenants; et organiser le travail de communication.* Chacune de ces fonctions peut alors être développée et opérationnalisée.

c) Les variables individuelles

Pour chaque acteur, quel que soit le niveau où il se situe dans le dispositif, quels que soient ses fonctions et son rôle, on doit prendre en considération différents aspects, différentes variables individuelles qui le constituent et le caractérisent. Chacun possède en effet :

- **des caractéristiques personnelles** qui sont classiquement celles prises en compte par l'identification sociologique du sujet : sexe, âge, niveau d'études et/ou de qualification.
- **des représentations, des visions** : chaque acteur possède des valeurs, des conceptions, des représentations, des pensées et des croyances - individuelles ou socialement partagées par le groupe ou la collectivité auxquels il appartient - qui l'aident à comprendre son environnement et à agir sur celui-ci.
- **des compétences et des ressources** (y compris matérielles, économiques) dont dispose chacun pour mener à bien les tâches et le projet. On peut bien sûr distinguer certaines sous catégories suivant en cela la littérature : les connaissances ou les savoirs qui sont reconnus, stabilisés et partagés, les savoir-faire intellectuels, les savoir-faire manuels ou des habiletés. Parmi ces compétences la dimension réflexive et la compétence métacognitive occupent une place importante.
- **des attitudes, des envies, des attentes, des motivations, des besoins, des craintes** : celles-ci dépendent souvent du parcours de formation de la personne, de sa vie personnelle, de ses projets professionnels. L'existence ou non d'un projet personnel est donc un facteur important.
- **des pratiques** : chacun se caractérise encore par ses pratiques réelles, par la façon dont il accomplit ses tâches dans le cadre de sa fonction, développe ses scénarios pédagogiques, et intervient auprès des apprenants, etc.
- Et enfin, l'**expérience professionnelle** de chacun.

d) Les variables des domaines

Les variables individuelles déterminent vraisemblablement l'interprétation du rôle de chacun des acteurs. Elles s'incarnent aussi dans les différents domaines qui constituent l'univers professionnel et social des acteurs. C'est en effet à propos de chacun de ces domaines, que l'acteur possède et incarne chacune de ses variables individuelles : représentations, compétences et ressources, attitudes, pratiques réelles et expérience professionnelle. *Quels sont alors ces principaux domaines ?*

- **la pédagogie** : les positions épistémologiques, les théories et les modèles d'apprentissage, les courants pédagogiques, les approches et les méthodologies, les objectifs d'apprentissage, etc. Sur chacun ou sur certains de ces aspects les acteurs ont tous leur idée, leur expérience.
- **les disciplines** : l'innovation s'inscrit bien sûr dans le cadre d'une discipline qui possède ses contenus et leur organisation propre (progression, modularisation), sa didactique, etc.
- **les technologies** : les technologies existantes et disponibles, leurs choix, leurs rôles dans l'environnement de travail, etc.
- **la médiatisation et la médiation** : il s'agit des aspects caractéristiques de la médiatisation des contenus et des systèmes (qui relèvent de l'ingénierie de la formation) d'une part, de la médiation et de l'utilisation de caractéristiques des dispositifs technologiques comme support aux activités et aux apprentissages (qui relèvent de l'analyse des outils cognitifs, des registres sémio-cognitifs, des formats de présentation, etc.) d'autre part.
- **l'organisationnel** : les formes et les modèles d'organisation de la formation comme l'importance et la place de la distance, etc.
- **l'économique** : les contraintes et les possibilités économiques, les besoins du marché, etc.

- **le politique** : conception et organisation politiques et modèle de la société civile, etc. Le politique secrète ses propres valeurs et son idéologie.

2^{ème} Axe : La dimension temporelle

Ce deuxième axe est celui du développement et du déploiement de l'innovation depuis son émergence jusqu'à sa pérennisation. L'ingénierie pédagogique et l'"instructional design" ont depuis longtemps identifiés, du point de vue méthodologique, une série d'étapes caractéristiques du processus de conception et de mise en œuvre d'un dispositif technologique.

Les analyses de cas ont confirmé l'existence d'étapes bien identifiées : analyse, conception, développement, mise en place, évaluation, intégration dans la pratique quotidienne, maintien du processus et diffusion.

On pourrait d'ailleurs, comme le font certains auteurs, simplifier le processus en trois étapes majeures en prenant comme axe d'analyse le point de vue de l'innovation : décision d'adoption, implémentation et routinisation. L'analyse du dispositif innovant dépend en effet de chacune de ces étapes puisque l'importance relative de certaines composantes, de certaines variables, évolue en fonction de chacune de ces étapes.

Par exemple, « C'est dès la phase d'analyse des contraintes, des objectifs et des conditions de réalisation du projet, qu'il faut aller valider sa vision et sa compréhension du projet avec les personnes concernées : des apprenants, des enseignants, avec des experts du domaine qui vont voir si on a oublié certaines parties. ».

Au-delà de ces moments privilégiés, la littérature et l'expérience montrent que de nombreux événements critiques constituent des indicateurs précieux dans la dynamique du projet puisque l'on sait qu'ils contribuent à faire avancer, retarder ou même échouer une tâche. Un pilotage qui suivrait strictement les modèles types de l'ingénierie pédagogique risque donc de laisser échapper des moments clés de la dynamique du projet.

Le pilotage prendra donc en compte ces deux aspects de la dimension temporelle : d'une part, la chronologie stricte des étapes de développement d'un projet et d'autre part les moments qui paraissent critiques au regard de sa dynamique de changement propre.

3^{ème} Axe : La démarche de pilotage

L'utilisation de la notion de « pilotage » pour désigner la gestion de l'innovation est aujourd'hui bien ancrée dans la littérature, mais il existe plusieurs façons de piloter. On peut distinguer à travers la littérature, deux visions principales de l'innovation, ainsi que deux approches du pilotage, du soutien et de l'accompagnement. Chacune d'elle relève d'ailleurs de cadres épistémologiques différents. La différence essentielle porte sur le rapport aux acteurs - y compris les chercheurs et les responsables de l'innovation - et leur place dans le processus de pilotage.

La première approche considère le pilotage comme une prise de données régulière sur le système en évolution par des chercheurs/analystes qui demeurent non impliqués par rapport au projet lui-même et à son développement. Les méthodes sont de l'ordre de la recherche conventionnelle et les résultats peuvent être réintroduits dans le processus afin d'en infléchir le cours, d'en corriger d'éventuels dysfonctionnements.

La seconde s'apparente aux démarches de type recherche-action-formation et font des acteurs et des chercheurs des partenaires d'un processus partagé. Selon l'expression de Fullan, chaque acteur devient alors un agent de changement. Il s'agit de mener conjointement la formation professionnelle des enseignants impliqués dans un projet innovant, le développement de ce même projet, et la recherche sur l'expérience en cours. Il s'agit donc d'une démarche participative qui tend à effacer la frontière entre milieu professionnel, et l'apprentissage de nouvelles compétences professionnelles.

Proposition d'une Infrastructure Statique et Dynamique pour le Support du Travail Collaboratif : « Application au tutorat dans le e-learning »

Les stratégies, sur la base d'un projet de formation mené par Charlier et Charlier (Peraya & al, 2004) concernent : a) l'explicitation des apprentissages, b) l'explicitation des pratiques, c) le rapport entre savoirs et techniques, d) la liaison entre la pratique professionnelle et l'objet de formation, e) une alternance entre théorie et pratique...

Le modèle ASPI et la vision qui l'inspire ont opté pour une démarche qui relève directement de la seconde orientation. Il faut donc définir les objectifs d'une telle démarche, les modes d'actions qui semblent aujourd'hui validés tant par nos études de cas que par ceux présentés dans la littérature, et enfin les outils qui permettent de les mettre en œuvre.

La démarche intègre, au-delà des principes et de la méthodologie de toute démarche action-recherche-formation, les éléments apportés par la critique de la définition du dispositif. La démarche porte donc sur l'élaboration « d'un objet commun à négocier entre et avec les sujets (social). Elle oblige à interroger ce qui se joue, au jour le jour (dimension dynamique) dans l'actualisation des pratiques nouvelles (dimensions innovantes) ».

Il s'agit donc d'analyser la dynamique du développement de l'innovation selon une régulation intégrant les acteurs et pour ce faire, le modèle descriptif du dispositif innovant permet parfaitement les prises d'informations nécessaires aux différentes étapes du processus. De plus, chacune des grandes familles de variables - et même chacune des variables - peut constituer un point d'entrée afin de faire émerger des représentations du dispositif sous différents angles de saillance. D'autre part, une telle démarche a pour objectif d'élucider avec les acteurs le sens des « mieux visés par l'innovation ». Le schéma ci-dessous rend compte de ces aspects :

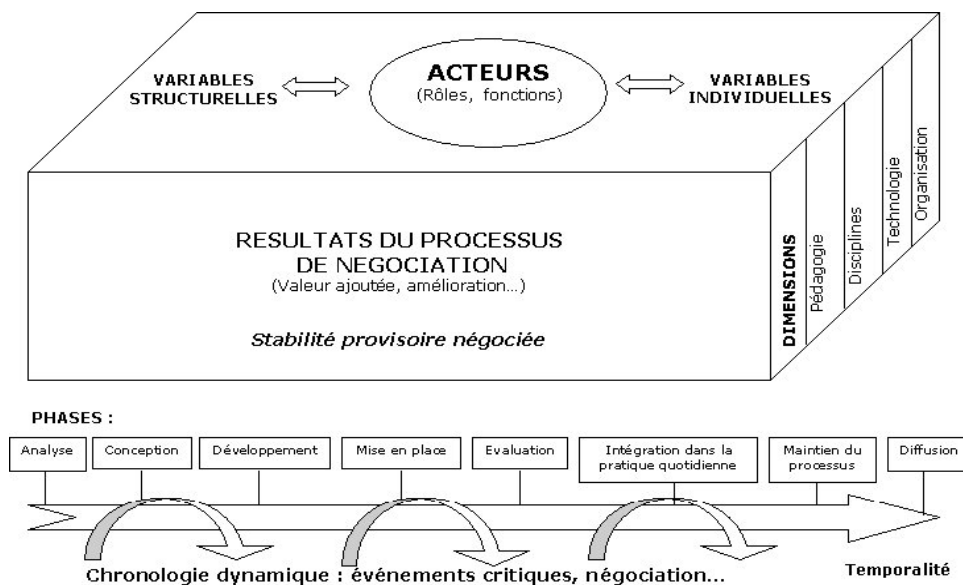


Schéma : Analyser, Soutenir et Piloter l'Innovation (ASPI)

L'entrée dans le modèle peut se faire par les acteurs, les dimensions, ou l'axe temporel. Parmi les modes d'action à privilégier sur la base d'études de cas réalisées dans le cadre de mémoires universitaires : a) un pilotage proactif plutôt que réactif ; b) un pilotage flexible et évolutif intégrant une dimension adhocratique ; c) un soutien à l'action et une garantie des ressources nécessaires à celles-ci (le cas d'InterSTICES est de ce point de vue exemplaire) ; d) une facilitation de l'articulation entre projets individuels et organisationnels; e) une réorganisation du travail en termes de temps et de lieux de concertation. Pour ce faire, de nombreux outils existent qui ont pu être validés par la pratique.

III. Conclusion

Le modèle ASPI considère donc que la démarche de pilotage fait partie intégrante du dispositif innovant et en constitue le centre moteur dans la mesure où elle rend possible les conditions d'émergence de l'innovation. Elle en assure la cohésion choisissant, à chaque moment important, parmi les différentes variables les données qui sont nécessaires au diagnostic, à la régulation du dispositif autant qu'à l'explicitation du sens de l'innovation pour les acteurs.

Ceux-ci, quel que soit le niveau où ils interviennent, doivent donc à certains moments ré-analyser, et éventuellement modifier, leur posture, leur rôle au sein du dispositif : acteurs du dispositif, œuvrant à l'intérieur de celui-ci pour son développement, ils sont aussi agents de changement ce qui leur impose une attitude réflexive les mettant alors en position d'extériorité par rapport au dispositif. Ils construisent et développent par là-même une culture technopédagogique favorable à la mise en place de projets innovants.

Annexe C

Le Dispositif « TP-SI-01 »

Spécification du dispositif "TP-SI-01" : Travaux Pratiques du Module Système d'Information (Version 01)

1. Résumé

"TP-SI-01" est un cours-TP fournis aux étudiants de 2^{ème} année de la filière informatique de la faculté d'Electronique et d'Informatique de l'USTHB (Université des Sciences et de la Technologie Houari Boumédiène). Mis à la disposition des apprenants par le biais de son campus numérique, la faculté offre un ensemble de cours à distance au service de ses apprenants afin de renforcer certains aspects de leur formation. Ce cours s'inscrit dans le domaine de l'ingénierie en informatique, en particulier l'introduction aux systèmes d'informations.

L'objectif principal de "TP-SI-01" est de prendre en charge la partie pratique (Réalisation de Travaux Pratiques) du module « Introduction aux Systèmes d'Information » (I.SY.I) dispensé aux élèves ingénieurs de 2^{ème} année de la formation Ingénieur en Informatique. L'apprenant doit d'abord découvrir le langage de programmation qu'il devra utiliser dans son TP puis à réaliser son TP individuellement, en binôme ou en trinôme, en appliquant les connaissances acquises dans les cours et TD présentiels.

L'évaluation finale des TP se fera en salle de TP par des séances de démonstration en présence des tuteurs de TP-SI-01. La note de TP sera définie à partir des résultats de la démonstration de TP plus celle des notes obtenues à distance. La note de TP comptera pour 1/3 de la moyenne du module « I.SY.I ».

2. Identification du dispositif

<i>Titre du projet</i>	TP-SI-01
<i>Adresse Internet</i>	Non encore disponible (en cours de conception)
<i>Auteur(s) – Nom & Coordonnées</i>	L.MAHDAOUI (mahdaoui@lsi-usthb.dz)
<i>Date de dernière mise à jour</i>	En cours de conception
<i>Public et/ou niveau d'enseignement visé</i>	Etudiants de 2 ^{ème} année Informatique de la formation Ingénieur, 2 ^{ème} année LMD (License ACAD / ISIL) et toute personne intéressée par le sujet.
<i>Discipline</i>	Informatique de gestion, Systèmes d'information.
<i>Type de situation d'apprentissage</i>	Apprentissage Individuel Auto-Apprentissage Apprentissage Collaboratif
<i>Brève description du scénario – tâches et résultats attendus</i>	<p>Le dispositif doit être inséré dans le cadre du module « Introduction aux Systèmes d'Information » (I.SY.I) prévoyant une série de cours, TD et TP destinés en particulier aux élèves ingénieurs de la 2^{ème} année informatique. TP-SI-01 est un cours à distance visant à couvrir la partie TP du module.</p> <p>Le programme de TPSI 01 est prévu en trois périodes de temps devant se dérouler en parallèle avec les séances de cours/TD présentielles. L'apprenant aura à réaliser un ensemble</p>

	<p>d'activités dans chacune des périodes, en plus d'une séance présentielle hebdomadaire de TP de 1h30 mn prévue pour chaque groupe de TP (auparavant, il n'y avait que cette séance et cela s'est avéré très insuffisant !).</p> <p>Une fois son login et mot de passe obtenu par l'enseignant du module, l'apprenant peut se connecter au site du cours pour commencer ses travaux.</p> <p>Les tuteurs du TP évaluent les rapports de TP individuels ou collectifs (par binôme ou trinôme) remis par les apprenants en fin de troisième période et fournissent les feedbacks selon les points prévus à cela dans le planning de l'activité. Les tuteurs procéderont ensuite à l'évaluation des produits logiciels réalisés par les apprenants selon un calendrier présentiel de démonstration prévu en fin d'année.</p> <p>Les activités sont décrites pour chaque période selon un calendrier présence distance et chaque activité peut correspondre à plusieurs tâches à exécuter. En fin d'apprentissage l'apprenant est censé acquérir les concepts du domaine abordé (conceptuel) ainsi que la maîtrise des outils et procédures permettant de mettre en pratique ses connaissances, c'est à dire, apprendre à réaliser une application logicielle (Base de données + Traitements) sur la base d'une étude conceptuelle menée sur un domaine d'étude donné.</p> <p>Dans TP-SI-01, l'apprenant devra apprendre à appliquer les connaissances acquises dans le cours I.SY.I notamment la conception avec la méthode MERISE et fera sa partie pratique avec le langage CLIPPER 5.2 ou 5.3.</p>
<p>Conditions d'utilisation</p>	<p>Micro-ordinateur + Connexion à Internet + Droit d'accès au système + outils logiciels spécifiés dans le cours tel que le langage CLIPPER.</p>

3. Scénario pédagogique

Ce dispositif se base sur deux formes d'enseignement : l'enseignement présentiel où des séances seront administrées aux apprenants par les assistants de TP (qui sont aussi des tuteurs à distance) et l'enseignement à distance. Le dispositif favorisera aussi bien l'apprentissage individuel que collaboratif.

Le cours est organisé en 3 périodes réparties sur toute l'année et traitant chacune d'un sujet se rapportant au TP du cours I.SY.I comme montré dans le tableau suivant :

Période (Du -- Au --)	Sujet abordé	Objectifs
<p>Du 15/10/2006 Au 15/12/2006</p>	<ul style="list-style-type: none"> • Expérimentation d'une version d'évaluation de logiciels existants sur le marché et développés localement (par des ingénieurs en informatique). • Découvrir le langage de programmation NANTUCKET CLIPPER. 	<p>✓ L'apprenant devra se faire une idée que tous les produits informatiques existants sur le marché ont été développés par des informaticiens qui ont été étudiants comme lui (cela met les apprenants en confiance et crée en eux une situation de défi).</p> <p>✓ Les logiciels développés ont fait</p>

Proposition d'une Infrastructure Statique et Dynamique pour le Support du Travail Collaboratif : « Application au tutorat dans le e-learning »

		l'objet d'étude et de travail de programmation ardu avant d'être dans leur version commerciale. ✓ L'apprenant apprendra un des langages utilisés dans le développement d'applications orientés gestion.
Du 16/12/2006 Au 31/01/2007	•Expérimentation du langage de programmation et réalisation de quelques manipulations.	✓ Acquérir les éléments essentiels du langage de programmation.
Du 01/02/2007 Au 25/05/2007	•Réalisation du TP du module I.SY.I	✓ Savoir mettre en pratique toutes les connaissances acquises dans le module I.SY.I.

En plus d'assister aux séances présentiels (1 fois par semaine/groupe), l'apprenant doit effectuer une série de tâches d'apprentissage et de travaux individuels et/ou collaboratifs relatifs aux sujets traités et selon un planning détaillé pour chaque période, chose qui guide vraiment l'apprenant à chaque étape de son apprentissage.

La remise des travaux aux tuteurs se fait via les wikis individuels ou collectifs selon le type de la tâche à faire. Les tuteurs procèdent alors à l'évaluation des travaux et retournent les FB¹(s) aux apprenants. Certains FB(s) peuvent être discutés pendant les séances présentiels.

De façon générale, les séances présentiels de TP sont prévues en salle d'ordinateur afin de répondre aux questions des apprenants, faire de la pratique à travers quelques exercices et orienter les apprenants dans la phase de réalisation de leur TP. Notons toutefois que l'apprenant pourrait se suffire du dispositif TP-SI-01 sans être obligé d'assister à toutes les séances de TP.

Activités de la 1^{ère} Période :

Les délais sont donnés de manière approximative, l'essentiel est que l'apprenant arrive à effectuer tous ses travaux dans les limites temporelles de la période. Chaque activité peut être décomposée en tâches si nécessaire :

Activités	Tâches
<u>Activité 1</u>	<ul style="list-style-type: none"> ✻ Télécharger le logiciel (version d'évaluation) après l'avoir sélectionné et l'installer sur sa machine en suivant les instructions décrites dans la page correspondante du site. ✻ Explorer les différents menus du logiciel. ✻ Utiliser le forum pour discuter sur le logiciel évalué ✻ Rédiger un rapport individuel en utilisant le Wiki individuel prévu à cet effet.
<u>Activité 2</u>	<ul style="list-style-type: none"> ✻ Récupérer le support de cours (disponible au format pdf) élaboré par le chargé de cours sur le langage CA-NANTUCKET CLIPPER. ✻ Télécharger et installer le langage CLIPPER selon les instructions fournies sur le support. ✻ Effectuer les exercices fournis sur le support pour une première prise en

¹ FeedBack ou retour d'évaluation.

	main du langage.
<u>Activité 3</u>	<ul style="list-style-type: none"> ✿ Appliquer les instructions pour exécuter le « Norton Guide » (NG) fourni avec le CLIPPER. ✿ Expliquer ce que fait l'ensemble des instructions utilisées dans le code source fourni en annexe du support à l'aide du NG. ✿ Utiliser le Wiki pour rendre le devoir (respecter la grille fournie dans les ressources)

Les feedbacks des tuteurs seront envoyés en fin de chaque activité, afin de permettre à l'apprenant de se situer dans son apprentissage.

Activités de la 2^{ème} Période :

Cette période porte sur une expérimentation plus approfondie du langage CLIPPER ainsi que la réalisation de quelques manipulations. Durant cette phase, l'enseignant responsable du module I.SY.I aura fait circuler une liste entre les apprenants pour former binômes ou trinômes de TP. Cette liste permettra au responsable du cours et son staff de tuteurs de préparer des sujets en adéquation avec le nombre d'étudiants par équipe de TP (binôme ou trinôme).

A partir de ce point, les apprenants utiliseront des Wikis collectifs car ils travailleront en équipe de 2 à 3 apprenants. Dans certaines conditions exceptionnelles, le travail en monôme peut éventuellement être accepté car le dispositif vise à encourager plus le travail collaboratif.

Activités	Tâches
<u>Activité 1</u>	<ul style="list-style-type: none"> ✿ Taper, compiler et exécuter le code source de la Manip #01 donné dans la ressource puis répondre à travers le WIKI aux questions du document « Questions » fourni dans les ressources. ✿ Trouver l'erreur de compilation se trouvant dans la manip #02 et utiliser le NG pour l'expliquer. Répondre dans le WIKI. ✿ Dans la manip #03, se trouve une erreur d'édition de liens, la trouver et répondre dans le WIKI. ✿ La manip #04 ne comporte ni erreur de compilation, ni d'édition de lien. Portant un problème subsiste lors de l'exécution. Trouver pourquoi et répondre dans le WIKI.
<u>Activité 2</u>	<ul style="list-style-type: none"> ✿ Ecrire les 05 programmes relatifs aux 05 exercices à résoudre fournis dans la ressource correspondante. ✿ Envoyer vos codes sources et vos exécutables en attachement dans le WIKI. Chaque attachement doit être documenté par des commentaires explicatifs.

Activités de la 3^{ème} Période :

C'est durant cette phase que les apprenants réaliseront le TP selon le sujet qui leur aura été attribué. L'équipe du module composé de l'enseignant (assurant le cours et les TD) ainsi que les assistants de TP (qui sont aussi les tuteurs distants) élaboreront une variété de sujets de TP touchants à des domaines divers tels que la comptabilité, la médecine, le transport, ... etc.

La consistance du sujet variera selon que ce dernier est destiné à un monôme ou à un binôme ou trinôme pour une meilleure équité dans la quantité de travail. De plus, pour réduire les possibilités de plagiat, il sera défini plusieurs sujets par catégorie. L'attribution des sujets de fera par le staff pédagogique et de manière aléatoire.

Activités	Tâches
<u>Activité 1</u>	<ul style="list-style-type: none"> ☼ Lire le sujet de TP et discuter dans le groupe du procédé à suivre pour sa réalisation. ☼ Convenir entre vous d'un partage du travail à effectuer. ☼ Utiliser le WIKI pour enregistrer vos premières impressions.
<u>Activité 2</u>	<ul style="list-style-type: none"> ☼ Rédiger un rapport de synthèse de votre TP en respectant le format du prototype fournis dans cette activité. ☼ Envoyer le rapport en attachement par le biais du WIKI.
<u>Activité 3</u>	<ul style="list-style-type: none"> ☼ Préparer des disquettes ou CD contenant votre code source et votre logiciel exécutable et les ramener avec vous le jour de la démonstration. Un planning des moments et lieux de démonstrations vous seront affichés à la fin de la troisième période.

4. Description du projet

4.1. Contexte d'insertion du dispositif

"TP-SI-01" est un cours de TP proposé par le département d'informatique de la Faculté d'Electronique et d'Informatique (FEI) qui se trouve dans l'USTHB. Cette dernière a mis à la disposition de sa communauté enseignante et apprenante un campus numérique permettant la mise en place de dispositifs d'enseignement à distance prévu pour le moment pour améliorer la qualité de l'enseignement mais aussi plus tard pour arriver à mettre en œuvre des formations semi ou complètement à distance. Donc, si l'on se réfère aux variables structurelles du modèle ASPI, on aura :

USTHB	1 (macro)
FEI	2 (méso)
Département INFO	3 (micro)
CAMPUS NUMERIQUE	1 (macro)
Module I.SY.I	2 (méso)
TP-SI-01	3 (micro)

Ce cours est ouvert essentiellement aux étudiants inscrits dans la formation d'ingénieurs en informatique assurée par le département d'informatique de la FEI. Mais il peut être aussi bénéfique pour des formations telles que le LMD.

4.2. Caractéristiques des acteurs, leur cahier des charges et leur rôle

Dans ce dispositif, les acteurs suivants ont été identifiés : les étudiants (apprenants), les assistants de TP (tuteurs) et l'enseignant du module I.SY.I (chargé de cours). Eventuellement, un administrateur ou gestionnaire de tous le campus numérique doit exister. Dans ce qui suit, nous décrivons les fonctions et rôles de chacun d'eux :

Apprenant : *Il est le bénéficiaire de cette formation et devra assumer à titre non exhaustif, les fonctions suivantes :*

- ▲ Assister aux séances de TP présentielle prévues 1 fois/semaine pour une durée de 1h30mn.
- ▲ Réaliser les tâches décrites dans les activités des trois périodes ou phases décrites précédemment.

- ▲ Utiliser le Wiki Individuel pour le travail individuel.
- ▲ Utiliser le Wiki de groupe pour le travail collectif
- ▲ Consulter l'état de ses travaux individuels ou au sein de son groupe à l'aide des feedbacks retournés en fin de chaque activité.
- ▲ Découvrir et comprendre les concepts du domaine des systèmes d'information, en particulier la méthode MERISE.
- ▲ Découvrir, comprendre et apprendre à programmer avec le langage CLIPPER.
- ▲ Explorer les différentes ressources mises à sa disposition dans le cours.
- ▲ Acquérir l'usage de quelques outils technologiques supportant les concepts étudiés.

Chargé de cours : *C'est le professeur chargé de préparer et de dispenser le cours I.SY.I sur les systèmes d'information et de préparer tous les supports nécessaires au bon déroulement du cours TP-SI-01. Ses fonctions consistent entre autres en :*

- ▲ Dispenser les cours et TD en présentiel du module I.SY.I.
- ▲ Préparer le support de cours-TP du dispositif TP-SI-01.
- ▲ Collaborer avec son équipe d'assistants de TP pour préparer les différentes activités, les ressources, les manipulations, ... etc.
- ▲ Répondre en direct aux questions des apprenants en séances de cours et TD.
- ▲ Communiquer aux apprenants leurs logins et mot de passe pour accéder au dispositif.

Tuteur : *A pour mission de guider les apprenants tout au long de leur apprentissage du TP en présentiel ou à distance qu'il soit individuel ou collectif. Parmi ses fonctions :*

- ▲ Evaluer le travail individuel de l'apprenant récupéré sur le Wiki individuel apprenant.
- ▲ Evaluer le travail collectif des groupes d'apprenants par l'intermédiaire des Wikis de groupe.
- ▲ Envoyer les FeedBack (FB) aux apprenants en précisant les aspects positifs et négatifs de leurs travaux, les appréciations, les attentes et d'éventuelles recommandations.
- ▲ Affecter une note à certains types d'activités ou tâches.
- ▲ Répondre aux questions posées via les forums.
- ▲ Orienter, conseiller et faire des recommandations aux apprenants sur leur TP.
- ▲ Noter le rapport final du TP et la démonstration de TP.

Administrateur ou Gestionnaire : *prendre en charges les aspects administratifs et les aspects système du dispositif. Parmi ses fonctions :*

- ▲ Procéder à l'inscription des adhérents au cours.
- ▲ Transmettre les logins et mots de passe à l'enseignant responsable du module.
- ▲ Intervenir pour tout problème de fonctionnement du dispositif.
- ▲ ... etc.

4.3. Caractéristiques individuelles des acteurs :

- ☼ **Visions :**

Tout le staff pédagogique travaille sous un même angle de vision, à savoir permettre à de futurs ingénieurs de faire leurs premières réalisations dans un module d'une grande importance pour la carrière d'ingénieur en informatique.

✿ **Pratiques :**

L'enseignant du cours doit être d'une grande expérience pratique surtout en matière de conception de systèmes d'informations touchant à des domaines divers.

Les tuteurs doivent avoir au minimum une bonne connaissance pratique du langage de programmation.

✿ **Connaissances :**

L'enseignant du module doit être connaisseur des systèmes d'informations et des différentes orientations pour leur développement.

Les tuteurs doivent connaître les principes de la méthode MERISE.

✿ **Souhaits :**

Par le cours TP-SI-01, le staff pédagogique souhaite d'abord palier à l'inconvénient d'une seule séance présentielle de TP par semaine pour des groupes de 20 apprenants au minimum.

Ce dispositif permettra de mieux guider les apprenants dans leur apprentissage et permettra d'atteindre des objectifs globaux de la formation d'ingénieurs, à savoir, former des ingénieurs plus compétents sur le plan opérationnel.

✿ **Besoins :**

Evidemment, pour un tel dispositif, on a besoin de moyens matériels et logiciels tels que les espaces de travail pour les tuteurs et les apprenants. Le campus numérique étant déjà installé, il reste à gérer et répartir les accès. De plus, pour les apprenants ayant des ordinateurs chez eux, un simple login et mot de passe leur permettront d'accéder au dispositif.

4.4. Les différentes dimensions du domaine

- **Pédagogie et formation :** ce cours fait partie d'une formation d'ingénieurs en informatique. Il renforce l'apprentissage individuel et collaboratif. La pédagogie de projet s'y trouve aussi grâce au projet de réalisation de TP. Certaines activités sont basées sur la pédagogie de la découverte (par exemple, les manips).
- **Technologie :** ce cours sera intégré par le biais du campus numérique de l'USTHB. Les activités se basent essentiellement sur l'usage des WIKI(s) qui sont des outils de rédaction de contenus pouvant être utilisés individuellement ou collectivement.
- **Organisation et gestion :** le cours est organisé par périodes gérés selon un calendrier pré-établi par l'équipe encadrant ce cours.
- **Discipline ou contenus spécifiques :** le cours concerne la partie TP du module I.SY.I dispensé en 2^{ème} année de la formation d'ingénieur en informatique.

4.5. Ressources

Pour chaque période, chaque activité et chaque tâche que l'apprenant devra effectuer, une série de ressources est mise à sa disposition. Dans ce qui suit, l'ensemble des ressources est présenté par période d'étude :

Période	Activité	Ressources
P1	Activité 1	○ Version d'évaluation d'un logiciel à expérimenter (logiciel compressé au format ZIP).

	Activité 2	<ul style="list-style-type: none"> ○ Support de cours du langage CLIPPER. (format pdf). ○ Le langage CLIPPER, logiciel compressé au format « ZIP ».
	Activité 3	<ul style="list-style-type: none"> ○ Grille du format des réponses. (format word ou pdf)
P2	Activité 1	<ul style="list-style-type: none"> ○ Manipulations #01, #02, #03, #04 et #05. (format word ou pdf)
	Activité 2	<ul style="list-style-type: none"> ○ Série d'exercices à programmer. (format word ou pdf)
P3	Activité 1	<ul style="list-style-type: none"> ○ Sujet de TP à réaliser. (format word ou pdf)
	Activité 2	<ul style="list-style-type: none"> ○ Prototype du rapport de TP. (format word ou pdf)
	Activité 3	<ul style="list-style-type: none"> ○ Explication du procédé de préparation des disquettes ou CD-ROM pour le jour de la démonstration. (format word ou pdf).

4.6. Annexes à fournir aux acteurs du projet

En plus des ressources documentaires fournies aux apprenants, les éléments suivants seront disponibles dans TPSI 01 :

- **Forum du cours TP-SI-01** : il permet à l'apprenant de poser ses questions à tout moment.
- **WIKI Individuel** : permet de déposer le travail individuel relatif aux tâches individuelles de l'apprenant.
- **WIKI de groupe** : permet d'effectuer les tâches collectives et favorise les échanges dans le groupe surtout si ce dernier est bien coordonné.
- **FAQ** : les foires aux questions peuvent s'avérer très utiles surtout lorsqu'il s'agit de chercher des réponses à des questions détaillées et précises.

5. Pilotage de l'innovation

5.1. Caractéristiques innovantes du projet

TP-SI-01 mixe entre deux formes d'enseignements : présentiel et à distance. Cependant, la proportion consacrée à la formation à distance est beaucoup plus importante en termes d'activités et de volume de temps consacré.

Dans ce dispositif, les séances présentiels servent à coordonner et à clarifier certains aspects du cours TP avant de passer à l'apprentissage à distance. Cette façon de faire décharge les apprenants d'assister à « trop » de séances présentiels mais les responsabilise d'avantage quand au travail individuel et/ou collectif à fournir pour atteindre les objectifs de formation.

5.2. Conditions d'intégration de l'innovation

L'engagement de tous les acteurs du dispositif est primordial pour sa bonne marche. Dès les premières séances présentiels, les apprenants en particulier doivent être préparés à cette forme d'étude où ils devront compter en bonne partie sur eux même.

L'équipe enseignante est sensé connaître tout ce qui concerne la formation et le travail de suivi à faire. TP-SI-01, encadre chaque période d'enseignement par tous les éléments nécessaires (recommandations, ressources, critiques et orientations) pour faciliter l'intégration des apprenants dans le dispositif.

Les enseignants du staff doivent comprendre que TP-SI-01 est à sa première version et vise à améliorer l'enseignement dans les TP d'où le suivi rigoureux du planning établi est primordial. La quantité de travail en présentiel sera réduite si l'effort fourni pour l'encadrement à distance est conséquent.

5.3. Conditions de pilotage

Afin de pouvoir piloter efficacement ce dispositif, le respect rigoureux de chacun des acteurs pour ses fonctions et son rôle est demandé.

Pour les apprenants, le respect des consignes tels que les délais contribuera à l'accomplissement des tâches à temps. Pour les enseignants et tuteurs, veiller au bon déroulement des sessions présentiels et distantes dans les temps, la remise des feedbacks, prévoir ou remédier à tout retard sans altérer les parties qui vont suivre dans l'enseignement sont autant de conditions pour un bon pilotage du projet. Sans oublier une partie qui est peut-être invisible et qui consiste au bon fonctionnement du système sur les plans matériel, logiciel et une bonne qualité de connexion.

Etant à sa version 1, TP-SI-01 vise à s'étendre plus tard pour couvrir tout le module I.SY.I, d'où la maîtrise de la partie TP donnera des indicateurs très pertinents pour le futur projet.

5.4. Les modalités d'évaluation du dispositif

Un ensemble d'actions insérées dans le dispositif permettront l'évaluation de ce dernier, entre autre :

- L'analyse des questions posées par les apprenants au travers des activités et tâches permet d'extraire des informations pertinentes pour l'évaluation du dispositif.
- Les questions posées à travers les forums et FAQ ainsi que les discussions dans les Wikis de groupes peuvent aussi éclairer les décideurs du dispositif.
- Les séances présentiels sont aussi une riche source d'information pour l'évaluation de la partie distante.
- La comparaison des résultats obtenus par les étudiants utilisant le dispositif à ceux d'autres sections par exemple ne l'utilisant pas pourrait porter des éléments intéressants.