

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA  
RECHERCHE SCIENTIFIQUE**

**Université de 08 Mai 45 Guelma  
Faculté des sciences et de l'Ingénierie**



**Département de l'informatique**

**Présenté pour l'obtention du diplôme de**

**Magister en Informatique**

**Option : Intelligence Artificielle et Imagerie**

**Titre du Mémoire**

=====  
***Modélisation et standardisation des situations  
d'apprentissage collaboratif***  
=====

Présenté par :  
**Benhamdi Soulef**

Devant le Jury :

Président :	Dr Seridi Hamid	Université de Guelma
Rapporteur :	Dr Seridi Hassina	Université Badji Mokhtar d'Annaba
Examineur :	Dr Belleili Habiba	Université Badji Mokhtar d'Annaba
	Dr Atil Fadila	Université Badji Mokhtar d'Annaba

**2008**

# Remerciements

*Ce mémoire est le fruit d'un grand effort et d'un travail toute au long d'une année. Ce travail qui n'aurait pu voir le jour sans la participation, l'aide, les conseils, ou encore la présence de certaines personnes.*

*Que tous ceux qui, de près ou de loin, ont contribué, par leurs conseils, leurs encouragements ou leur amitié à l'aboutissement de ce travail, trouvent ici l'expression de ma profonde reconnaissance.*

*Mes vifs remerciements vont tout d'abord à mon encadreur Mme Seridi Hassina, docteur de l'Université d'Annaba, pour m'avoir encadré, guidé et jugé mes travaux. Je lui suis infiniment reconnaissante de m'avoir encouragée et soutenu dans ce travail.*

*Mes remerciements vont également à :*

*-Messieurs : Assasla Younes pour son aide, Guerziz Ibrahim pour son encouragement, Doghmen Nabil pour le soutien moral.*

*-Mesdemoiselles : Leila, Rima, Amel et Naima Misoulhi auprès desquelles j'ai toujours trouvé l'encouragement et le soutien moral.*

*Pour leur encouragement, leur soutien moral et la patience qu'ils m'ont manifestée durant toute l'année, je remercie fortement tous les membres de ma famille.*

*Enfin remercier mes parents serait se répéter, citer leur affection serait pléonasme, parfois pour exprimer plus que ce qu'on a envie de dire on a recours au silence.*

# ***Dédicaces***

*« Louange à Dieu, le seul et unique »*

*Je remercie dieu de m'avoir donné le courage, et la force pour accomplir ce  
modeste travail que je dédie à:*

*Mes parents qui n'ont cessé de me soutenir et de m'apporter de la tendresse,  
l'amour et les conseils nécessaires pendant les moments les plus critiques.*

*Mes chers frères: Issam, Khaled et mes chères sœurs : Khaoula, Houda.  
A mes très chères amies : Affef, Randa, Nassira, Wassila, Leila, Rima, Amel, Naima  
Berkache et Naima Misoulhi.*

*A toute ma promotion ' PG Guelma ' : Issmahen, Saossen, Karima, Karim,  
Mouhammed, Rabeh, Abdellatif, Feissal, Abdessalem.*

*A la promotion des ingénieurs (2005) surtout : Djazia, Saossen, Wissem, Brahim,  
Younes, Nabil.*

*Avec l'expression de tous mes sentiments de respect,  
Je dédie ce modeste travail.*

## Résumé

Les recherches actuelles tendent vers la modélisation des applications du CSCL en utilisant les langages de modélisation éducatifs (EML). Le domaine du CSCL est pauvre en techniques de standardisation et de modélisation. Le langage le plus utilisé aujourd'hui est l'IMS LD réalisé par l'IMS Global consortium. L'IMS LD est un langage neutre permettant la description des processus d'apprentissage de façon formelle pour être traité automatiquement par des outils logiciels.

La caractéristique principale de l'apprentissage collaboratif 'CSCL' est de promouvoir la collaboration entre apprenants par la spécification, la modélisation et l'implémentation de situations d'apprentissage collaboratives. L'IMS LD ne permet pas de définir comment les apprenants collaborent et lors de quelles activités. Il indique seulement s'il existe une collaboration entre les apprenants ou non. Dans le but de décrire l'activité de collaboration en utilisant un cadre formel, nous proposons une approche pour la spécification et l'implémentation d'unités d'apprentissages collaboratives. Le concepteur décrira les activités de collaboration en se basant sur un ensemble de modèles prédéfinis et formalisés.

L'unité d'apprentissage collaborative sera jugée incomplète si elle se résume à une simple description des activités. En effet, une détermination des services et des outils nécessaires pour supporter ces activités est requise. Pour l'implémentation de ces services, nous proposons l'utilisation des composants logiciels et plus précisément les composants Fractals émergents pour améliorer la réutilisation et la flexibilité de l'application logicielle.

## Abstract

Actual researches try to modelling applications of CSCL, which is our domain, using educational modelling languages (EML), because this domain is poor in modelling standard technics. In our days, the most useful language is: IMSLD developed by IMS Global Consortium. This language formalizes learning process to be run by software tools.

However, the main feature of CSCL is the collaborative activity, IMSLD can't describe how students collaborate and in which activity, it only indicates if collaboration exists or not, we propose an approach of specification and implementation of collaborative units of learning. The designer will describe the collaborative activities basing on a set of models previously defined and formalized. The unit of learning will be judged incomplete, if it only contains a simple description of the activities. Effectively, a statement of the services and the tools indispensable to support these activities is required. We proposed to use software components specially fractal ones to implement these services. Software components are used to improve the reuse and flexibility of software applications.

# Table des matières

## Introduction générale

---

---

### Partie I : Contexte du travail et problématique

---

---

#### Chapitre 1- Contexte et état de l'art des EIAH : Environnements Informatique d'Apprentissage Humain

1.Contexte du travail et problématique .....	1
1.1L'enseignement assisté par ordinateur (EAO).....	1
1.1.1 EIAO <sub>1</sub> : enseignement intelligemment assisté par ordinateur .....	2
1.1.2 EIAO <sub>2</sub> : Enseignement Interactif Assisté par Ordinateur.....	2
1.1.3 EIAO <sub>3</sub> : Enseignement Interactif d'Apprentissage avec ordinateur.....	2
1.1.4. EIAO <sub>4</sub> : Enseignement Implexe Assisté par Ordinateur.....	3
1.2. Formation Ouverte et A Distance.....	3
1.2.1. EAD : Enseignement A Distance.....	3
1.2.2. FAD : Formation A Distance.....	3
1.2.3. FOAD : Formation Ouverte et A Distance.....	4
1.2.4. E-formation ou e-learning .....	5
1.2.5. L'apprentissage collaboratif dans la FAD.....	5
2. Les plates formes d'enseignement.....	5
3. Les normes pour le e-learning.....	7
3.1. Learning Object Model (LOM).....	7
3.2.Sharable Content Object Référence Model(SCORM).....	7
3.3 Les normes IMS.....	9
3.3.1 IMS simple sequencing.....	9
3.3.2 IMS Learner information package.....	10
3.3.3 IMS Content Packaging.....	10
3.3.4 IMS Learning Design.....	11
4. Conclusion .....	12

#### Chapitre 2- Méthodes et outils pour la conception des situations d'apprentissage collaboratives

1.Introduction.....	13
2. Le Contexte général.....	13
2.1. L'Environnement Informatique d'Apprentissage Humain (EIAH).....	13
2.2.Le domaine du CSCL .....	13
3. L'ingénierie centrée sur les scénarios.....	16
3.1. Les apports des approches à base de scénarios.....	16
4. Le langage de modélisation pédagogique.....	16
4.1. IMS Learning Design .....	17
4.2. Editeurs de Learning design.....	20
4.3.Les Players .....	21
4.4.Impact d'IMSLD .....	22
4.5.Les défis d'IMSLD.....	23
5. Quelques approches du CSCL.....	23
5.1. la démarche de modélisation pour une chaîne éditorial de scénarios.....	23
5.1.1. Principe séparé du PIM et génération du PSM.....	24
5.1.2. Formalisation selon IMSLD.....	24

5.1.3. Formalisation de scénarios d'apprentissage collaboratif selon le modèle de participation.....	25
5.2. Méthode scénarisation des activités collaboratives prenant en compte la mobilité de l'utilisateur.....	26
5.2.1. Scénarisation pédagogique collaborative et mobilité.....	26
5.2.2. Le processus de scénarisation.....	27
5.2.2.1. La définition des aspects.....	27
5.2.2.2. Proposition du processus de scénarisation.....	27
5.3. Une approche basée sur les CLFPs pour la conception des scénarios d'apprentissage collaboratif.....	28
5.3.1. Collaborative Learning Flow Patterns.....	28
5.3.2. L'extension d'IMSLD pour CSCL.....	29
5.3.3. Vers une création d'une unité d'apprentissage basée CLFPs.....	30
5.3.4. Mise en oeuvre.....	30
6 L'IMS LD et le CSCL.....	31
7. Conclusion.....	31

### **Chapitre 3- Les EIAH et les techniques de Génie logiciel**

1. Introduction.....	32
2. L'apports des approches et des technique du Génie Logiciel (GL) aux EIAH.....	32
2.1. L'apport de l'approche basée composant aux EIAH.....	32
2.2. L'apport de l'approche orientée patrons aux EIAH.....	33
3. L'approche basée composant et les EIAH.....	33
3.1. Définition du composant.....	33
3.2. Classification des composants et EIAH.....	34
3.2.1. Les composants logiciels pédagogiques.....	34
3.2.2. Les composants logiciels de service.....	34
3.2.3. Les composants logiciels techniques.....	35
3.2.4. Les composant logiciels de fabrication.....	35
3.3. Le Modèle Fractal.....	35
3.3.1. Principe.....	35
4. Les composants logiciels et les services web.....	39
5. L'approche orientée patron (pattern).....	40
5.1. Définition du patron.....	40
6. Les patrons et l'IMS LD.....	41
7. Conclusion.....	41

---



---

## **Partie II : Conception, implémentation et mise en œuvre**

---



---

### **Chapitre 4 Spécification et scénarisation des situations d'apprentissage collaboratives**

1. Introduction.....	42
2 Les modèles de situations d'apprentissage collaboratifs (jigsaw, pyramid,...).....	42
2.1 Le modèle pyramid.....	43
2.2 Le modèle jigsaw.....	43
3 Approche méthodologique de conception de structures pédagogiques.....	43
3.1. Architecture du modèle de scénarisation de situations d'apprentissage collaboratif.....	44
3.1.1. Le choix du modèle d'activité.....	46
3.1.2. Particularisation du scénario.....	47
3.1.3. Création d'unité d'apprentissage.....	52

3.2. La structure d'une unité d'apprentissage en IMSLD.....	54
3.2.1. Les éléments d'un manifest.....	55
3.2.1.1. Les rôles : <i>roles</i> .....	55
3.2.1.2. Les activités : <i>activities</i> .....	56
3.2.1.3. La méthode : <i>method</i> .....	56
3.2.1.4. Les ressources : <i>resources</i> .....	57
4. Conclusion.....	57

## **Chapitre 5 Implémentation d'un scénario de collaboration fondé sur les composants logiciels**

1. Introduction .....	58
2. Schéma fonctionnel du système.....	58
3. Les outils utilisés.....	58
3.1 Serveur Web.....	59
3.2. Le choix de langage de programmation .....	59
3.3 Environnement de programmation .....	59
4. Implémentation du prototype.....	60
4.1. Barre de menu .....	61
4.2. Fenêtre d'affichage.....	62
4.3. Les interfaces de l'outil.....	62
4.3.1. Interface Questionnaire.....	62
4.3.2. Interface Modèle.....	63
4.3.3. Interface du nombre de niveaux .....	65
4.3.4. Interface d'activité.....	65
4.3.5. Interface Ressource.....	68
5. Implémentation des manifests.....	69
6. l'implémentation du service chat .....	72
7. Exécution d'une unité d'apprentissage par CopperCore.....	75
8. Conclusion.....	78

## **Chapitre 6 Exemple d'une unité d'apprentissage dédiée à la collaboration**

1. Introduction .....	79
2. Un exemple d'une unité d'apprentissage.....	79
3. Conclusion .....	87

## **Conclusion et Perspectives**

### **Bibliographie**

Annexe A Java, un langage de programmation  
 Annexe B XML (eXtensible Markup Language)  
 Annexe C UML (Unified Modeling Language)  
 Annexe D Le Questionnaire

# Listes des figures

Figure1 : Exemple d'architecture de plate forme pour la FAD .....	6
Figure2 : L'environnement d'exécution de SCORM.....	8
Figure3 : IMS Content Packaging scope.....	11
Figure4 : Modèle conceptuel de la structure de tout Learning Design.....	18
Figure5 : Implémentation (binding) de la structure contenant les concepts d'IMS LD.....	19
Figure6: Schéma de l'extension proposée pou l'IMSLD ( <i>groupservice</i> ).....	29
Figure7 : Schéma descriptif d'un composant.....	35
Figure8 : Schéma descriptif des interfaces.....	36
Figure9 : Composant Fractal avec les différents binding.....	37
Figure 10 : Etapes de scénarisation d'une ressource dédiée à l'apprentissage collaboratif.....	44
Figure11 : Schéma de conception.....	45
Figure12: L'extraction du modèle de collaboration.....	46
Figure13 : le modèle de collaboration : Pyramid.....	49
Figure14: Le modèle de collaboration jigsaw.....	51
Figure15 : Diagramme d'activité UML représentant le modèle de collaboration:Jigsaw.....	52
Figure16 : Définition des ressources et des services.....	52
Figure17 : Architecture du composant chat.....	53
Figure18 : La structure d'une unité d'apprentissage.....	54
Figure19: Les éléments d'un manifest.....	55
Figure20: Diagramme en arbre dérivant l'élément <i>roles</i> .....	56
Figure21 :Diagramme en arbre dérivant l'élément <i>activities</i> .....	56
Figure22 : Diagramme en arbre dérivant l'élément <i>method</i> .....	57
Figure23 : Diagramme en arbre dérivant l'élément <i>resources</i> .....	57
Figure24 : Schéma fonctionnel du système.....	58
Figure25 : l'Interface de l'environnement eclipse.....	60
Figure26:Interface principale de l'outil.....	61
Figure27 : Interface du questionnaire.....	63
Figure28 : Interface du modèle pyramid.....	64
Figure29 : Interface du modèle jigsaw.....	65
Figure30 : Interface du nombre de niveaux.....	65
Figure31 : Interface d'activité ' pyramid avec deux niveaux'.....	66
Figure32 : Interface d'activité ' pyramid avec trois niveaux'.....	67
Figure33 : Interface d'activité ' pyramid avec quatre niveaux'.....	68
Figure34 : Interface de création de ressources.....	69
Figure35 : la fenêtre principale du composant chat.....	74
Figure36:fenêtre du client.....	74
Figure37 : Fenêtre du serveur.....	75
Figure38 : la publication de l'UOL.....	76
Figure39 : interface des utilisateurs.....	76
Figure40 : interface d'exécution de l'UOL.....	77
Figure41 : L'utilisation du service Chat.....	77
Figure42 : Les réponses choisies par l'enseignant.....	80
Figure43: choix du nombre de niveaux.....	81
Figure44 : définition des informations nécessaires pour le rôle : enseignant (niveau1).....	81
Figure45 : définition des informations nécessaires pour le rôle : étudiant (niveau1).....	82
Figure46 : définition des informations nécessaires pour le rôle : enseignant (niveau2).....	83
Figure47 : définition des informations nécessaires pour le rôle : groupe étudiant (niveau2).....	83
Figure48 : Création du cours d'architecture.....	84



# *Listes des Tableaux*

Tableau1 : la définition du modèle pyramid et le modèle 10 d'UP.....	48
Tableau2 : Play d'un learning design basé sur le modèle :Pyramid.....	50
Tableau 3: Diagramme d'activité UML représentant un exemple de scénario.....	79

# **Introduction Générale**

## 1. Introduction :

L'évolution actuelle des systèmes de formation à distance va vers plus de personnalisation et de réutilisation dans un environnement informatique marqué par les technologies du web.

Aujourd'hui des normes et standards émergents s'imposent dans le domaine du e-learning. En effet, les cours, les produits et différents services qui empruntent les voies électroniques d'Internet circulent à travers de multiples systèmes ou plates-formes logicielles et épousent des formats ou des structures variées. Aussi, vu le nombre et la diversité du matériel numérisé et afin de permettre aux utilisateurs un repérage, une distribution, un échange et des mises à jour de ce matériel, les concepteurs des objets pédagogiques adoptent des normes et des standards qui se présentent comme des langages communs pour décrire des ressources pédagogiques numérisés. Ces langages sont interprétables par les machines et les humains et offrent une interopérabilité des systèmes éducatifs. Nous ne manquons pas de citer :

- L'organisme de normalisation l'IEEE LTSC (IEEE, 2001) qui définit le LOM.
- SCORM (Sharable Content Object Reference Model) est un modèle réalisé à partir d'une initiative de l'ADL (Advanced Distributed Learning) (ADL, 2001).
- IMS Learning Design (IMS LD) est un langage qui favorise la description des processus d'apprentissage. Il aide les concepteurs pédagogiques à modéliser le contenu, les rôles des différents acteurs dans une activité pédagogique et les services nécessaires pour atteindre les objectifs d'apprentissage. Il soutient également les divers types d'apprentissage collaboratif. Cet aspect est particulièrement important pour sa reconnaissance à la fois dans le cadre de la formation à caractère commercial et globalement dans l'éducation.

Nos travaux se focalisent sur l'utilisation des langages de modélisation et nous cherchons à mieux analyser la notion de scénarisation en se basant sur le langage de spécification IMS LD pour être capable de formaliser ensuite des scénarios pédagogiques beaucoup plus réutilisables et flexibles. L'innovation et la contribution principale de ce projet résident dans l'étude même de la notion de scénarisation en se basant sur les modèles d'activité qui servent à décrire la collaboration entre les apprenants et l'utilisation des composants logiciels pour implémenter les services indispensables pour ce type d'activité. Nous ne manquerons pas d'étudier le standard IMS LD qui est appelé aujourd'hui à créer des cours pour le e-learning qui soient interopérables, flexibles, effectives, attractive et personnalisés.

## **2. Problématique :**

Le rôle de l'enseignant est très important dans le processus. Il doit non seulement préparer le cours, déterminer les différentes activités à réaliser, identifier les différents rôles qui vont être joués durant l'apprentissage, mais aussi, il doit contrôler le travail des apprenants et les assister en répondant à leurs éventuelles questions. L'idée est donc de faciliter le travail de l'enseignant le plus possible. Nous nous intéressons plus particulièrement à la modélisation des situations d'apprentissage collaboratives.

Le travail présenté dans ce mémoire se situe à l'articulation de deux problématiques :

1. Etudier certains modèles de collaboration qui permettent de décrire l'activité de collaboration et proposer une démarche permettant d'assister l'enseignant dans son travail.
2. Etudier le problème lié au fait que, lors de l'apprentissage, les apprenants ont besoin d'utiliser des services qui leurs permettent de réaliser les activités définies par l'enseignant. Les services peuvent être réutilisés par les enseignants et intégrés sous forme de composants logiciels dans les unités d'apprentissage.

## **3. Hypothèse:**

Notre travail repose sur l'hypothèse que l'enseignant en tant que concepteur pédagogique, doit concevoir des situations d'apprentissage collaboratives en choisissant l'un des modèles de collaboration existants ainsi que le langage de modélisation permettant de la normaliser, alors que l'enseignant n'a aucune idée sur ces modèles. La modélisation d'un scénario d'apprentissage par un langage de modélisation quelconque requiert en l'absence d'outil d'aide à la conception, des compétences d'ingénierie pédagogique et informatique. Comme nous l'avons évoqué, les apports de la collaboration sont multiples. De ce fait, nous avons consacré notre travail à l'étude de la collaboration, ou encore l'apprentissage collaboratif. Notre objectif est de proposer une approche qui rehausse l'apprentissage collaboratif en dotant un enseignant scénariste de situations d'apprentissage d'un outil lui offrant un moyen d'identifier un modèle de scénario pédagogique, de le normaliser et de le particulariser sans pour cela être expert en psychopédagogie ou en matière de standard émergeant dans le domaine du e-learning.

## **4. Méthodologie :**

La méthodologie que nous avons suivie comprend trois phases :

### ***1. Etude et compréhension :***

Cette phase correspond à l'étude des travaux de recherche en premier lieu sur notre domaine de travail : l'EIAH puis le CSCL qui est dérivé de ce dernier. Puis, on a étudié le langage de modélisation l'IMSLD qu'on a adopté pour décrire de façon formelle le processus d'apprentissage. Par la suite, nous avons étudié les travaux sur les modèles qui permettent de décrire le processus de collaboration entre les individus. Ces modèles sont appelés : modèles de collaboration.

Cette étude nous a permis de mieux comprendre notre domaine de recherche : le langage IMSLD, les modèles de collaboration.

### ***2. Proposer une approche :***

Notre approche de conception est constituée de trois étapes, que nous présentons par la suite :

#### **Etape 1 : Définition du modèle de collaboration :**

Les modèles de collaboration permettent de décrire l'enchaînement des activités qui doivent être réalisées par les individus lors de l'apprentissage. L'idée est donc de poser un ensemble de questions à l'enseignant. Cet ensemble correspond à un ensemble d'objectifs, l'enseignant doit choisir l'un parmi ces objectifs. Suivant les choix de l'enseignant, on peut déterminer le modèle de collaboration qui permet de satisfaire ses besoins. En effet, l'enseignant doit non seulement choisir l'objectif de l'apprentissage, mais il doit aussi spécifier le modèle de scénario approprié, pour cela on doit donc poser un deuxième questionnaire à l'enseignant. Ce questionnaire correspond à l'ensemble des modèles de scénario. Ces modèles seront utilisés comme des ressources durant le processus d'apprentissage.

#### **Etape 2 : Particularisation du scénario :**

Dans cette étape le manifest doit être adapté suivant le modèle de collaboration et le modèle de scénario définis dans l'étape précédente. La particularisation du manifest consiste réellement en l'ajout de certaines informations liées aux modèles spécifiés à partir des réponses de l'enseignant et qui n'existaient pas dans le manifest. Ces informations permettent de donner plus de détail sur le travail aux apprenants.

### **Etape 3 : Définition des ressources et services :**

Le processus de l'apprentissage nécessite un certain nombre de ressources. Ces ressources peuvent être utilisées par l'apprenant au moment de l'apprentissage pour l'aider dans son travail. Les apprenants ont besoins aussi d'outils ou de services (service de différents types tels que : les services de communication). Cette définition se traduit ensuite par l'ajout de ces ressources et ces services dans le manifest.

Nous détaillerons notre approche de conception dans la troisième partie.

#### **3. La réalisation de l'outil :**

La dernière phase de notre travail est l'implémentation d'un outil permettant à l'enseignant de concevoir des situations d'apprentissage collaboratives suivant l'approche proposée.

L'outil qu'on a développé décharge, l'enseignant du fait qu'il ne sera pas obligé de choisir le modèle de collaboration qu'il doit utiliser lors de la conception, du moment qu'il ne peut avoir aucune connaissance sur ces modèles. l'enseignant sera soumis à un questionnaire pour connaître son objectif, et par conséquent, déterminer le modèle de collaboration approprié. Le résultat est donc une unité d'apprentissage qui peut être exploitée par l'apprenant en l'exécutant par un *playe* ( *coppercore* par exemple) .

### **5. Organisation du mémoire :**

-Dans le premier chapitre, nous présentons l'origine de notre domaine de recherche EIAH (Environnement Informatique pour l'Apprentissage Humain), partant de l'EAO (Enseignement Assisté par Ordinateur), et les sigles qui lui sont associés : EIAO<sub>1</sub>,...,EIAO<sub>4</sub>, passant par la formation à distance (FAD) et son évolution vers la formation ouverte et à distance(FOAD), ainsi que l'apport que revêt l'apprentissage collaboratif dans la FAD.

-Le deuxième chapitre est consacré à la description du domaine au sein duquel se déroule notre travail : le CSCL, qui dérive de l'EIAH, avec une présentation de ce dernier. Dans ce chapitre, nous définissons aussi le langage de modélisation que nous avons utilisé ainsi que quelques approches du CSCL qui ont été proposées pour modéliser des scénarios collaboratifs.

-Le troisième chapitre concerne la définition de quelques approches issues du domaine de Génie Logicielle : l'approche basé composant et l'approche orientée patterns, et l'apport de leur utilisation dans l'EIAH. Nous terminons par une description des services web afin de les distinguer des composants logiciels.

-Le quatrième chapitre présente les deux modèles de collaboration introduits dans notre approche : Pyramid et Jigsaw, une petite définition de l'ingénierie centrée sur les scénarios et son

apport, nous exposons par la suite la relation entre le formalisme IMSLD et le CSCL. Enfin, les différentes étapes de l'approche que nous avons proposé afin de concevoir et modéliser des situations d'apprentissage collaboratives.

-Nous avons présenté les différents outils utilisés pour l'implémentation de notre outil ainsi que le prototype réalisé, dans le cinquième chapitre.

-le sixième chapitre, nous exposons les différentes étapes de création d'une unité d'apprentissage en utilisant l'outil implémenté.

-Nous terminons par une conclusion et des perspectives.

-L'annexe propose plusieurs élément évoqués dans la mémoire tels que : le langage de programmation Java et le langage XML, le langage de modélisation UML avec la présentation de certains de ses diagrammes (diagramme de classe, d'activité,...) et le questionnaire proposé à l'enseignant.

# **Partie I**

**Chapitre1 : Contexte et état de l'art sur les Environnements Informatique d'Apprentissage Humain (EIAH)**

**Chapitre2 : Méthodes et outils pour la conception des situations d'apprentissage collaboratif**

**Chapitre3 : Les EIAH et les techniques de Génie Logiciel**



# **Chapitre 1**

## **Contexte et état de l'art sur les Environnements Informatique d'Apprentissage Humain**

## 1. Contexte du travail :

Depuis plusieurs années, le glissement des EIAO vers des environnements permettant l'interaction entre des machines et des humains distribués dans l'espace, a permis l'apparition de ce qu'on appelle « les Environnement Informatique pour l'Apprentissage Humains » (EIAH).

L'e-formation est née grâce à ces environnements et l'apparition des technologies de l'information et de la communication appliquées à l'éducation (TICE) ainsi que les grands réseaux d'information. La formation à distance (FAD) trouve un renouveau avec la e-formation. Dans ce chapitre, nous définissons l'EAO ainsi que les sigles qui lui sont liés (EIAO<sub>1</sub>,..., EIAO<sub>4</sub>). Par la suite, nous évoquerons l'évolution de la formation à distance vers la formation ouverte et à distance en utilisant les TICE<sup>1</sup>. Enfin, nous allons donner une définition des plates – formes d'enseignement.

**1.1 L'enseignement assisté par ordinateur (EAO) :** Depuis plusieurs années, de nombreuses recherches ont eu pour objectif d'utiliser les machines en éducation. Les premières tentatives sont apparues dans les années 50 [Geo, 01]. La Construction et l'utilisation de ces machines étaient très compliquées. Dans l'enseignement programmé par ordinateur développée en 1970. L'acte de l'apprentissage est considéré comme une réaction par rapport à l'environnement. L'apparition de l'ordinateur donne la naissance à L'EAO et offre la possibilité de l'individualisation de l'enseignement.

Les années soixante ont été marquées par l'apparition des premiers systèmes informatique tels que les systèmes de type « PLATO » avec présentation imagée et variée de l'information, suivie de questionnaire de différents types. Ces systèmes nommées « si..alors..Sinon » étaient très coûteux à produire et rigide à utiliser car les éditeurs étaient moins sophistiqués à cette époque. L'EAO est un domaine pluridisciplinaire et dépasse largement le domaine de l'informatique, ce qui signifie qu'il dépend des résultats des recherches d'autres disciplines telles que : la didactique, la psychologie.... Ce domaine comprend aussi le sigle « EIAO » avec ses différentes significations :

- EIAO<sub>1</sub> : Enseignement Intelligemment Assisté par Ordinateur.
- EIAO<sub>2</sub> : Enseignement Interactif Assisté par Ordinateur.
- EIAO<sub>3</sub> : Environnement Interactif d'Apprentissage par Ordinateur.
- EIAO<sub>4</sub> : Enseignement Implexe Assisté par Ordinateur.

---

<sup>1</sup> Technologies de l'information et de la communication

### ***1.1.1 EIAO<sub>1</sub> : enseignement intelligemment assisté par ordinateur :***

Les limites constatées des systèmes d'EAO et des recherches menées en intelligence artificielle (IA), en science de l'éducation et en sciences cognitives ont conduit à l'émergence du sigle EIAO<sub>1</sub> aux états –Unis, avec le célèbre système SCHOLAR de Carbonell , d'où la connaissance représentée dans des structures d'information étaient utilisées pour déterminer des questions à poser à l'apprenant. Pour vérifier ses réponses et pour répondre à des questions de ce dernier, non explicitement prévues à sa conception. [Mon ,95]

Les systèmes tuteurs intelligent (STI ou en anglais : intelligent tutoring system) sont apparus dans les années 80, étaient fortement liés au développement des systèmes à base de connaissance en IA, en particulier les premiers « systèmes experts» sur le modèle du système GUIDON de W.J.Clancey à la fin des années 70.

### ***1.1.2 EIAO<sub>2</sub> : Enseignement Interactif Assisté par Ordinateur :***

Les approches qu'on a déjà présentées précédemment sont toutes centrées sur le processus de l'enseignement, d'où les systèmes cherchant principalement à jouer le rôle de l'enseignant (ITS). Dans d'autres approches développées en parallèle, l'ordinateur est considéré comme un moyen permettant un apprentissage par la découverte et l'exploration en donnant le contrôle à l'apprenant. Ces approches sont basées sur les théories constructivistes de l'apprentissage, selon laquelle l'apprenant construit ses connaissances en interagissant avec un environnement, ce qui signifie que l'apprenant et son apprentissage sont mis en premier lieu. Dans ce sens, le terme : « micromonde » est apparu, l'idée était donc de permettre aux apprenants de manipuler des concepts abstraits et construire leurs propres connaissances en agissant sur des objets d'un monde simplifiés dans des environnements créés auparavant.

### ***1.1.3. EIAO<sub>3</sub> : Enseignement Interactif d'Apprentissage avec ordinateur :***

Ces environnement sont créés en combinant les deux dernières approches : l'apprentissage par résolution de problème (EIAO<sub>1</sub>) et l'apprentissage par la découverte (EIAO<sub>2</sub>). En effet, l'arrivée du terme « Interactif » signifie que ces environnement s'intéresse beaucoup plus à l'interactivité qui recouvre d'une part les activités qu'ils permettent à un apprenant et d'autre part leurs capacités d'intervention et d'adaptation à leur utilisateur [Bar 91]. Le glissement du terme « enseignement » vers « apprentissage » révèle qu'on s'intéresse plus à la construction des connaissances par l'apprenant. Le terme « avec ordinateur » souligne que la machine accompagne l'apprenant tout au long du processus de l'apprentissage.

#### **1.1.4. EIAO<sub>4</sub> : Enseignement Implexe Assisté par Ordinateur :**

L'équipe SEVE [Cla, 88] a donné une acceptation du sigle EIAO par « Enseignement Implexe Assisté par Ordinateur », qui signifie qu'une telle représentation de l'enseignement ne peut être une seule représentation dans le temps (d'où le terme implexe). Elle est indépendante des mouvements technologiques et s'intéresse au processus de l'enseignement dans son totalité.

#### **1.2. La Formation Ouverte et A Distance :**

Avant de passer à la définition de la FOAD, il faut tout d'abord définir les termes « Enseignement », « Apprentissage » et « Formation ». Le premier a pour objectif de transmettre le savoir. Il est organisé pour préparer, assister, suivre, contrôler et améliorer l'apprentissage. Ce dernier est le processus de transformation des informations acquises en entrée vers la sortie pour construire des connaissances chez un apprenant. La formation est un point de vue qui englobe les deux processus : l'enseignement (point de vue enseignant) et l'apprentissage (point de vue apprenant).

Dans cette partie, nous présentons une définition des sigles : EAD (Enseignement A Distance), FAD (Formation A Distance), et nous terminons par la présentation de la Formation ouverte et A Distance (FOAD). La formation est une relation triangulaire, d'où l'apprenant construit ses connaissances en interagissant avec son environnement, le formateur ou l'enseignant contrôlent l'activité de l'apprenant.

##### **1.2.1. EAD : Enseignement A Distance :**

C'est le sigle utilisé, il désigne le transfert des connaissances dans une situation éducative en dehors d'une situation en face à face ou en « présentiel ». Donc, il prend en compte la distance géographique, d'où l'usage des nouveaux outils de communication (messagerie, téléchargement de fichier,...) au lieu d'utiliser l'envoi postal [Gen 00].

##### **1.2.2. FAD : Formation A Distance :**

La formation à distance couvre l'ensemble des dispositifs techniques et des modèles d'organisation qui ont pour but de fournir un enseignement ou un apprentissage à des individus qui sont distants de l'organisme de formation prestataire du services [Swi,04]. En effet, il existe pas mal de définitions proposées par plusieurs spécialistes de la FAD. Keegan(1996), l'un de ces spécialistes, qui a proposé une définition de la FAD indépendamment de l'avènement des TICE et d'Internet :

La formation à distance est une forme d'enseignement caractérisée par :

- la séparation quasi-permanente entre le formateur et l'apprenant tout au long du processus d'apprentissage (ceci différencie la formation à distance de la formation présentielle);
- l'influence d'une organisation administrative aussi bien en ce qui concerne la planification et la préparation des matériaux pédagogiques que la mise à la disposition des apprenants des services d'accompagnement et de support (ceci différencie la FAD de l'autoformation);
- l'utilisation de médias techniques (imprimerie, audio, vidéo, ordinateurs) pour assurer le lien entre le formateur et l'apprenant et médiatiser le contenu de la formation;
- L'existence de mécanismes de communication bidirectionnelles afin que l'apprenant bénéficie mais prenne aussi l'initiative de dialogues avec le formateur (ceci distingue la FAD des autres usages de la technologie dans l'éducation);
- **la quasi-absence de la notion de groupe tout au long du processus d'apprentissage**, de sorte que les apprenants sont toujours vus comme des individus isolés et non comme faisant partie d'un groupe, avec la possibilité d'organiser occasionnellement des rencontres, soit en présentiel, soit via des moyens électroniques à des fins didactiques ou de socialisation.

### ***1.2.3. FOAD : Formation Ouverte et A Distance :***

Le Collectif de Chasseneuil (2000) sur les formations ouvertes et à distance, a proposé une définition de la formation à distance, en l'élargissant à la formation ouverte et à distance (FOAD).

« *La formation ouverte et à distance...*

- *est un dispositif organisé, finalisé, reconnu comme tel par les acteurs ;*
- *qui prend en compte la singularité des personnes dans leurs dimensions individuelle et collective ;*
- *et repose sur des situations d'apprentissage complémentaires et plurielles en termes de temps, de lieux, de médiations pédagogiques humaines et technologiques, et de ressources.etc.*

La Formation Ouverte et A Distance englobe ses concepts de formation ouverte (entre différents techniques d'enseignement) ainsi que la formation à distance. Cette approche, cherche à intégrer tous les éléments et toutes les phases du cycle de formation. Les débuts de ce mouvement d'intégration en Europe étaient l'apparition du concept du moteur d'enseignement [Cla,88] : système qui tente de répondre à toutes les phases de la formation, aussi bien dans la préparation (contenu et cursus), dans l'utilisation (apprentissage et soutien) et dans le suivi de la formation (pédagogique et administratif). Le domaine de la FOAD est approprié à l'introduction des technologies (CD-ROM, Internet, Intranet, etc) dans la

formation ainsi que les différents types de situations d'apprentissage (présentiel, à distance, autoformation). L'utilisation de l'Internet et les TICE dans la FOAD constitue ce que l'on désigne couramment par : e-formation. Grâce aux TIC, la FOAD devrait pouvoir aujourd'hui concilier économie (réduction des frais), réponse aux contraintes (individualisation de la formation) et efficacité (rapidité du déploiement à grande échelle).

#### ***1.2.4. E-formation ou e-learning :***

La e-formation peut être définie comme toute utilisation de l'Internet dans la gestion et la diffusion de la formation. Les concepteurs de la e-formation cherchent essentiellement un certain nombre de qualités comme : une distribution plus large et plus efficace de la formation, l'individualisation et l'adaptation de parcours de formation aux besoins individuels des apprenants, des possibilités de suivi et de tutorat, la rapidité,....

Bien que la e-formation permette de satisfaire un nombre important de besoins, elle pose des problèmes très généraux qui apparaissent aussi bien dans la formation en présentiel que pour certains dans la FAD.

#### ***1.2.5. L'apprentissage collaboratif dans la FAD :***

L'apprentissage collaboratif est considéré comme une activité essentielle dans la FAD, car il présente plusieurs avantages. Pour pallier certains problèmes considérés comme des écueils dans la FAD : l'isolement de l'apprenant, la perte de motivation,...., l'apprenant est intégré dans un groupe au lieu de travailler seul. L'activité de collaboration entre les différents membres du groupe suppose l'apparition de l'interaction sociale pour échanger leurs idées et apprendre ensemble. L'apprentissage permet de créer un cadre sociologique afin d'éviter l'isolement de l'apprenant, ce qui signifie donner plus de motivation à ce dernier.

## **2. Les plates formes d'enseignement :**

Le domaine de la formation à distance a bénéficié de l'évolution des technologies de l'information et de la communication éducative (TICE), du fait qu'elle apporte un plus value organisationnelle et pédagogique. Ces technologies offrent la possibilité d'accéder à des contenus distants et d'échanger des informations entre les différents acteurs (apprenant/enseignant, apprenants entre eux, enseignants entre eux).

Les outils de formation qui adoptent les réseaux informatiques comme support sont appelés : plates formes informatique pour la formation à distance ou encore plates forme d'enseignement (PFE ou en anglais : LMS Learning Management System). Ces plates formes sont développées dans le but de la consultation à distance des contenus pédagogiques,

l'individualisation de l'apprentissage et le tutorat à distance. Les PFE intègrent des dispositifs permettant de faciliter les activités réalisées par les différents acteurs de la FAD : enseignant, concepteur informatique, tuteur, apprenant et administrateur [Etu, 99]. On commence par le rôle de l'enseignant qui est chargé de concevoir le matériel pédagogique. Les médias (texte, image,...) sont alors réalisés par le concepteur informatique, qui peut être l'enseignant lui-même. L'enseignant conçoit ensuite les parcours pédagogiques typés et individualisés de son enseignement. Le tuteur prend en charge le suivi et le contrôle du travail de l'apprenant. Il peut éventuellement réguler leurs parcours pédagogiques. L'apprenant consulte, télécharge les contenus nécessaires pour son travail, effectue les activités qui lui sont demandées, ensuite il envoie son travail au tuteur pour l'évaluation. L'apprenant peut communiquer avec l'enseignant pour, par exemple, discuter sur le thème proposé. L'installation et la maintenance du système, l'inscription administrative des étudiants, la gestion des droits d'accès aux ressources pédagogiques, sont réalisés par l'administrateur.

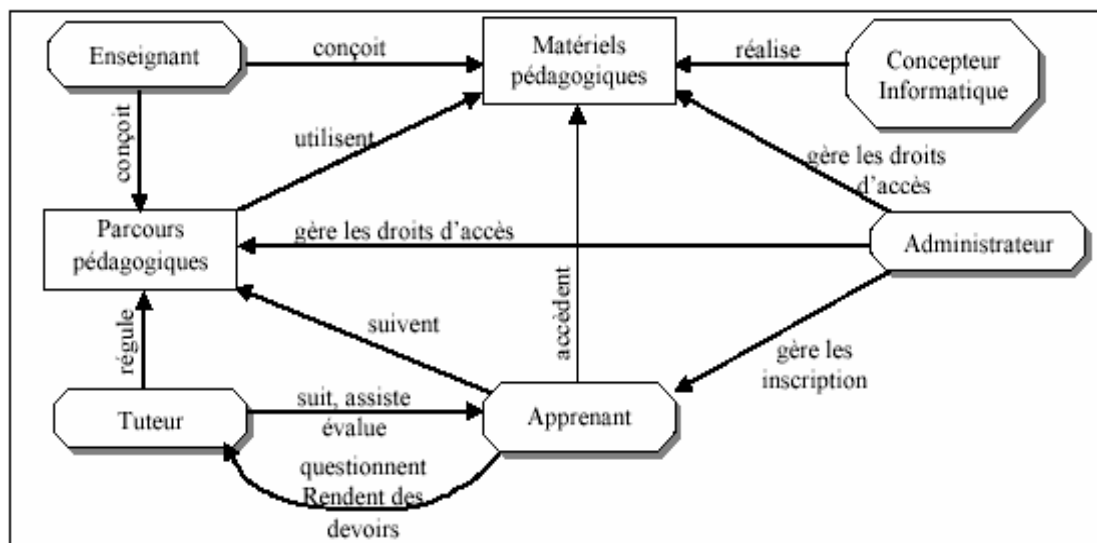


Figure 1: Exemple d'architecture de plate forme pour la FAD [Geo 01].

Cette figure présente un modèle de PFE avec cinq acteurs. Il existe des plates formes qui ne prennent pas en considération tous ces acteurs, alors que d'autres sont plus complètes. Dans certains cas, l'enseignant joue le rôle d'un concepteur informatique ou d'un tuteur [Geo ,01][Etu, 99].

### **3- Les normes pour le e-learning :**

Avant de déterminer l'intérêt des normes, on doit tout d'abord définir le terme « norme » : une norme est un ensemble de règles de conformité, étudié par un organisme de normalisation au niveau national et international [ste, 05].

Le domaine du e-learning permet de présenter aux apprenants un ensemble de cours et faciliter la création de ces cours pour les enseignants. La normalisation d'après Michel Arnaud : « a pour but d'assurer l'interopérabilité et la réutilisabilité des outils d'enseignement médiatisé et à distance, ainsi que des systèmes de description des contenus organisés en métadonnées ». La mise en œuvre des cours en utilisant une norme permet de les rendre présentable sur n'importe quelle plate forme, ce qui signifie que l'enseignant n'écrit qu'une seule fois son cours.

#### **3.1. Learning Object Model (LOM) :**

LOM est un standard qui permet la description des objets pédagogique selon 9 catégories et comporte 71 attributs.

Cette description est exprimée à l'aide d'un fichier XML, contenant les caractéristiques d'un document, ou le contenu d'un cours ou d'un exercice pour l'e-Learning.

Le schéma de métadonnées LOM est très détaillé et fournit un ensemble de vocabulaire de référence. Plusieurs organismes de standardisation (IEEE LTSC, IMS, ADL, ARIADNE) ont adopté le modèle LOM qui a été intégré le plus souvent dans des profils d'application (SCORM, CanCore, etc.).

Le modèle de métadonnée générique Dublin Core qui permet de décrire n'importe quel type de document a été intégré au LOM. [Ste, 05].

#### **3.2. Sharable Content Object Référence Model (SCORM) :**

SCORM est le modèle de référence pour le partage de « contenus » et définit un modèle d'agrégation permettant de créer et exécuter des objets pédagogiques structurés. SCORM regroupe un ensemble de spécifications techniques permettant aux systèmes d'apprentissage de réaliser plusieurs fonctionnalités telles que : trouver, importer, partager, réutiliser et exporter les contenus de manière normalisée.

##### **Découpage en plusieurs parties :**

SCORM peut être constitué de plusieurs parties:

*1. Modèle d'agrégation de contenu* : qui assure la promotion de méthodes cohérentes en matière de stockage, d'identification, de conditionnement d'échange et de repérage du contenu.



Ce modèle peut comporter plusieurs fonctionnalités : la première consiste en la définition des métadonnées LOM (Learning Object Metadata) qui permettent d'identifier les termes décrivant le contenu de l'objet d'apprentissage. La deuxième établit la liaison entre les métadonnées et les fichiers (XML), pour définir la manière de leur codage afin de les rendre compréhensibles par la machine. La dernière fonctionnalité définit comment empaqueter un ensemble d'objets d'étude, leurs métadonnées et des informations concernant le contenu.

**2. Environnement d'exécution :**

Décrit les exigences du système de gestion de l'application nécessaire à la gestion de l'environnement d'exécution. Il fournit [Mba, 03] :

- \_Un mécanisme commun de lancement des objets pédagogique.
- \_Un dispositif commun de communication entre ces objets et le système de gestion d'apprentissage (LMS : Learning Management System).
- \_Une base constituée d'un vocabulaire pour cette communication.

L'interface de programme (API) joue le rôle d'un intermédiaire entre un objet pédagogique et le LMS afin d'assurer la communication entre eux. Cette API informe le LMS de l'état de la ressource pédagogique (initialisée, terminée, en état d'erreur...). Il n'y a pas d'interaction entre les ressources pédagogiques «Assets» et le 'LMS'. Pour ceux, l'API n'est pas utile pour ce type de ressources. Le modèle de données fournis par l'environnement d'exécution définit un vocabulaire commun pour la communication entre le LMS et l'objet pédagogique (SCO).

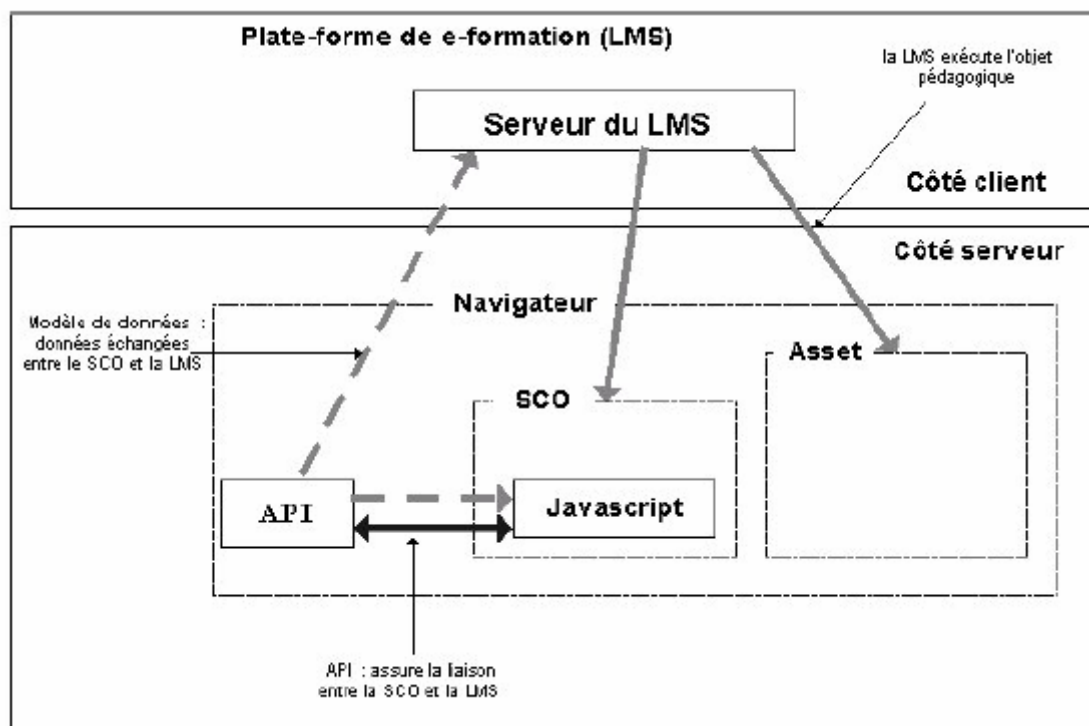


Figure 2 : L'environnement d'exécution de SCORM [Mbala, 03].

### **3. Le modèle de séquençement et de navigation :**

Il décrit comment le système interprète les règles de séquençement exprimées par un développeur et les événements de navigation qui peuvent être lancés par l'apprenant ou par le système.

### **3.3. Les normes IMS :**

Il existe également plusieurs spécifications du consortium IMS : IMS Content Package [IMS, 03a], IMS LIP et IMS Simple sequencing [IMS, 03d] et IMS LD.

#### **3.3.1 IMS simple sequencing :**

IMS SS permet de définir l'ordre de présentation des activités du e- Learning à l'apprenant et les conditions selon les quelles une ressource est sélectionnée, délivrée (delivered), ou ignorée lors de la présentation. Cette spécification ne prend en considération qu'un nombre limité des stratégies de séquençement communes. De ce fait, elle est nommée : simple IMS SS et utilise l'instruction de lecture du système d'ordinateur (computer system) pour déterminer l'activité suivante qui doit être présentée à l'apprenant. Ces instructions sont appelées : information de séquençement, qui sont divisées en deux catégories : le modèle de définition d'information et le modèle de trace d'information. Les informations définies sont ajoutées à un fichier XML par un auteur ou un outil. Actuellement, le fichier `imsmanifest.xml` utilisé dans IMS CP est le seul fichier utilisé pour cet objectif. Le résultat est donc un ou plusieurs arbres d'activités intégrés avec un ensemble d'instructions de séquençement. Les informations de trace sont obtenues lors de l'exécution des activités par l'apprenant.

Le processus qui détermine l'activité suivante est appelé : le comportement de séquençement.

Ce dernier est défini réellement par six processus : navigation, terminaison, rollup, sélection, séquençement et exécution. Le comportement de séquençement se déroule par la répétition de ces processus, formant ainsi une boucle répétitive. Cette boucle sera déclenchée quand l'apprenant clique sur un bouton ou un lien, pour passer à l'activité suivante. Cet événement fait référence au processus de navigation, qui génère deux événements : Le premier événement déclenche le processus de terminaison en assurant l'achèvement du traitement des informations concernant l'activité courante avant de passer à l'activité suivante. Le deuxième événement déclenche le processus du séquençement dans lequel une combinaison des règles de séquençement et des contraintes limitées permettant d'identifier l'activité suivante.

Le processus de séquençage donne la possibilité de contrôle au processus d'exécution (delivery process) qui initialise l'état du séquençage avant de présenter l'activité à l'apprenant. Si l'apprenant veut continuer, toute la boucle sera répétée.

Le processus du rollup est appelé dans le cas où l'activité est terminée et possède des activités précédentes et suivantes dans l'arbre des activités. Afin de déterminer le modèle de trace d'information du parent, l'état de l'activité terminée est combiné avec l'état des activités adjacentes.

Le processus de sélection est appelé quand l'activité possède des activités dites : filles (sous activité). Ce processus consiste donc à choisir un sous ensemble d'activités à partir de l'ensemble total d'activités.

### **3.3.2 IMS Learner information package :**

La spécification IMS LIP identifie une structure permettant l'échange d'informations concernant l'apprenant entre différents systèmes coopérants tels que : Les systèmes de gestion d'apprentissage, les systèmes des ressources humaines, les systèmes de gestion des données des apprenants, etc.

Les éléments contenus dans le package LIP sont généralement optionnels, l'utilisateur a le choix de les implémenter. On peut citer quelques exemples de ces éléments : l'identification pour identifier un individu en indiquant quelques éléments tels que : nom, prénom, adresse, etc. Les autres éléments sont : le but que l'apprenant vise à atteindre, les activités, les compétences, les intérêts, etc. Dans le but de satisfaire les différents besoins, LIP offre des possibilités d'exécution. En arrivant à un certain niveau de granularité, soit plus général ou plus détaillé, n'importe quel élément de la spécification peut être étendu. Des éléments extérieurs à la spécification, peuvent être aussi rajoutés au package.

### **3.3.3 IMS Content Packaging :**

IMS Content Packaging information model permet la définition des structures de données utilisées dans le but de fournir l'interopérabilité à l'Internet basé contenu avec : des outils de création de contenu, systèmes de gestion d'apprentissage (LMS), et environnement d'exécution. Cette spécification a pour objectif de définir un ensemble de structures standard pouvant être utilisés pour l'échange du contenu. Ces structures offrent la base pour les liaisons de données standards qui permettent aux développeurs et programmeurs de développer des matériaux d'apprentissage (instructional) interopérant via les outils auteurs et le LMS environnement d'exécution qui ont été créés indépendamment par différents développeurs.

- **Le modèle conceptuel d'IMS CP :**

Le diagramme conceptuel illustrant les différents composants d'IMSCP information model, est présenté par la figure suivante :

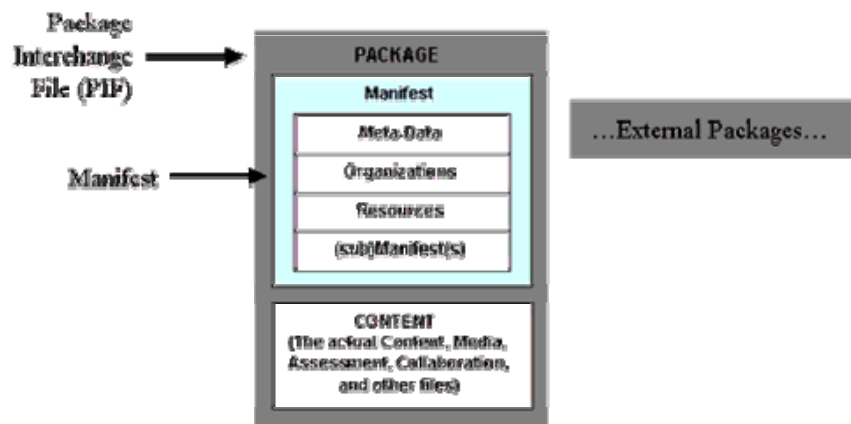


Figure 3 : IMS Content Packaging scope.

Cette figure montre que l'IMS package comporte deux éléments : un fichier xml décrivant l'organisation du contenu et les ressources dans un package, et les fichiers physiques écrits en xml. Le fichier xml est appelé imsmnifest.xml, car le contenu et l'organisation du cours sont définis dans le contexte des « manifests ». Une fois que le package soit incorporé dans un fichier pour le transporter, il est nommé : fichier d'échange de package (package interchange file). Ce dernier est un fichier unique compressé contenant un fichier manifest de haut niveau (top-level) appelé : imsmnifest.xml, ainsi que tous les autres fichiers physiques décrits par le manifest. Ce fichier est considéré comme un moyen de transport des informations structurées. Le package est un répertoire logique qui comporte un fichier xml et n'importe quel document xml de contrôle dont il fait référence, et les ressources physiques actuelles. Ces ressources peuvent être organisées en sous-répertoire.

### 3.3.4 IMS Learning Design :

IMS LD a été publié en 2003 par IMS/GLC (Instruction Manage System Global Learning Consortium) : Consortium pour l'apprentissage global avec les systèmes de gestion de normalisation. Il permet de décrire un processus pédagogique de façon formelle pour être déployé sur une plate forme de formation. Il supporte tout type de pédagogie et il n'est pas attaché à une classe particulière de plate forme. Nous donnons les principales caractéristiques de cette spécification en présentant son méta-modèle (pour plus de détail, sur cette spécification se reporter à (lafocade 2004)).

#### **4. Conclusion :**

La première section est consacrée à la présentation du domaine de l'EAO, d'où la machine qui est considérée comme un moyen d'enseignement. Ce domaine a connu beaucoup d'évolution comme par exemple l'introduction des systèmes experts pour jouer le rôle d'un enseignant (EIAO<sub>1</sub>). Après, une approche s'appuyant sur l'apprentissage par la découverte et l'exploration est apparue (EIAO<sub>2</sub>). Par la suite, les chercheurs ont tenté de combiner les deux approches précédentes : l'apprentissage par la découverte et l'apprentissage par résolution des problèmes (EIAO<sub>3</sub>). Finalement, on a présenté une autre approche qui est indépendante des mouvements technologiques et qui prend en compte la globalité du processus de l'enseignement (EIAO<sub>4</sub>).

Dans la deuxième section nous avons défini le domaine de la formation à distance (FAD) et son évolution vers la formation ouverte et à distance (FOAD), ainsi que les plates formes de formation.

Nous avons terminé par la troisième section, où on a défini le concept de normalisation et des langages de modélisation éducatifs dans le domaine du e-learning.

## **Chapitre 2 :**

# **Méthodes et outils pour la conception des situations d'apprentissage collaboratif**

## **1. Introduction**

Nous présentons dans ce chapitre le domaine lié à notre travail : les environnements informatique d'apprentissage humain (EIAH) et plus précisément l'apprentissage collaboratif pour 'Computer Supported Collaborative Learning (CSCL), la définition du langage de modélisation pédagogique que nous avons adopté dans notre travail : IMS LD et la présentation de quelques travaux réalisés dans le domaine du CSCL.

Aujourd'hui, le formalisme le plus approprié à la modélisation des scénarios collaboratifs tout en permettant la définition d'un espace de régulation des interactions sociales, est : Learning Design language (LDL) [Mar et al, 06], qui présente une version évoluée du modèle de participation, issu des travaux du [Fer et al, 05]. En effet, LDL fait appel à des notions de communauté, d'enceintes, des rôles, des règles et des positions, qui sont très proches de la théorie de l'activité. Cependant, LDL est un langage très pauvre en termes d'outils (éditeurs et infrastructures), ce qui nous a obligé de choisir l'IMSLD, malgré son non adéquation à l'apprentissage collaboratif. Comme solution à cette limite de IMSLD pour le CSCL nous proposons dans ce travail des extensions non pas de la spécification elle-même à priori de toute scénarisation mais une extension d'un scénario à posteriori par intégration d'un ensemble de propriétés sur la communication, l'interaction,...

## **2. Le Contexte général**

### **2.1. L'Environnement Informatique d'Apprentissage Humain (EIAH)**

La conception des systèmes informatiques n'était pas un processus systématisé, il est devenu aujourd'hui un processus technique basé sur des méthodes de Génie Logiciel.

Un EIAH (environnement informatique pour l'apprentissage humain) est défini comme « un environnement d'informatique conçu dans le but de favoriser l'apprentissage humain, c'est à dire la construction des connaissances chez un apprenant ». Cet environnement introduit des agents humains (élève, enseignant, apprenant, tuteur) et artificiels (agents informatiques) et leur offre des moyens d'interactions, localement ou à travers des réseaux informatique, et des contraintes d'accès à des ressources formatives (humaines et / ou médiatisées) » [Tch et Bak, 04].

Ce sigle dénote d'un point de vue diachronique, l'évolution du domaine de recherche, partant de l'enseignement programmé des débuts de l'usage des ordinateurs et de l'EAO (« Enseignement Assisté par Ordinateur »), passant à l'EIAO dans une première acception (« Enseignement Intelligemment Assisté par Ordinateur », d'où l'usage des techniques d'Intelligence Artificielle),

puis une seconde acception (« Environnement Interactif d'Apprentissage avec Ordinateur », d'où l'importance fondamentale de l'interactivité des systèmes). L'EIAH dénote aussi l'évolution vers la recherche de partenariat entre l'homme et la machine, en utilisant surtout les technologies de l'information et de la communication (TIC), et étend le champ d'étude à l'apprentissage humain dans toutes ses déclinaisons (enseignement, formation,...) [Tch, 02].

## **2.2. Le domaine du CSCL :**

Nous présentons dans cette section le domaine lié à notre recherche, l'apprentissage collaboratif assisté par ordinateur (ACAO) ou en anglais: 'Computer supported Collaborative Learning (CSCL)'. Ce domaine est basé sur l'usage des technologies pour l'apprentissage collaboratif en éducation. Le CSCL fait preuve d'une puissance importante dans les rapports pédagogiques entre l'enseignant et l'apprenant, l'apprenant et la tâche ou encore entre les apprenants. Timothy Koshman a proposé la définition suivante de ce champ :

'Un champ d'analyse concerné essentiellement d'une part par le sens et les pratiques de construction du sens lors d'une activité commune et d'autre part la manière dont les pratiques sont médiatisés à travers des artefacts qui ont été élaborés dans ce but. [Kos, 02].

Le CSCL regroupe plusieurs modèles méthodologiques, ce qui est conclu lors du colloque international CSCL 2002, à Boulder, Colorado. Des débats sur les travaux futurs dans ce domaine pour mettre un consensus sur le plan méthodologique ont eu lieu lors du colloque international CSCL 2005, à Taiwan et s'intitule « CSCL : les dix prochaines années ». [Zou, 06]. Il n'existe pas une définition consensuelle de l'apprentissage collaboratif mais nous avons retenu la définition suivante : 'L'apprentissage collaboratif est un processus d'interaction entre deux ou plusieurs apprenants pour apprendre ensemble'.

Beaucoup de chercheurs ne font pas la différence entre la "collaboration" et la "coopération", alors que d'autres disent que ces deux termes sont différents en ce qui concerne l'organisation du travail : en collaboration, tous les apprenants effectuent le même travail, alors qu'en coopération, il y a une division de l'activité, où chaque apprenant réalise une partie de cette dernière.

Cette distinction entre les deux termes est une particularité anglophone. Il existe des travaux anglophones réalisés par des auteurs comme Roy Pea [Roy, 94] et d'autres (George [Geo, 01], Germann et dellenbourg[Ger et Del, 99], Betbeder[Bet, 03]) favorisant l'utilisation du terme « collectif » pour indiquer un apprentissage 'collaboratif' ou 'coopératif'.



### **Situation d'apprentissage collaborative :**

Un groupe d'apprenant vivent une situation d'apprentissage collaboratif en effectuant un ensemble d'activités avec un même objectif. Il s'agit peut être d'un projet, étude de cas, résolution de problème....

### **Scénario :**

Un scénario pédagogique peut être décrit comme le déroulement du processus de l'apprentissage, ou encore la manière d'exécution des activités par les différents acteurs. Les activités à réaliser, les acteurs et leurs rôles et les scénarios sont déterminés lors de la phase de conception.

### **Acteurs :**

L'un des concepts qui doit être défini lors de la conception de situation d'apprentissage est l'acteur qui joue un certain rôle durant l'apprentissage. Dans notre travail on s'intéresse à deux types d'acteurs : l'enseignant entant que concepteur pédagogique et contrôleur, et l'apprenant.

#### ***L'enseignant :***

L'enseignant a le rôle d'un concepteur des situations d'apprentissage collaboratives et d'un contrôleur d'activité de l'apprenant. Le processus de conception consiste en premier lieu à déterminer l'objectif de l'apprentissage et le type du contenu pédagogique qui doit être utilisé durant l'apprentissage en répondant à un questionnaire. Ensuite, il doit donner une description détaillée de chaque activité pour la rendre plus compréhensible à l'apprenant. Finalement, il doit choisir les services indispensables pour la communication entre les apprenants. L'enseignant peut exercer un contrôle sur le travail des l'apprenants, il peut ainsi donner des informations ou des conseils, répondre à leurs questions....

#### ***L'apprenant :***

L'apprenant est la personne intéressé par l'apprentissage, et pour laquelle des situations d'apprentissage sont conçues, dans le but de construire un savoir. De ce fait, l'apprenant doit effectuer l'ensembles des activités déterminées par l'enseignant en utilisant des ressources et des services disponibles dans l'environnement d'apprentissage. Ces ressources peuvent être des fichiers ou des documents contenant des : cours, exercices, projets, problèmes à résoudre,....

Les services permettent aux apprenants d'interagir et de communiquer entre eux ainsi qu' avec l'enseignant.

### **3. L'ingénierie centrée sur les scénarios [Nod, 05] :**

Les propositions qui ont apparues à la fin des années 90, donnent plus d'importance au rôle de l'enseignant lors de la conception des environnements d'apprentissage. Un exemple de ces propositions, des standards centrés contenus comme le langage d'indexation des données LOM qui vise un enseignant créateur, prospecteur, référenceur et capable d'agréger des ressources en utilisant au titre d'exemple des modèles de mise en œuvre comme SCORM. Cependant, d'autres propositions visent la nécessité de prendre en considération l'activité de l'apprenant et utilisent donc la notion de scénario d'apprentissage, des approches basées sur la scénarisation pédagogique sont proposées par les langages de modélisation pédagogique (EML) : IMSLD , MISA , ....

#### **3.1. Apport des approches à base de scénarios :**

Nous avons choisi de limiter notre étude à l'EML centré scénario : IMSLD. Cette proposition d'EML s'appuie sur un méta modèle qui permet de décrire les caractéristiques de ce qu'on peut exprimer par les modèles valides, c-à-d respectant la relation « conforme à ». Ce méta-modèle définit les concepts du domaine (vocabulaire compréhensible par les concepteurs d'EIAH), ainsi que les relations qui peuvent être établies entre ces concepts.

L'IMSLD suit un processus qui consiste à effectuer une analyse des besoins et des buts d'apprentissage, ensuite extraire des spécifications pédagogiques à partir de cette analyse, puis créer un support de diffusion répondant à ces besoins d'apprentissage. La notion de scénario sert à guider la conception des contenus pédagogique et l'analyse des besoins, permettant de rendre l'EML plus approprié aux enseignants.

Le modèle conceptuel de l'IMS LD est basé sur l'activité : les acteurs (apprenant ou enseignant) réalisent des activités les conduisant à utiliser des ressources et des outils.

#### **4. Le langage de modélisation pédagogique :**

En ingénierie logicielle, l'activité de modélisation est primordiale comme elle l'est en EIAH. Les travaux de recherches ont montré qu'il n'est pas suffisant de mettre des ressources d'apprentissage à la disposition des plates formes de formation à distance. Il faut également fournir des outils permettant de les gérer. Dans ce sens, des langages de modélisation pédagogique ou éducative (educational modeling language) sont apparus.

La définition d'EML qu'on utilise dans ce document est utilisée comme définition de référence de nos jours :

*“An EML is a semantic information model and binding, describing the content and process within a “unit of learning “from a pedagogical perspective in order to support reuse and interoperability” [Raw, 02].*

Les EML proposent un modèle d'information décrivant les concepts, leurs relations, leur sémantique et le processus dans une unité d'apprentissage selon une perspective pédagogique donnée pour supporter la réutilisabilité et l'interopérabilité, ce qui implique que les modèles décrits avec les EML doivent être indépendants des diverses théories d'apprentissage : ils permettent aussi de décrire une correspondance vers une formalisation de ce modèle dans un langage compréhensible par la machine.

Une unité d'apprentissage (unit of Learning appelée aussi : unit of study ou unit of instrument) est la plus petite unité définissant un modèle d'apprentissage, elle peut être un cours, un module de formation,....

#### **4.1. IMS Learning Design**

Le méta-modèle d'information d'IMS-LD se décompose en trois composants principaux:

\_ Le modèle conceptuel : il représente le vocabulaire, les relations entre les concepts et les relations avec la spécification IMS Content Packaging.

\_ Le modèle d'information : il décrit les éléments d'IMS-LD sur trois niveaux :

- le niveau A décrit le vocabulaire au coeur de la spécification afin de concevoir des unités d'apprentissage supportant diverses pédagogies ;
- le niveau B ajoute au niveau A de nouveaux éléments permettant une personnalisation ainsi que des scénarios et des interactions plus élaborées ;
- le niveau C ajoute un nouveau mécanisme de notification permettant d'étendre la dynamique des scénarios construits.

La description de chacun des trois modèles d'informations s'appuie sur une version restrictive du modèle conceptuel précédent selon le niveau concerné.

\_ Le modèle de comportement : il décrit un ensemble de comportement d'exécution (runtime) que des systèmes délivrant les unités d'apprentissage devront implémenter.

Le concept au coeur de cette spécification pour la conception d'apprentissage est qu'une personne (person) joue des rôles (role) dans un processus d'apprentissage/enseignement (typiquement, un rôle d'apprenant learner ou de personnel staff). En jouant ce rôle la personne cherche à aboutir à des résultats (outcomes) en réalisant des activités d'apprentissage (Learning activity) ou de soutien (support activity) plus ou moins structurées (par le biais des structures d'activités activity-structure). Ces activités se réalisent dans le contexte d'un environnement (environment). Cet environnement est constitué d'objets d'apprentissage (Learning Object) et de services (services) qui seront utilisés pendant la réalisation des activités associées à cet environnement.

La méthode (method) permet de déterminer quel rôle assurera quelles activités à tel moment du processus. Le mécanisme de notification (notification) permet également de préciser certaines structures du processus de l'apprentissage ; ce mécanisme appartient au niveau C. La méthode est conçue pour atteindre des objectifs d'apprentissage (Learning objectives) qui correspondent à la spécification de résultats pour l'apprenant.

La méthode est décrite sur l'hypothèse de certains pré-requis (prerequisites) correspondants alors à la spécification des pré-requis d'entrée pour les apprenants.

La méthode est constituée d'une ou plusieurs pièces de théâtre (play) concurrentes (leur réalisation est en parallèle). Un play est constituée d'un ou de plusieurs actes (act) séquentiels. Chaque acte fait référence à un ou plusieurs rôle-part concurrents également ; ces rôle-part permettent d'associer un rôle avec une activité ou à une structure d'activité.

Au niveau B, une méthode peut contenir des conditions (condition) ou règles (par exemple du type If-Then-Else) qui permettent alors de raffiner la visibilité des activités et des environnements pour les personnes ou rôles. Ces conditions définissent alors des expressions booléennes sur les propriétés de ces personnes ou rôles. Une propriété peut être regroupée dans des property-groups qui sont un exemple de propriétés globales (global éléments) pouvant être définies. Les propriétés peuvent être de différents types, locales comme globales pour les personnes comme pour les rôles.

Une notification peut être générée par un résultat et peut ainsi créer une nouvelle activité disponible pour un rôle donné (la personne recevant la notification n'est pas forcément celle qui l'a émise). Par exemple, si un apprenant termine une activité (ce qui correspond au fait d'avoir atteint un résultat), alors un autre apprenant ou bien un enseignant peut se voir attribuer une nouvelle activité en conséquence. Ce mécanisme peut aussi être utilisé pour des conceptions d'apprentissage dans lesquelles la ressource d'une activité est dépendante du type de résultats obtenus à des activités précédentes (utiles pour des apprentissages coopératifs).

Deux rôles explicites sont spécifiés dans IMS-LD : learner et staff. Ces rôles peuvent être spécialisés en sous-rôles d'après la spécification d'IMS-LD mais cela reste à la charge du concepteur d'apprentissage.

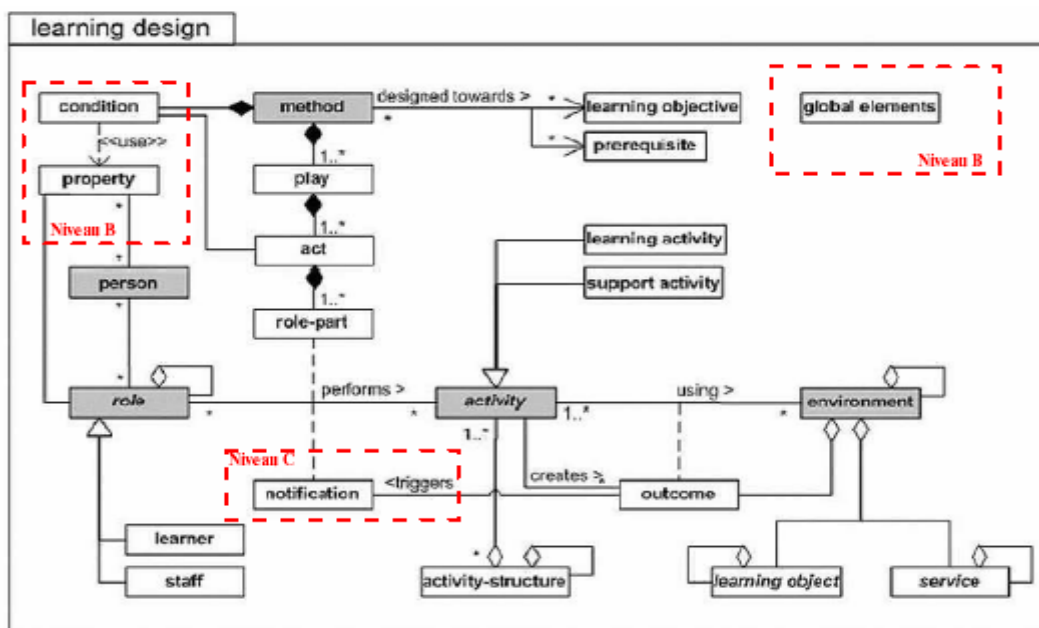


Figure 4 : Modèle conceptuel de IMS Learning Design.

Nous avons vu que les activités pouvaient être assemblées dans des structures. Celles-ci sont alors une agrégation d'un ensemble d'activités dans une unique structure ; les activités sont alors associables à un rôle grâce au rôle-part. Une structure peut modéliser une séquence d'activités, qui devront tous être complétés dans l'ordre fourni, ou bien une sélection d'activités pour laquelle le rôle associé décidera du nombre d'activités qu'il voudra réaliser. Les structures d'activités peuvent également faire référence à d'autres structures comme à d'autres unités d'apprentissage externes.

Les environnements peuvent contenir deux types de base :

- \_ Des objets d'apprentissage localisés (typiquement spécifiés avec une URL accompagnée parfois de métadonnées)
- \_ Des services génériques relatifs à des services concrets disponibles à l'exécution. Ces services n'ont pas d'URL associée à la conception. L'URL sera donnée quand la situation d'apprentissage conçue (Learning Design) sera instanciée à l'exécution.

Dans le but d'offrir aux développeurs des possibilités de traitement automatique par des systèmes informatisés, le consortium IMS fournit une implémentation du modèle d'information sous forme de schéma XML.

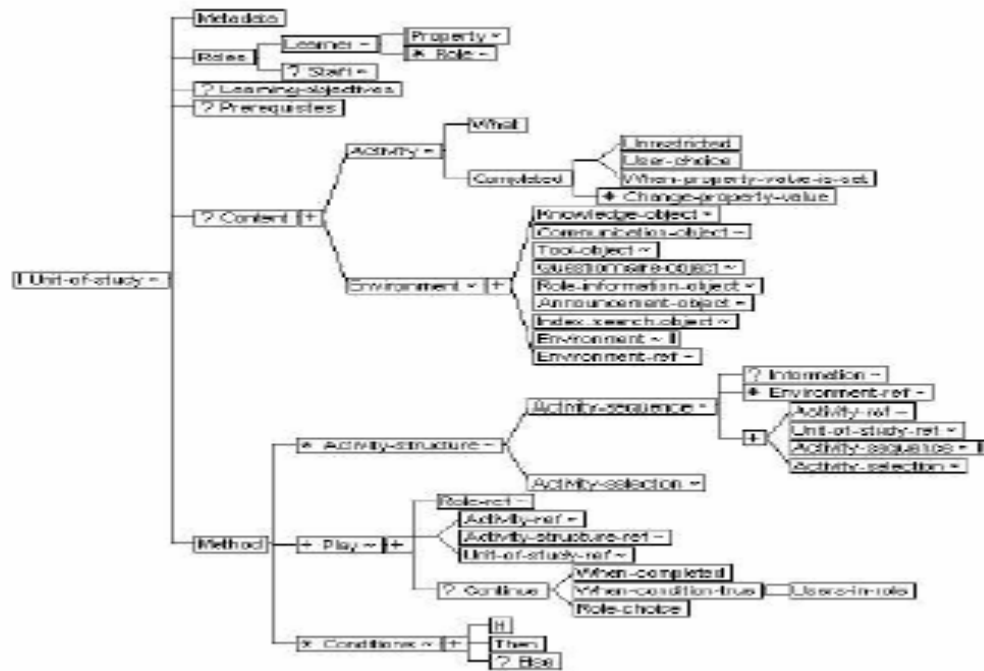


Figure 5 : Implémentation (binding) de la structure contenant les concepts d'IMS LD

Ce langage permet de modéliser des unités d'apprentissage grâce à des concepts pédagogiques.

La modélisation d'un plan de cours en IMSLD se fait par la définition des différents rôles, activités, environnement, méthodes, propriétés, conditions et notifications. Il est utilisé pour transformer les plans de cours en unités d'apprentissage (UoL) décrite de façon formelle et pouvant être exécutés avec l'éditeur IMS LD basé sur un moteur. Ces unités exécutable pourront être conçues dès le début en utilisant un éditeur tel que Copper Author ou Reload.

Comme les enseignants ne peuvent pas se familiariser avec le formalisme (XML) utilisé par le langage IMSLD, ces éditeurs sont donc nécessaires pour guider les enseignants dans le processus de conception des scénarios pédagogique. Dans ce qui suit nous définissons des exemples d'outils basés sur IMSLD.

Il existe trois catégories principales de ces outils : éditeurs du Learning design, outils d'exécution et les gestionnaires du learning design (players) [Gia, 06].

#### 4.2. Editeurs de 'Learning design' :

##### *L'éditeur Reload LD:*

Reload (Reusable e\_learning Object Authoring and Delivery) est un environnement développé par Philippe Beauvoir et Paul Charples, dans le but de concevoir et importer des packages conformes à IMSLD. Un ensemble de boutons correspondants aux concepts d'IMSLD : rôles, activités, propriétés, environnement, ainsi qu'un ensemble d'acts et role\_part constituants des

plays sont associés à chaque unités d'apprentissage. Les trois niveaux d'IMSLD : A, B et C, sont pris en compte par la deuxième version de Reload

#### ***CopperAuthor :***

Cet éditeur est développé par OUNL. Il offre une interface graphique permettant aux utilisateurs de développer et de valider des UoLs, en affichant le code XML résultant et en unifiant les unités d'apprentissages incomplètes. Il ne permet pas encore d'importer des UoLs.

#### **4.3. Les Players :**

En effet, Il existe plusieurs players, nous définissons parmi eux :

#### ***CopperCore :***

C'est un environnement d'exécution développé par OUNL dans le cadre du projet AFANET. Il sert à charger et valider des package IMSLD. Il permet ensuite de définir des exécutables, créer des utilisateurs et leur associer des rôles, cela se fait dans une fenêtre de commandes. L'unité d'apprentissage peut être ensuite exécutée par l'enseignant dans le player. Lorsque l'apprenant termine la réalisation d'une activité, elle est cochée et il peut passer à l'activité suivante. L'enseignant peut suivre le travail de l'apprenant.

#### ***EduBox :***

Ce player est développé par *Peros systems* et *OUNL*. Il est utilisé par ce dernier pour mettre ses cours en ligne. Des travaux en cours cherchent à intégrer EduBox dans le système Blackboard et développer une version respectant la spécification IMSLD. Ce player n'est pas téléchargeable.

#### ***L'unité d'apprentissage :***

L'utilisation d'une approche conceptuelle d'apprentissage nous permet d'organiser les unités d'apprentissage. Cette dernière assemble les ressources liées, les liens Web et plusieurs matériaux et services d'apprentissage dans un dossier compressé.

L'IMS LD est le plus approprié d'après Giacomini pour la conception d'un outil d'aide à la conception de sites Web éducatifs car il prend en considération plusieurs aspects tel que : les théories et modèles d'enseignement et d'apprentissage, les domaines d'enseignement, les rôles des acteurs, les interactions entre ces différents acteurs dans l'apprentissage et l'évolution dynamique d'un cours de formation.

Rasseneur [Rass, 04] a vu que pour la description d'une formation à distance pour supporter l'appropriation de la formation, le niveau A de la spécification IMSLD est la partie la plus intéressante, car à ce niveau, les modules d'enseignement sont structurées sous forme de modules, activités et sous-activités et les différents prérequis et les objectifs pédagogiques du module sont explicités. Cependant, d'après le même auteur, certains aspects de cette modélisation restent peu satisfaisants :

\_L'IMS LD ne permet pas la catégorisation des objectifs et des prérequis.

\_L'IMS LD ne différencie pas entre un objectif et une compétence.

\_Il ne permet pas de donner les dates de début et de fin des activités.

Dans le contexte des travaux de Caron [Car, 06] concernant l'implémentation de dispositif pédagogique, il considère que l'utilisation d'IMSLD peut être utile pour formaliser des scénarios pédagogique beaucoup plus flexibles et réutilisables, indépendamment de toute plate forme de formation. Cependant, il peut ne pas être capable de conduire correctement à une implémentation sur une plate forme de formation. Il est difficile de décrire les spécificités d'une plate forme donnée en IMSLD et de faire correspondre entre les fonctionnalités du scénario définies en IMSLD et les fonctionnalités réellement disponibles. Pour cela, l'implantation automatique du scénario pédagogique à partir du langage IMSLD pourrait être inutile. Caron a proposé donc d'adopter l'approche orientée modèles appliquée aux EIAH. Dans cette approche, l'IMS LD est considéré comme une étape intermédiaire entre la description du scénario en langage naturel et la traduction de son modèle par un langage qui prend en compte les spécifications de la plate forme de formation.

#### **4.4. Impact d'IMSLD :**

Il existe aujourd'hui, plus de 30 groupes et projets internationaux qui adoptent la spécification IMSLD comme : LORNET, L'ENT 'cartable électronique'..., ou des éléments qui sont directement attachés à elle. Les trois facteurs clés démontrant la pertinence et l'impact d'IMSLD, définis par Burgos et d'autres auteurs sont :

*Le premier facteur :*

L'existence de plusieurs applications qui sont basées sur IMS LD ou qui l'intègre dans leur modèle conceptuel (LAMS, ASKLDT, MOT Plus,...).

*Deuxième facteur :*

L'augmentation des communautés virtuelles autour d'IMS LD, l'intérêt de ces communautés réside dans le fait qu'elles permettent aux spécifications d'améliorer l'apprentissage en ligne en adaptant l'enseignement aux plates formes virtuelles.

*Troisième facteur :*

L'étendue géographique d'IMS LD, partant de son pays d'origine : les Etats-Unis, vers l'Europe, le Canada, l'Australie, etc.

#### **4.5. Les défis d'IMSLD :**

L'IMS LD comme les autres spécifications peut avoir quelques défis, parmi lesquels, on cite :

\_La difficulté d'importation de lots d'information 100% conforme IMSLD bien que l'échange et l'interopérabilité de ces lots d'information pour être exécutables par plusieurs systèmes



d'exécution, soient les buts les plus importants de toutes les spécifications de e-formation, ainsi que l'obtention de la plus grande flexibilité possible.

\_Les outils auteur et les éditeurs sont plus centrés sur les performances techniques au lieu d'être plus orientés vers les utilisateurs finaux qui sont : l'enseignant ou le concepteur pédagogique pour garantir une meilleure utilisation de ceux-ci.

\_la mise en place d'un environnement virtuel compatible a' 100% avec des UoL de IMS LD, tout en facilitant l'échange et l'interopérabilité avec les éditeurs d'UoL et les services interconnectés.

\_l'élaboration d'une liste la plus complète possible des modèles pédagogiques pour être réutilisés directement par l'enseignant, contenant des scénarios pédagogique modélisés auparavant et stockés dans une bibliothèque. L'enseignant peut donc économiser du temps et des efforts en utilisant ces scénarios avec une personnalisation et une adaptation à ses besoins.

## **5. Quelques approches du CSCL :**

Depuis quelques années, le domaine du CSCL a connu l'émergence de plusieurs travaux ayant pour objectif de proposer des approches et des outils pour la conception et la modélisation de situations d'apprentissage collaboratif. Cette section est consacrée pour la présentation de certaines de ces approches. Nous commencerons par l'approche proposée par Ferraris et son équipe [Fer et al, 05], puis nous définissons l'approche de Cyril Vachet et enfin, nous exposons la démarche de Hernandez Léo et son équipe.

### **5.1. Démarche de modélisation pour une chaîne éditoriale de scénarios [Fer et al ,05] :**

Le problème était la difficulté de planifier à priori des activités collaboratives pour une classe virtuelle, cela est dû aux environnements disponibles qui utilisent seulement des enchaînements de boîtes de dialogue et ne conservent pas une version explicite et réutilisable des actions réalisées pour cette mise en œuvre.

Les équipes SYSCOM de l'université de Savoie et ARCADE du CLIPS –IMAG de Grenoble ont décidé de regrouper leurs travaux dans le domaine de l'approche basé scénario et de modèle pour l'apprentissage collaboratif concrétisé dans l'ENT du « cartable électronique », dans le but d'offrir aux enseignants utilisant des modèles pédagogiques éventuellement différents, des moyens qui permettent la conception des scénarios collaboratifs pour leurs classes.

Cette démarche sert à transformer un scénario pédagogique collaboratif exprimé librement par des enseignants en une activité exécutable au sein de l'ENT, en effectuant un traitement appelé : « chaîne éditoriale ». Le Modèle –Driven – Architecture (MDA) permet de spécifier facilement des systèmes interopérables en se basant sur des modèles formels. Elle a été adoptée pour définir

le schéma de transformation, de manière à situer la démarche proposée au niveau des modèles plutôt qu'au niveau des composants.

### **5.1.1. Principe séparé du PIM et génération du PSM :**

La première étape de l'approche MDA est d'élaborer un modèle d'application indépendant de son plate forme d'exécution (PIM : Platform Independent Model). Puis, la description de la plate forme par le modèle : PDM (Platform Description Model). La transformation de ce modèle prenant en compte la plate forme PDM donne comme résultat un « PSM : Platform Specific Model » qui se traduit par un code exécutable sur la plate forme cible.

L'idée était donc de traduire un scénario décrit en langage naturel dans le méta-modèle IMSLD ensuite d'enrichir ce dernier par l'introduction du méta-modèle de participation. Le résultat est donc « un modèle générique de scénario indépendant de toute plate forme (PIM). L'opérationnalisation de ce modèle doit prendre en considération les deux modèles qui composent le modèle PDM, qui sont : le modèle de l'ENT cible et le modèle qui sert à définir l'ordonnancement des activités du scénario (par exemple : Workflow). L'adaptation du modèle générique à ces modèles conduit au : « modèle de scénario opérationnalisable (PSM) ». Enfin, l'affectation des valeurs aux différentes propriétés de ce modèle, telles que : ressources, rôles,...dans le but de l'instancier. Cette équipe a envisagé de concevoir des outils s'appuyant sur cette approche. L'objectif de ces outils est de permettre aux enseignants d'éditer des scénarios pour construire une bibliothèque de modèles et d'instances de scénarios partageables, modifiables et réutilisables.

### **5.1.2. Formalisation avec IMSLD :**

Les auteurs disposant initialement de textes descriptifs qui présentent un jeu de rôles pour l'analyse d'une œuvre littéraire. Ces textes permettent d'identifier plusieurs informations telles que : le contexte, l'objectif et les différentes étapes du scénario pédagogique. La première tâche est l'extraction des différents rôles d'apprenants et d'encadrement, les activités (au sens IMSLD), en se basant sur les ressources requises et le déroulement chronologique du scénario. Puis le groupement de certaines activités structurées afin de garantir les synchronisations ainsi que la description des propriétés nécessaires au contrôle de l'exécution. La dernière tâche consiste à présenter le déroulement du scénario par un diagramme d'activité UML.

A cause des insuffisances d'IMSLD comme l'absence d'un mécanisme de description des interactions des membres d'un groupe dans une activité collaborative, ces auteurs ont été obligés d'adopter « le modèle de participation », conçu pour la description des activités conjointes.

### **5.1.3. Formalisation de scénarios d'apprentissage collaboratif selon le modèle de participation :**

Il s'agit ici de définir un espace de régulation, qui est un niveau « méta », utilisé pour permettre une construction conjointe de l'activité du groupe et son évolution dans le temps, au cours de l'activité elle-même. Cet espace comporte : la constitution et vie des groupes, définition des droits et devoirs au sein du groupe, mise en œuvre des règles de fonctionnement, et tout ce qui concerne l'organisation du groupe. Le modèle de participation a pour objectif de structurer l'espace de régulation des membres des groupes considérés comme des participants à une activité conjointe doivent spécifier les conditions de leurs engagements. Il permet de décrire formellement les interactions entre des participants à une activité, en utilisant la notion du scénario de modèle de participation (scénario MP).

Le modèle de participation est un modèle basé sur la notion des « enceintes », qui signifient : espaces de groupes. L'espace de travail d'un groupe est délimité par une enceinte. Les participant à l'activité où les acteurs sont mis en scène par l'enceinte, possèdent des caractéristiques tel que : leurs statut, leurs rôles et leurs position dans des groupes où sont définis des scénarios (histoires qui décrivent le cadre des interactions) construit par les acteurs eux même. La personne qui est chargée de la gestion de ces scénarios est : le médiateur qui met en œuvre les scénarios, vérifie leurs application, observer la dynamique du groupe, et éventuellement intervenir au sein de l'enceinte, les acteurs ont besoin de manipuler des objets via des outils spécifiques. Deux types de rôles sont identifiés : les rôles « thématiques » signifient une 'mission première' associée à un acteur (par exemple : directeur, chef de comité,...). Les rôles « actanciels » correspondent à l'action effectuée (exemple : je dépose un document dans l'enceinte, je suis « celui-qui-dépose »). Dans l'enceinte, les scénarios définissent les règles (règles du jeu de la coopération) en vigueur : droits, devoirs, obligations, recommandations,....

### **5.2. Méthode de scénarisation des activités collaboratives prenant en compte la mobilité de l'utilisateur [Cyr, 06] :**

L'intégration des dispositifs mobiles dans l'apprentissage fait l'objet de plusieurs travaux de recherches. Un dispositif mobile tel que : téléphone portable, est devenu très présents avec les apprenants dans les salles, ce qui signifie qu'ils peuvent l'utiliser à des fin d'apprentissage.

L'objectif de Cyril Vachet est d'élaborer une méthode de scénarisation des activités collaboratives prenant explicitement en compte la mobilité de l'utilisateur et de développer une infrastructure d'exécution pour opérationnaliser de tels scénarios sur des dispositifs tel que : le téléphone portable. Une méthode, selon cet auteur comporte : un langage de scénario (un ou des formalismes), une démarche explicite de scénarisation, des outils permettant de valider cette

démarche et une infrastructure permettant de fournir un support d'exécution de scénario. Jusque là, la mobilité de l'utilisateur est totalement absente.

Pour y remédier, l'auteur a choisi d'adopter l'Ingénierie Dirigée par les Modèles (IDM), qui utilise les bonnes pratiques issues du domaine des systèmes d'informations, car d'après cet auteur, les préoccupations de ce domaine (modélisation des systèmes et garantie de leur opérationnalisation) sont très proches de celles de la scénarisation des activités pédagogiques et collaboratives.

### **5.2.1. Scénarisation pédagogique collaborative et mobilité :**

Le langage le plus utilisé pour modéliser et formaliser les scénarios pédagogiques est : IMSLD, et l'environnement d'exécution (infrastructure) le plus adopté : Coppercore pour exécuter ces scénarios.

D'après les recherches menées dans le domaine des collecticiels, la collaboration est très importante pour l'évolution des apprenants. L'organisation et la coordination du groupe dans un espace de travail collaboratif (régulation [Fer et al, 05]) sont essentielles pour l'expression d'une dimension sociale et interactionnelle, qui véhicule un sens supplémentaire dans la mobilité de l'utilisateur avec la notion d'espace de travail physique (environnement). Selon Renevier [Ren, 04], un utilisateur mobile possède les caractéristiques suivantes : son « nomadisme » pour indiquer un utilisateur équipé d'un dispositif portatif et qui se déplace immersivement dans son environnement, « l'ubiquité » qui implique la réalisation des activités n'importe où et à n'importe quel moment, et « la sensibilité au contexte physique » qui est essentielle dans un cadre mobile, ce qui rend l'action située physiquement et géographiquement.

### **5.2.2. Le processus de scénarisation :**

La scénarisation prend en compte plusieurs préoccupations : pédagogiques, sociales, interactionnelles et techniques. L'IDM a pour objectif de rassembler ces préoccupations hétérogènes (cadre agrégateur). Elle se base sur l'opérationnalisation des modèles par raffinement (transformation) afin de les rendre compréhensibles par une machine. Pour cela, Cyril Vachet a adopté la notion d'aspect pour exprimer les préoccupations mentionnées précédemment. Un aspect est un point de vue différent des autres sur le système à modéliser. La définition des liens entre ces aspects peuvent être définis par des langages spécifiques du domaine (DSL). L'avantage de l'approche par aspect est de simplifier le problème en le décomposant en sous – problème. L'IDM offre les moyens qui servent à identifier les outils et les langages pour lier ces aspects entre eux et gérer les niveaux d'abstraction (raffinement des modèles). Les aspects qui peuvent être extraits ici sont : mobilité, interaction homme-machine,

activité pédagogique, collaboration, etc. Dans ce travail, il s'agit de mettre en relation des préoccupations d'ordre pédagogique, social et informatique.

#### **5.2.2.1. La définition des aspects :**

La modélisation par aspect constitue une étape préliminaire pour mieux cerner les différentes préoccupations des scénarios. Ces aspects sont :

-**les acteurs** : ce sont les personnes jouant des rôles différents pour réaliser les activités pédagogiques.

-**les activités** : il existe trois types : les processus, fournir des résultats observables, les interactions et les opérations (unité de traitement élémentaire dispensée par un service).

-**les ressources** : présentent les différents objets pédagogiques et les services utilisés pendant l'apprentissage.

#### **5.2.2.2. Proposition du processus de scénarisation :**

La démarche suivie est proche de celle de conception d'un système d'information. La modélisation par aspect conduit l'auteur à définir un langage inspiré d'un processus RUP constitué de deux phases afin de formaliser ces aspects. Nous définissons par la suite les différentes phases de cette démarche.

- **La phase d'étude organisationnelle :**

L'objectif de cette phase est d'identifier les aspects structurels et comportementaux de l'organisation «classe» en prenant en considération les contraintes physiques liées à la mobilité (contrainte sur les objets, leurs localisations, leurs relations et les zones). Il s'agit de définir un processus générique par l'enseignant indépendamment de son opérationnalisation.

- **La phase opérationnalisation :**

Cette phase sert à identifier les moyens indispensables pour informatiser une sous partie du processus définis dans la phase précédente.

- **La phase d'instanciation du modèle de scénarios :**

L'instanciation d'un modèle de scénario concerne :

-La mise en relation des utilisateurs réels avec l'acteur identifié dans le scénario.

-L'association des services requis lors de l'opérationnalisation aux services fournis par l'infrastructure d'exécution.

-L'initialisation de la position des différents objets physiques prédéfinis dans le modèle de l'environnement (prise en compte de la mobilité).

### **5.3. Une approche basée sur les CLFPs pour la conception des scénarios d'apprentissage collaboratif [Her et al, 05] :**

Vu l'importance des interactions sociales dans le processus d'apprentissage, les développeurs cherchent à créer des applications réutilisables et flexibles qui supportent les activités collaboratives réalisées par un ensemble d'acteurs. Plusieurs approches ont été adoptées pour répondre à ces besoins : l'approche basée composant, l'approche orientée services,....Hernandez Léo et son équipe signalent que les travaux réalisés pour atteindre un but de réutilisation et de flexibilité dépendent d'une identification spécifiques des outils utilisés (composants, services,...) compréhensibles seulement par les informaticiens. Dans le domaine du CSCL, la distance entre les abstraction utilisées par les experts dans l'apprentissage collaboratif et celles définies par les informaticiens est minimisée par l'introduction de l'approche intermédiaire : Collaborative Learning Flow Patterns(CLFPs). Cette approche permet de définir des outils du CSCL réutilisables. L'IMSLD est le formalisme sélectionné par ces auteurs pour décrire de manière formelle ces CLFPs.

#### **5.3.1. Collaborative Learning Flow Patterns :**

Un CLFP sert à décrire les techniques de collaboration qui permettent de structurer les interactions entre les participants. Ils représentent les meilleures pratiques pour définir le flux des différents types d'activités d'apprentissage. Ces types sont le plus souvent : collaboratifs. Un CLFP est une structure qui comporte la description des éléments suivants : problème, contexte, solution, acteurs, type des groupes, types de tâches, types d'informations,....L'avantage le plus important des CLFPs est de faciliter la communication entre des personnes possédant des perspectives différentes.

Le processus de collaboration est décrit en premier lieu, en utilisant les CLFPs, qui introduisent à leur tour, les modèles de collaboration : le CLFP qui utilise jigsaw est appelé : le CLFP jigsaw, celui qui adopte le modèle pyramid est : le CLFP pyramid,....ensuite, ces CLFPs sont modélisés par l'usage d'IMSLD.

#### **5.3.2. L'extension d'IMSLD pour CSCL :**

La collaboration est l'élément primordial du domaine de CSCL. L'IMSLD confirme qu'il supporte ce type d'activités, mais, il ne permet pas de décrire la manière de collaborer entre les individus pendant chaque activité d'apprentissage. La réalisation d'une activité collaborative se fait via des outils définis dans l'environnement supportant l'apprentissage collaboratif. A cette fin, cette équipe a proposé quelques services de base : 'Forum de discussion' et 'e-mail'. D'autres

services sont nécessaires pour définir quelques propriétés liées à l'activité de collaboration, l'IMSLD ne permet pas de les offrir. Ces propriétés peuvent être :

- **Type of awareness** : permet de fournir quelques informations sur le travail des autres acteurs comme par exemple : *l'identity* : qui participe, *l'action* : qu'est-ce qu'il fait ? *location* : où travaille-t-il, ...
- **Floor control policy** : sert à guider les actions des apprenants. Cette propriété indique si ces actions sont : fixées (prédéfinies par l'enseignant), dynamique (ne sont pas prédéfinies), ou aucune des deux.
- **Type of communication skills** : peut être : écrite, orale,....
- **Roles** : pour identifier les acteurs qui utilisent le même service.
- **Type of interaction** : plusieurs types d'interactions sont distingués : direct, indirect, médiatisé par des objets partageables, etc.

Dans ce contexte, ces auteurs ont proposé une extension pour l'IMSLD qui concerne la définition d'un ensemble de services appelé : *groupservice*, leurs principales caractéristiques sont celles mentionnées ci-dessus. La figure suivante présente les différents éléments du *groupservice* :

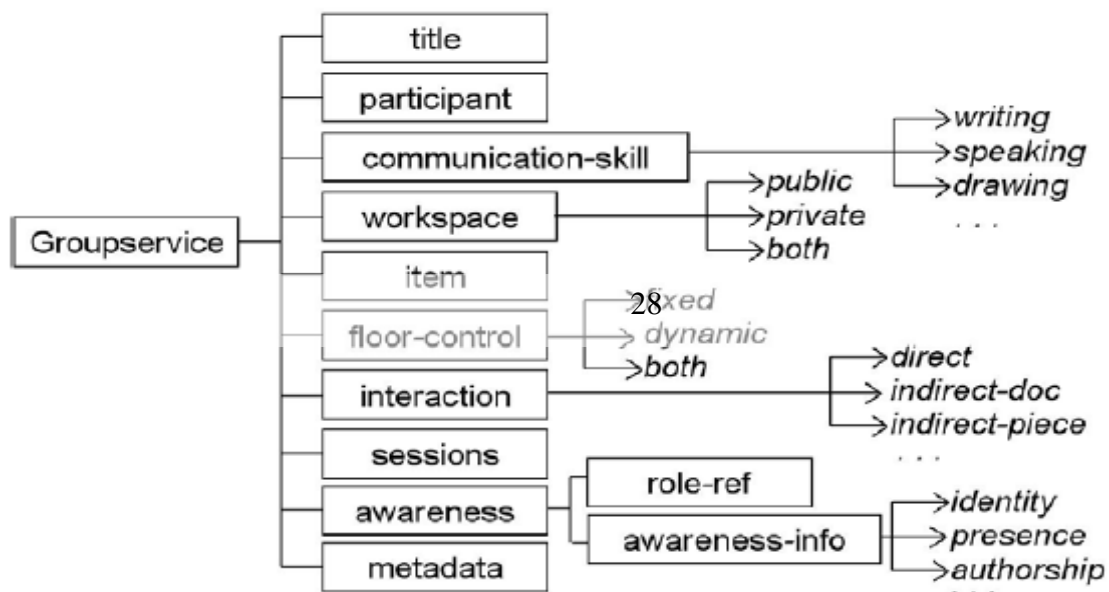


Figure 6 : schéma de l'extension proposée pour l'IMSLD (*groupservice*).

### 5.3.3. Vers une création d'une unité d'apprentissage basée CLFPs :

Les éléments décrits par un CLFP peuvent être facilement traduits en concepts décrits par IMSLD. Les types des tâches peuvent être transformés en activités d'apprentissage et de support. Les acteurs et les types des groupes en rôles, les types d'informations en propriété ou objets pédagogiques utilisés lors de l'apprentissage,....

Le processus de création d'une unité d'apprentissage comporte trois étapes : la première étape est la formalisation des CLFPs en utilisant le formalisme IMS LD, cela se traduit par la génération d'un document XML (manifest) permettant de décrire ces CLFPs (scénario d'apprentissage). Cette étape inclut la transformation des éléments du CLFP en des éléments IMS LD, précédemment mentionnés.

La deuxième étape consiste à particulariser le document précédent en décrivant en détail tout les éléments d'un learning design complet. La dernière étape concerne la définition des ressources qui peuvent être utilisées, dans content package avec le manifest, une unité d'apprentissage est alors obtenue.

#### **5.3.4. Mise en oeuvre:**

Ces auteurs ont implémenté deux outils auteurs basés sur l'approche proposée. Le premier outil est : *collage*, qui implémente les trois étapes proposées. Cet outil sert de guide aux concepteurs pour créer leurs propres situations d'apprentissage collaboratives en utilisant les CLFPs formalisés en IMS LD. Cette équipe a implémenté un deuxième outil : Gridcole, d'où l'introduction de l'ensemble de services proposés: *groupservice*.

### **6. IMS LD et CSCL [Her, 05]:**

On peut décrire des scénarios d'apprentissage avec IMSLD, contenant des rôles, chacun peut être joué par plusieurs personnes (un groupe). Le processus de collaboration peut être défini par l'association de plusieurs individus et/ou de plusieurs rôles à une même activité. Le domaine du CSCL, au sein duquel se déroule notre travail, se caractérise principalement par les mécanismes qui supportent la collaboration, alors que l'IMSLD n'est pas capable de décrire les différentes interactions entre les membres d'un groupe lorsqu'ils réalisent une activité d'apprentissage. Ces interactions peuvent être : discussion, argumentation, échange des idées, etc. Ainsi, quand un groupe d'individus travaillent ensemble, ils ont besoin des outils leurs permettant de communiquer. De ce fait, on était obligé d'adopter des modèles, qui ont pour vocation de spécifier la manière de collaborer entre les individus. Ces modèles sont appelés : modèles de collaboration. En ce qui concerne les outils de communication, on a implémenté un service « chat », pour permettre aux apprenants de discuter,....



## **7. Conclusion :**

Après une étude approfondie, nous avons présenté dans ce chapitre une définition du domaine de recherche : l'EIAH et le sous domaine CSCI ainsi qu'une description détaillée de la spécification IMSLD que nous avons utilisé dans notre travail, et nous avons terminé par la présentation de quelques approches proposées pour la modélisation des scénarios d'apprentissage collaboratif. L'objectif commun entre ces méthodes est de pallier au problème de la mise en charge des interactions entre les apprenants. Ferraris et son équipe [Fer et al, 05] ont proposé l'enrichissement d'IMSLD par des concepts du modèle de participation afin de capturer la richesse des interactions inhérentes aux activités collaboratives. La deuxième méthode proposée par Cyril Vachet [Cyr ,06], est basée sur la décomposition par aspects, d'où la collaboration est considérée comme l'un des aspects que doivent être pris en compte par cette méthode. Dans la dernière approche Hernandez Léo et son équipe [Her, 05], on constate l'introduction des CLFPs basés sur les modèles de collaboration (Pyramid, Jigsaw, ...) pour décrire les différentes interactions.

Nous avons abordé le même problème que celui approché par ces auteurs. Pour y remédier, nous avons adopté les modèles de collaboration utilisés par Hernandez Léo.

## **Chapitre 3**

# **Les EIAH et les techniques de Génie Logiciel**

## **1. Introduction :**

Nous présentons dans ce chapitre, quelques approches définies dans le domaine du génie logiciel en commençant par les apports de l'utilisation de l'approche basée composant et l'approche basée patterns dans l'EIAH, puis nous définissons la notion des composants logiciels, le modèle des composants : Fractal, qu'on a introduit dans notre travail, ainsi qu'une petite discussion sur les composants logiciels et les services web. Par la suite, nous décrivons la notion des patterns et nous discutons leur relation avec l'IMSLD.

## **2. Les apports des approches et des techniques du Génie Logiciel (GL) aux EIAH**

Depuis plusieurs années, de nombreux travaux de recherche ont pour objectif le développement et la conception des EIAH permettant d'offrir des fonctionnalités aux différents acteurs : les apprenants, les enseignants,... Ces environnements sont le plus souvent incomplets et non opérationnelles, c'est-à-dire n'offrant pas toutes les fonctionnalités indispensables à leurs usage en situation réelle. De ce fait, les chercheurs sont obligés de développer d'autres programmes complémentaires pour tester et faire fonctionner ces environnements en situation réelle. Ces programmes peuvent être disponibles ailleurs et il sera plus utile de les réutiliser au lieu de les re-développer [Reb, 05]. Pour cela, les chercheurs adoptent des approches du domaine de génie logiciel telles que : l'approche basée Composant et l'approche orientée patron (en anglais : pattern), pour développer des programmes facilement réutilisables et adaptables.

### ***2.1. L'apport de l'approche basée composant aux EIAH***

L'approche basée composant est fondée sur des techniques et des langages de construction d'application qui intègrent d'une manière homogène des entités logicielles provenant de diverses sources.

L'utilisation de cette approche dans le domaine des EIAH est nécessaire pour construire un nouvel environnement, en intégrant des programmes préexistants. La réutilisation permet de :

- Réduire le temps et le coût de développement des applications logicielles,
- Améliorer la qualité des applications programmées et testées auparavant.

Cette approche permet aussi de mettre en œuvre des propriétés d'interopérabilité et de coopération, qui permettent de construire des activités pédagogiques à partir des fonctionnalités complémentaires provenant de différents systèmes dans un environnement donné [Mac – Ros, 03].

## **2.2. L'apport de l'approche orientée patrons aux EIAH :**

L'idée d'utiliser les patrons dans le e-learning afin d'améliorer la qualité de l'enseignement et de la formation vient du domaine d'architecture (patrons d'architecture) L'approche patron, comme l'approche basée composant, présente plusieurs avantages [Bag & Rét ,03] :

- La diminution du coût et du temps de conception et de développement.
- L'augmentation de la qualité des logiciels.
- L'augmentation de l'efficacité d'apprentissage (qualité pédagogique).

Cette approche permet aux développeurs des logiciels, la réutilisation des solutions aux problèmes récurrents de conception. Dans l'EIAH, et plus précisément dans le domaine du CSCL, les patrons servent à réduire la distance conceptuelle entre les abstractions adoptées par les experts dans le domaine de l'apprentissage collaboratif et celles utilisées par les informaticiens, ce qui aide à obtenir des outils de CSCL réutilisables [Her , 05].

## **3. L'approche basée composant et les EIAH**

La notion de composant est apparue pour la première fois en 1968 dans le Workshop NATO Software engineering, où elle a été utilisée par dong McILLroy lors de la présentation de son article intitulé « Mass produced Software component). Ce visionnaire prévoyait que la résolution de la crise logicielle se faisait par la production massive des composants logiciels [Szy, 02]. Cette approche est apparue pour pallier les défauts de l'approche objet et dépasser ses limites et viser des notions non prévues initialement. Cette méthode dénommée CBSE (Component Based Software Engineering) consiste à concevoir une application en assemblant des composants préfabriqués. Elle a pour finalité de favoriser la réutilisation.

### **3.1. Définition du composant :**

Dans cette section, nous présentons quelques définitions qui ont été données au concept de composant logiciel. Selon Allen [All & Fro, 98] : « *un composant est une unité exécutable ayant la forme d'une boîte noire encapsulant les services qu'elle fournit. Ces services ne sont accessibles que par les interfaces publiées correspondantes et ce à travers un standard d'interaction* ».

Brown [BRO &WAL, 98] a défini le composant comme suit : *Un composant est un fragment utile d'un système logiciel pouvant s'assembler avec d'autres fragments pour former des pièces plus grandes d'applications complètes* ».

La définition retenue par Oubahssi [Oub, 05], à partir de ces différentes définitions :

« Un composant peut être défini comme une « brique logiciel », avec une interface bien spécifiée (le service rendu) ». Il peut être manipulé sans connaissance sur son contenu, importé, détruit, activé, configuré, adapté, déployé dans un environnement d'exécution, et un sujet de composition par un tiers pour concevoir des applications logicielles. Les composants réutilisables, dans l'ingénierie des systèmes d'information permettent de fournir des solutions conceptuelles et /ou logicielles testées, acceptée et adaptables aux problèmes récurrents.

Un composant peut être caractérisé par plusieurs critères tels que :

- *Type de connaissance* : Produit ou processus;
- *Couverture* : Générale (horizontale), métier (verticale), entreprise;
- *Portée* : Etapes de l'ingénierie;
- *Nature de la solution* : Conceptuelle ou logicielle;
- *Technique de réutilisation* : Spécialisation, composition, instanciation, duplication, etc.
- *Ouverture*: Boîte noire (composants binaires distribués sans code source), blanche (composants livrés en Open Source transformables et adaptables aux besoins spécifiques de l'application) ou en verre (forme mixte entre boîte noire et boîte blanche) ;
- *Granularité* : En nombre de classes, de modules, etc.

### **3.2. Classification des composants et EIAH :**

Les composants logiciels sont classifiés en catégories selon le rôle qu'ils jouent dans l'EIAH, afin de faciliter leur recherche par les futurs usagers.

#### **3.2.1. Les composants logiciels pédagogiques :**

*Définition 1* : « Un composant logiciel pédagogique (CLP) est un composant "métier" participant directement au processus d'apprentissage humain ».

« Un composant métier » signifie qu'il comporte des aspects fonctionnels. Cette notion est conforme à l'approche dirigée par les modèles qui sépare les aspects métier des aspects techniques lors de la conception d'une application [Bla, 05]. Des exemples de ce type de composant sont les gestionnaires (ou players) de scénarios pédagogiques, les simulateurs ou les résolveurs de problèmes à vocation pédagogique, les outils d'évaluation de l'apprenant.

#### **3.2.2. Les composants logiciels de service :**

*Définition 2* : « Un composant logiciel de service (CLS) est un composant apportant une plus-value fonctionnelle aux EIAH, mais non spécifique aux EIAH ».

Ce type de composant optionnel c'est à dire n'est pas indispensable pour assurer la vocation pédagogique de l'EIAH. Les services fournis par ce composant permettent d'atteindre un certain niveau de qualité en assurant par exemple une meilleure ergonomie et une flexibilité pour l'utilisateur. Il peut être utilisé dans plusieurs domaines d'application. Des exemples de

composant logiciel de service : des synthétiseurs vocaux, des adaptateurs d'interfaces ou des outils de communication (messagerie, chat, forum, etc.).

### 3.2.3.-Les composants logiciels techniques :

*Définition 3* : « Un composant logiciel technique (CLT) est un composant fournissant les mécanismes de base nécessaires à la constitution de l'infrastructure des EIAH pour garantir le bon fonctionnement des CLP et des CLS. C'est un composant n'apportant pas de fonctionnalités pédagogiques aux EIAH. Il est susceptible d'être utilisé dans tous les domaines ».

Ces composants seront adoptés pour la construction des plates formes EIAH, comme ils peuvent jouer le rôle d'un intermédiaire entre des CLP, et des CLS. C'est le cas, par exemple, des composants de sauvegarde des données (de persistance), d'impression de transformation d'un texte XML en pdf , etc .

### 3.2.4. Les composants logiciels de fabrication :

Cette classe de composant (CLF) regroupe des constituants des outils auteurs servant à créer des scénarios pédagogiques, des images animées, des QCM, etc.

Le développement de systèmes complexes à base de composant suscite aujourd'hui beaucoup d'intérêts aussi bien dans le monde de la recherche que dans celui de l'industrie. Ces dernières années ont vu émerger différents modèles de composant telle que les **EJBs** ou **CCM**. Mais il s'est avéré que leurs spécifications sont complexes et leurs mises en œuvre sont plus complexes. De ce fait, la communauté académique a proposé un autre modèle : le **modèle de composant Fractal**, dont l'objectif principal est de faciliter la spécification, la construction, le déploiement, et la reconfiguration d'application à base de composants.

## 3.3. Le Modèle Fractal:

### 3.3.1. Principe:

Le modèle de composant **Fractal** est un modèle général de composant logiciel, modulaire et extensible qui permet d'implémenter, de déployer, et de gérer des systèmes et des applications complexes sur différentes plates-formes avec divers langages de programmation [BCS, 04]. Ce modèle est spécifié conjointement par **France Telecom R&D** et **l'INRIA Rhone-Alpes** dans le cadre du consortium **Objectweb**. Ce projet a vu le jour à partir de **janvier 2002**. Ce modèle est défini à travers cinq concepts : composant, interface, liaison et contrôleur. Il est caractérisé par :

- **La récursion:**

Fractal est un modèle intrinsèquement récursif. Un composant Fractal peut être composé d'un ou plusieurs composants. Ces derniers sont auto similaires (d'où le nom de "Fractal") : quelle que soit l'échelle à laquelle on les observe, ils apparaissent toujours comme étant la composition d'un

contrôleur et d'un contenu. Le contrôleur portant les interfaces qui seules permettent d'interagir avec les composants.

○ **La réflexion:**

Fractal est un modèle intrinsèquement réflexif. Les contrôleurs permettent une introspection totale de la structure des composants, c'est à dire, une réification de la topologie des assemblages de composants et permettent la navigation au sein de ces assemblages. Les contrôleurs exercent également un contrôle arbitraire sur leur contenu, c'est-à-dire, exhibent des capacités d'intercession arbitraire.

○ **Le partage:**

Un composant peut appartenir au contenu de plusieurs autres composants englobants.

Fractal est un modèle développé pour répondre aux limitations des modèles actuels de composants logiciels, il est caractérisé par [BCL<sup>+</sup> 04, Kra 03]:

- Un fondement sur une base mathématique rigoureuse (réflexion et récursivité d'où la nomination du modèle: Fractal).
- Un partage de composants.
- Une composition (assemblage) hiérarchique de composants.
- Une possibilité d'adaptation et de reconfiguration.
- Une présence d'outils de description globale

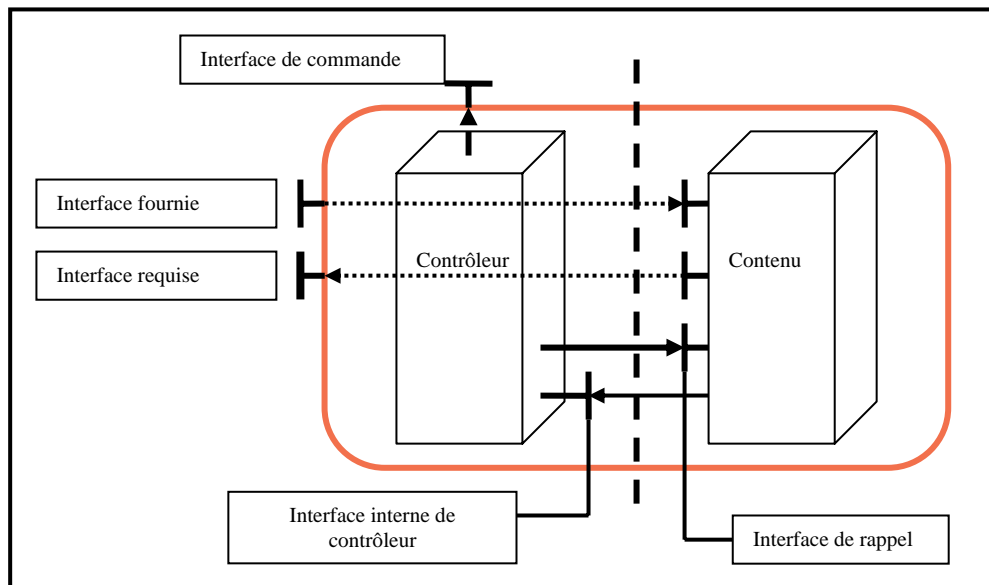


Figure 7: Schéma descriptif d'un composant

Un composant Fractal est une entité logicielle exécutable qui est encapsulée et a une identité distincte. Un composant Fractal est composé d'un contrôleur, appelé également membrane et d'un contenu. Le contrôleur; à travers des points d'accès nommés Interfaces; exerce (par réflexion) un contrôle arbitraire sur le contenu qui est lui-même défini récursivement

comme un ensemble de sous-composant. Ainsi, une interface est le point d'accès vers le composant (similaire à un "port" dans les autres modèles de composants) [BCL<sup>+</sup>, 04] et qui supporte un nombre fini d'opérations.

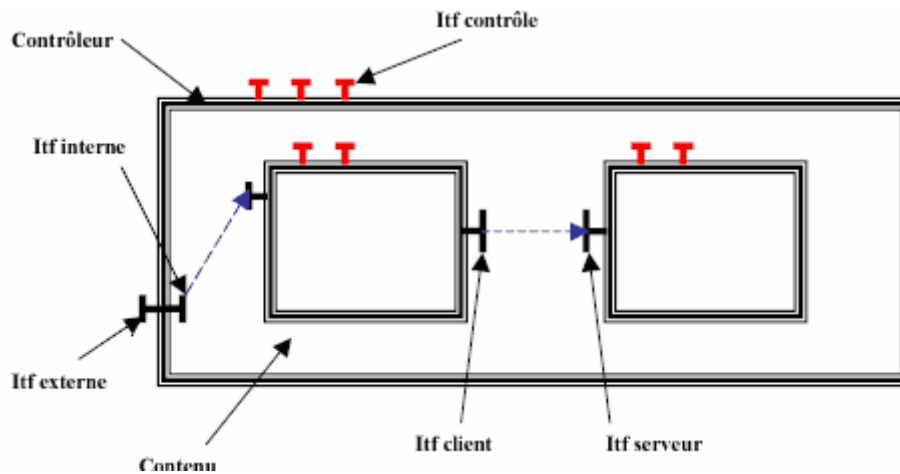


Figure 8: Schéma descriptif des interfaces.

**Les interfaces:** on parle ainsi de deux types d'interface:

-*les interfaces serveurs:* correspondent aux points d'accès acceptant des invocations d'opérations entrantes qui indiquent les services offerts par le composant.

-*les interfaces clients:* correspondent aux points d'accès acceptant des invocations d'opérations sortantes qui indiquent les services qu'utilise le composant pour réaliser ses propres services.

-Chaque interface d'un composant Fractal a un nom unique le distinguant des autres interfaces de ce composant.

**Le contenu:**

Le contenu d'un composant Fractal peut être constitué d'un ou plusieurs d'autres composants, nommés sous-composant qui peuvent être soit:

-*Primitifs:* encapsule un seul composant.

-*Composites:* une sorte de composant Fractal qui encapsule un ensemble de composants avec un niveau d'emboîtement quelconque (ainsi une autre propriété qui explique l'appellation Fractal) tel que le partage [Szy, 98].

Le contenu gère tous ce qui est fonctionnel dans le composant, c'est la partie "métier".

**Le contrôleur:**

Le contrôleur représente la partie "non-métier". Il permet de superviser un ou plusieurs aspects non fonctionnels (technique) tel que : L'introspection, la configuration, la sécurité, les transactions....Le contrôleur peut avoir un ensemble d'interfaces internes et d'interfaces externes.



Les interfaces externes sont accessibles de l'extérieur du composant, tandis que les interfaces internes sont accessibles uniquement des composants sous-composants.

### La communication et l'assemblage :

Les différentes entités d'une application basée composant doivent interagir entre eux pour accomplir une tâche. On parle donc de la communication et l'assemblage des composants.

La communication entre composant Fractal n'est possible que si leurs interfaces sont liées. Dans les autres modèles de composants le "binding" correspond au concept du "connecteur". Dans le modèle Fractal on parle de deux sortes de liaison (binding) :

-**Le binding primitif** (simple) : est une liaison entre une interface cliente et une interface serveur à la même adresse mémoire signifiant que l'opération émise par l'interface cliente doit être acceptée par une interface serveur spécifique (et non pas par n'importe quelle interface serveur).

-**Le binding composite** (composé) : représente le chemin de communication entre un nombre arbitraire d'interfaces composant.

On parle également d'un binding normal, import, ou export.

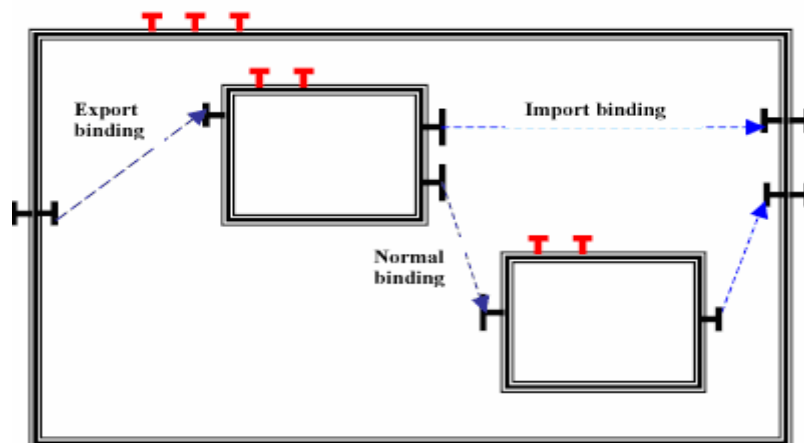


Figure 9: Composant Fractal avec les différents binding.

L'assemblage est une caractéristique importante du modèle Fractal, signifie qu'on peut inclure dans un composant un ensemble d'autres composants, partagés ou composés.

### Le typage :

Le modèle Fractal est doté d'un système de type pour les composants Fractal et ses interfaces. Le type de composant Fractal (non modifiable à l'exécution) est défini par l'ensemble de type de ses interfaces qui sont :

- **Le nom** de l'interface.
- **Le rôle** de l'interface qui est soit client ou serveur.
- **La signature** qui représente le type du langage.

- **La contingence** qui spécifie une garantie sur la disponibilité des fonctionnalités de l'interface : Obligatoire "Si l'interface serveur, il doit être disponible. Si interface client doit être liée à une interface serveur" ou Optionnel "pas de garantie, ni d'obligation".
- **La cardinalité** de l'interface selon sa liaison : singleton ou collection.

#### 4. Les composants logiciels et les services web :

Un service Web est une technologie permettant à des applications de dialoguer à distance via Internet, et ceci indépendamment des plates-formes et des langages sur lesquelles elles reposent. Ils ont pour objectif de faciliter les échanges de données, mais aussi l'accès aux applications au sein des entreprises mais surtout entre les entreprises.

Pour ce faire, les services Web s'appuient sur un ensemble de protocoles standardisant les modes d'invocation mutuels de composants applicatifs : WSDL et SOAP.

Malgré le nombre important d'avantages offerts par les services Web, ils présentent aussi un certain nombre d'inconvénients comme par exemple :

- la nécessité de développer d'autres logiciels pour utiliser les services Web, ce qui signifie qu'on doit prendre attentivement en considération l'architecture des applications et des services dans les systèmes qui utilisent ces services.
- pour développer des SW avec succès, on doit comprendre tout un ensemble de protocoles et d'APIs.
- les outils automatisés qui supportent les SW tels que : Microsoft Visual Studio, Net,..., sont récemment développés et leur nombre est très limité.
- la modification et la création de nouveaux services sont très complexes, et par conséquent, le temps pour obtenir de nouveaux services est plus long, car cela nécessite la modification ou l'ajout de nouveaux paramètres dans le réseau, ce qui rend leurs réutilisations plus difficile.

Les utilisateurs veulent que les services offerts par l'Internet soient plus performants, au moment où ils constatent que le modèle classique d'applications Web (groupe de pages, documents générés dynamiquement,...) ne répond pas à leurs besoins. La création de ces applications est plus complexe à cause de la nécessité d'assembler plusieurs fichiers et scripts afin de rendre un système exécutable en ligne. Elles sont très lentes et difficiles à utiliser. De ce fait, les développeurs cherchent à proposer de nouveaux modèles de programmation et de déploiement plus riches et plus évolutifs. Pour cela, Jeremy Allaire [Jer, 02] a introduit un modèle de programmation très puissant, avec le lancement de produit Macromedia MX, utilisé pour le développement des composants et des services Web. Macromedia MX offre une nouvelle technologie qui permet aux développeurs d'implémenter des services Web en utilisant

l'approche basée composant. Les composant ColdFusion (CFCs) est considéré comme un modèle simple et performant, servant à créer rapidement des composants Web. Des développeurs moins compétents peuvent facilement développer des composants à base d'objets grâce à une syntaxe basée sur les balises. Les développeurs familiarisés avec les langages de scripts tels que : CFML, JavaScript, PHP, ..., peuvent facilement maîtriser ce modèle.

Le but du composant ColdFusion est de fournir une structure simple avec une grande performance pour créer des composants à une architecture très importante : les applications Web.

## **5. L'approche orientée patron (pattern)**

La notion du patron a été utilisée pour la première fois en 1979 par Christophe Alexander dans le domaine de l'architecture, puis elle a été introduite au domaine d'ingénierie d'information. Ce concept est considéré comme un type particulier de composant permettant de capitaliser des solutions aux problèmes récurrents et garantir une certaine qualité des résultats.

### **5.1. Définition du patron :**

Les concepteurs de logiciels utilisent les patrons afin d'explorer leurs avantages. Ces concepteurs définissent le terme patron comme suit : « c'est à la fois une partie du système et une description qui explique comment le construire ». Plusieurs définition de cette notion de patron ont été trouvées dans la littérature spécialisée : « a pattern is a description of a general solution to a common problem or issue from which a detailed solution to specific problem may be determined » [AMB, 98].

“An idea that has been useful in one practical context and will probably be useful for others” [FOW, 97].

Le point commun entre toutes ces définitions est qu'elles considèrent le patron comme couple constitué d'un problème et d'une solution. Dans [FOW, 97], on trouve l'introduction de la notion du contexte : « un patron est une solution à un problème dans un contexte ». Ce patron peut être adopté pour un ensemble de situations récurrentes. Cet ensemble constitue ce qu'on appelle le contexte. Le problème c'est l'objectif à atteindre. Le modèle adapté pour résoudre le problème, c'est la solution.

Les patrons sont classifiés selon l'étape du cycle de vie à laquelle il s'adresse, donc on peut citer : patron d'analyse, patron de conception, patron d'implémentation, ...

Il existe ce qu'on appelle : des systèmes familles ou catalogue, regroupant les patrons. Un système de patron est un ensemble de patrons et des règles permettant de les combiner. Ce système peut être décrit à l'aide d'un formalisme, narratif ou structurés [HAS, 01], [KHA, 01].

## 6. Les patrons et l'IMS LD :

À cause des avantages offerts par les patrons, des auteurs comme Hernandez [Her et al, 05] ont proposés des CLFP (collaborative Learning Flow Pattern) pour décrire les techniques d'apprentissage collaboratif. Ces techniques permettent la structuration des interactions entre les apprenants lors des activités d'apprentissage ainsi que les informations échangées entre eux. Pour décrire les CLFP de façon formelle. Ces auteurs ont utilisé le langage de modélisation éducatif IMS LD. Ce langage a la possibilité de définir de nombreuses caractéristiques collaboratives des CLFP. Donc, les CLFPs et l'IMS LD ont le même objectif qui est la description du processus d'apprentissage. Le premier présente une description narrative de ce processus en langage naturel, et le deuxième est considéré comme un langage formel. L'utilisation d'IMS LD pour formaliser les CLFP offre plusieurs avantages :

- 1-Le traitement automatique des CLFP par des outils logiciels.
- 2-La personnalisation des CLFP pour des situations d'apprentissages particulières des différentes disciplines.
- 3-La séparation entre les structures de scénarios (les contenus) et les ressources d'apprentissage (les outils).
- 4-Il facilite la réutilisation et l'intégration des CLFPs dans différents systèmes, car cette spécification fournit un langage standard.

## 7. Conclusion :

Plusieurs travaux de recherche ont eu pour objectif d'introduire les approches du GL dans l'EIAH, nous avons présenté dans ce chapitre l'apport de cette introduction ainsi qu'une description de quelques approches du GL qui sont : l'approche orientée composant et l'approche orientée patron.

Malgré la multitude des avantages offerts par l'approche orientée patron, elle reste une approche un peu plus difficile à utiliser, cela est dû à la nécessité d'étudier un ensemble de formalismes dédiés à la description de la syntaxe ou la structure des patrons afin de choisir le plus approprié parmi eux, ainsi que l'usage de l'un des outils permettant la mise en œuvre de ces patrons. Cependant, les composants logiciels peuvent être implémentés directement en utilisant l'un des langages de programmation connus par tous les informaticiens, et exécutés sur n'importe quel environnement supportant ce langage.

## **Partie II**

**Chapitre 4 : Spécification et scénarisation des situations d'apprentissage collaboratif**

**Chapitre 5 : Implémentation d'un scénario de collaboration fondé sur les composants logiciels**

**Chapitre 6 : Exemple d'une unité d'apprentissage dédiée à la collaboration**

## **Chapitre 4:**

# **Spécification et scénarisation des situations d'apprentissage collaboratif**

## **1. Introduction :**

Vu l'importance du processus de scénarisation dans le e-learning, nos travaux se focalisent sur l'analyse de ce processus en se basant sur l'utilisation de l'un des langages de modélisation pour être capable de formaliser des scénarios pédagogiques réutilisables et flexibles. Pour certains auteurs [fer et al, 05], le scénario est un cadre social propice aux interactions, au sein duquel nous pouvons exprimer la dimension sociale et les différentes situations d'interaction.

Pour ce faire, nous avons préconisé de particulariser les imsmanifest avec des propriétés rehaussant l'apprentissage collaboratif. Nous avons mené une étude sur les différents modèles de collaboration issus du domaine de la psychopédagogie et nous avons retenu les modèles d'apprentissage collaboratif : Pyramid, Jigsaw, Brainstorming,...qui spécifient clairement les méthodes de collaboration que les apprenants doivent suivre.

Après avoir présenté le contexte de notre travail : les EIAH et plus précisément le CSCL, les différents travaux réalisés dans ce contexte, les approches de génie logiciel introduites dans les EIAH et les différents problèmes abordés, nous détaillons par la suite les différentes étapes de l'approche de conception que nous avons proposée dans le but de pallier à ce problème. La première étape consiste à définir le modèle d'activité et de scénario satisfaisant les objectifs de l'enseignant, la deuxième étape concerne la particularisation du scénario. La création de l'unité d'apprentissage est la dernière étape. La première section est consacrée à la définition de quelques modèles de collaboration. La deuxième section présente l'ingénierie centrée sur les scénarios et son apport. La relation entre l'IMS LD et le domaine du CSCL est déterminée par la troisième section. Notre démarche conceptuelle est décrite dans la quatrième section.

## **2. Les modèles de situations d'apprentissage collaboratif :**

Nous avons déjà signalé l'importance de l'apprentissage collaboratif pour l'évolution des apprenants. L'IMS LD, le formalisme que nous avons choisi d'introduire dans notre travail, ne supporte pas ce mode d'apprentissage. Il permet seulement d'indiquer s'il existe une coopération entre les apprenants ou non. L'objectif de notre travail consiste à modéliser des situations d'apprentissage collaboratif, ce qui nous a conduit à l'utilisation des moyens permettant de bien décrire le scénario de collaboration. Ces moyens sont effectivement nommés : modèles de collaboration.

Ces modèles servent à identifier de façon précise la manière de collaborer, ou encore spécifier les activités collaboratives ainsi que leurs enchaînements. Dans ce qui suit, nous exposons les deux modèles que nous avons adopté: Pyramid et Jigsaw.

### **2.1. Le modèle pyramid :**

Le modèle Pyramid permet de structurer les groupes d'individus et leur travail sous forme d'une pyramide à plusieurs niveaux. La base de cette pyramide représente les groupes initiaux, qui se regroupent successivement jusqu'à l'obtention de toute la classe qui est représenté par le sommet de la pyramide. Il est caractérisé par le fait que tous les groupes d'un même niveau réalisent la même tâche.

Ce modèle décrit le processus collaboratif de la manière suivante :

Chaque participant étudie le problème et propose une solution. Les groupes de participants comparent et discutent leurs propositions pour proposer une nouvelle solution. Ces groupes se regroupent pour construire des groupes de plus grandes tailles afin de générer une nouvelle proposition. Finalement, tous les participants doivent proposer une solution finale.

Le modèle pyramid peut avoir plusieurs objectifs, parmi lesquels on cite :

- Encourager la discussion.
- Assurer une interdépendance positive.

### **2.2. Le modèle Jigsaw :**

Le modèle Jigsaw est un modèle basé sur la division de la tâche, c'est à dire la division du problème global en sous problèmes. A la fin, tous les groupes se rejoignent afin de résoudre le problème entier. En effet, ce modèle est constitué de trois niveaux. Nous définissons ainsi le principe de ce modèle :

Au premier niveau, chaque apprenant travaille sur un sous problème. Puis, les groupe travaillent sur le même sous problème et se regroupent pour construire des groupes de tailles plus grandes appelés : 'groupes experts', cela se fait au sein du deuxième niveau. Enfin, les groupes d'étudiants des différents sous problème se rejoignent pour discuter et proposer une solution finale.

Comme le modèle pyramid, le modèle jigsaw permet d'atteindre plusieurs objectifs tels que :

- Résoudre les problèmes complexes.
- La prise en compte de l'individu.

## **3. L'approche méthodologique de conception de structures pédagogique :**

Rappelons que notre objectif est d'aider les enseignants à la formalisation des situations d'apprentissage collaboratif. Comme ces enseignants n'ont aucune compétence informatique,



notre démarche doit être capable de les aider dans le processus de création de leurs propres Learning Design.

On a étudié certains modèles d'activité: pyramid et jigsaw, qui représentent la méthode de collaboration que les apprenants doivent suivre. Afin de représenter ces modèles, on a choisi le standard IMSLD car il offre une multitude d'avantages comme par exemple la prise en compte d'une grande variété de modèles pédagogiques, d'où réside sa flexibilité. Dans la section suivante, nous présentons notre approche méthodologique [Ben et Ser, 07]. Nous voulons permettre aux enseignants auteurs d'exprimer des modèles de scénarios au niveau 1, qui seront exécutés par les apprenants, voire les enseignants, au niveau 2 (Figure 10).

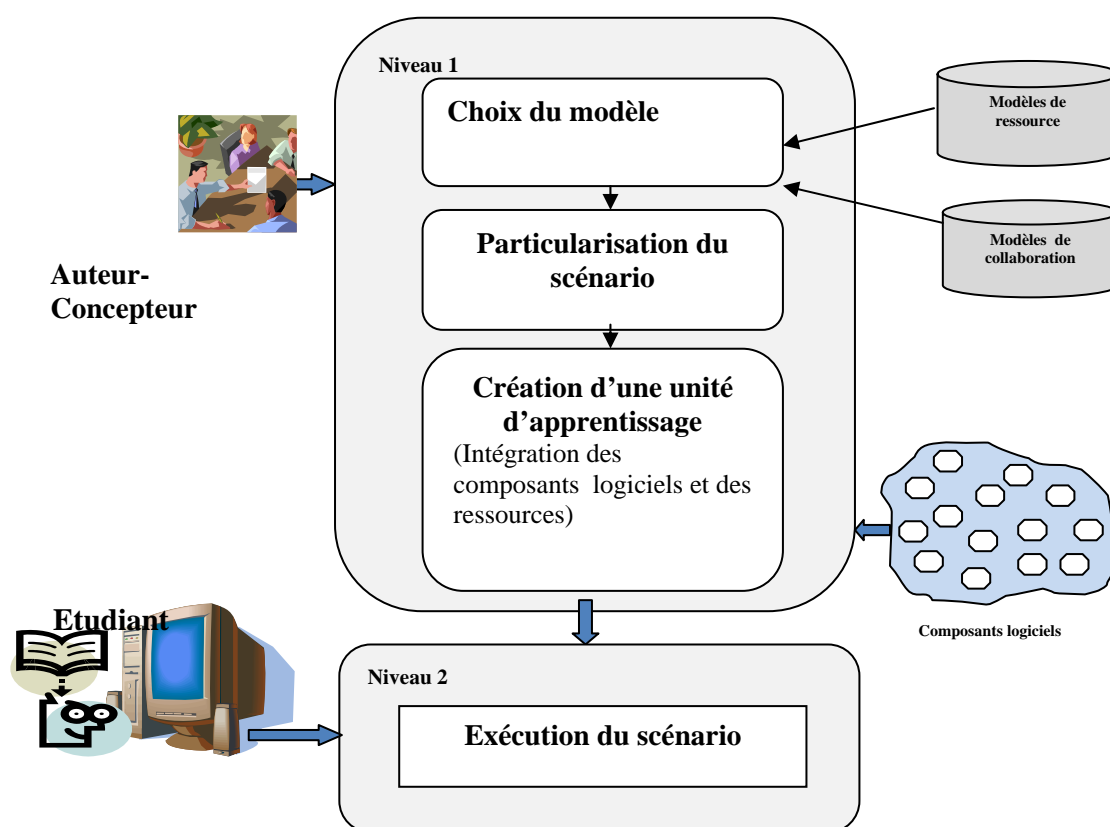


Figure 10 : Etapes de scénarisation d'une ressource dédiée à l'apprentissage collaboratif.

### 3.1. Architecture du modèle de scénarisation de situations d'apprentissage collaboratif :

Le schéma suivant décrit les différentes étapes de notre démarche de conception.

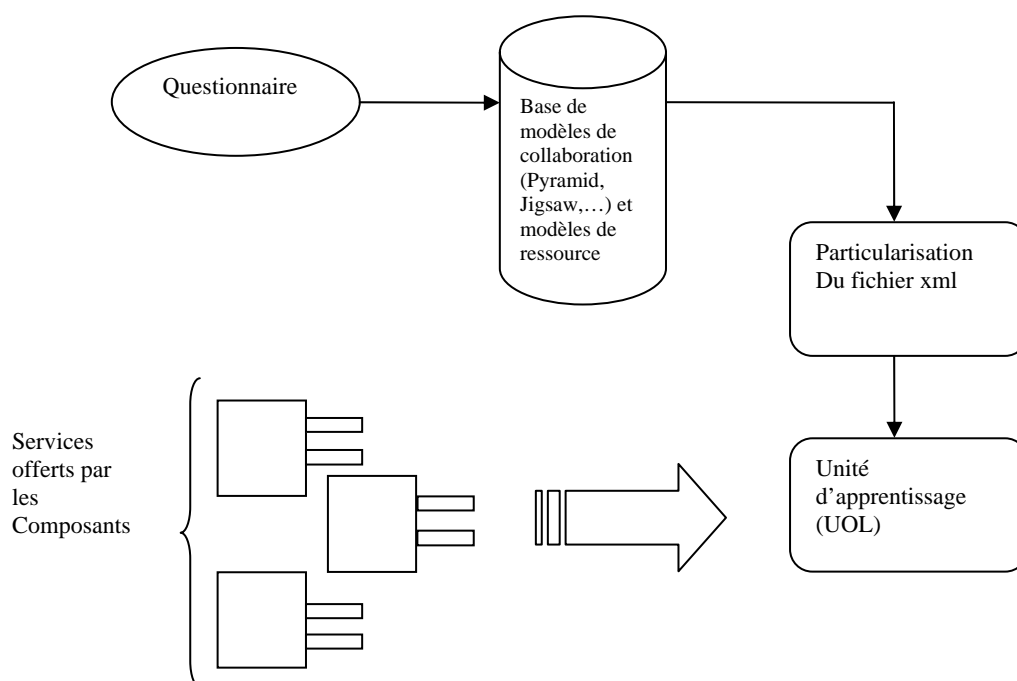


Figure11 : Schéma de conception

La première étape consiste en l'extraction d'un fichier XML décrivant le modèle de collaboration approprié selon les réponses d'un questionnaire, à partir d'une base de modèles. Ce questionnaire doit permettre aussi de déterminer le modèle de ressource qui doit être utilisé, tiré des modèles d'unités pédagogiques (UP).

Giacomini [Gia, 05] a proposé 16 modèles d'unité pédagogique et les scénarios qui leur sont associés, prenant en compte trois types d'unités d'apprentissage qu'elle a déterminé à partir de certains critères comme : existence des parties théoriques dans un module de formation, différents types d'exercices (QCM, problèmes à résoudre, texte à trous, etc), de projets, des types d'évaluation et/ou de l'auto évaluation,... Ces trois types sont : présentation des concepts théoriques, exercices comme application du cours et projets (travaux pratiques). Suivant, Giacomini l'unité pédagogique est composée d'un ou plusieurs scénarios pédagogiques.

La deuxième étape consiste en la personnalisation du fichier précédent selon le problème posé en détaillant tout les éléments d'un LD complet.

Après la définition des ressources et des services nécessaires durant l'exécution du LD et les mettre dans un package avec le fichier XML (le manifest), une unité d'apprentissage est créée. Dans ce qui suit, on donne plus de détail sur notre démarche de conception.

### 3.1.1. Le choix du modèle d'activité:

La première étape de notre démarche est la création d'une base contenant un ensemble de fichiers XML décrivant chacun l'un des modèles de collaboration mentionnés ci-dessus.

Ces fichiers XML dit « manifests » sont l'implémentation des scénarios pédagogiques modélisés avec la spécification IMSLD. L'IMSLD exige la définition de plusieurs aspects ou concepts dans le manifest. Dans chaque manifest, on peut trouver donc plusieurs informations : l'objectif de l'apprentissage, les différents rôles, les activités, les ressources et les services nécessaires pour l'apprentissage,.... Cette spécification comme nous l'avons déjà mentionné offre plusieurs avantages, le plus important est sa richesse, en d'autre terme, il permet de prendre en compte tout type de pédagogie, ce qui signifie qu'on peut modéliser les modèles de collaboration que nous avons adopté en utilisant ce formalisme.

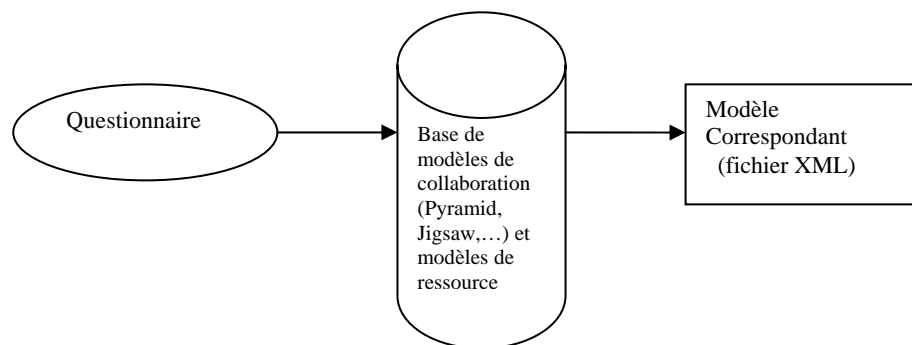


Figure12 : L'extraction du modèle de collaboration

Au début, l'enseignant doit répondre à un ensemble de questions. Ces réponses permettent de déterminer le modèle d'activité et le modèle de ressource appropriée au problème posé par l'enseignant, car ce dernier n'a aucune idée sur ces modèles. La définition de ces deux modèles, se traduit par l'extraction de fichier xml décrivant le modèle de collaboration à partir de la base des modèles d'activité, ainsi que l'ajout du modèle de ressource dans le fichier xml pour être utilisée lors du processus d'apprentissage.

En effet, il s'agit de deux questionnaires. Le premier permet de choisir l'objectif de l'apprentissage, et le deuxième permet de préciser le type du contenu pédagogique, ou encore les ressources qui vont être choisies par l'enseignant lors de la phase de conception du scénario et utilisées ultérieurement par les apprenants durant le processus d'apprentissage. De ce fait, on a choisi d'utiliser les 16 types d'UP définis par Giacomini, en tant que modèles de ressources. La première question donne la possibilité à l'enseignant de choisir entre plusieurs choix représentant

les différents objectifs d'apprentissage susceptibles d'être atteints par ce dernier. Par la suite, on donne un exemple d'un ensemble de questions posées à l'enseignant.

***Exemple d'un questionnaire :***

Quels sont les objectifs de l'apprentissage :

- Encourager la discussion.
- Assurer une interdépendance positive.
- Résoudre des problèmes complexes.

Quel est le type du contenu pédagogique :

- Concepts théoriques.
- Résolution des problèmes avec exercices.
- Concepts théoriques et Résolution des problèmes.
- Projets.

Les réponses sont représentées par les carrés noirs. Les réponses du concepteur pédagogique à ce questionnaire nous permettent de conclure que le modèle de collaboration et le modèle de ressource qui doivent être utilisés sont respectivement : Pyramid et le modèle 10.

Ces deux questionnaires servent donc à faciliter les choses pour l'enseignant, c'est à dire le guider dans la spécification du modèle de collaboration après avoir choisi l'un des objectifs mentionnés ci-dessus, sans pour cela être obligé d'avoir une connaissance préalable sur ces modèles.

***3.1.2. Particularisation du scénario :***

Cette étape consiste à ajouter des informations déterminées par l'enseignant dans le fichier XML décrivant le modèle choisi. Ces informations peuvent être :

- l'objectif de l'apprentissage
- la taille du groupe.
- les descriptions des différentes activités.
- le nombre de niveaux s'il s'agit du modèle : Pyramid.

Bien que ces informations soient définies par l'enseignant, il existe d'autres informations qui sont fixées à l'avance mentionnés dans le fichier XML comme :

- les rôles : enseignant et apprenant.
- les activités selon chaque modèle.

Comme nous l'avons déjà mentionné, nous avons choisi d'utiliser dans notre travail deux modèles : pyramid et jigsaw. Dans ce qui suit nous définissons deux exemples de scénarisation en utilisant le modèle pyramid qui est déterminé à partir des réponses de l'enseignant au questionnaire précédent, et le modèle jigsaw.

### L'exemple du modèle Pyramid :

Le tableau suivant décrit le modèle d'activité Pyramid, d'où différents individus se regroupent successivement pour construire des groupes de plus grande taille dans le but de proposer une solution finale agréée par tous les participants. Ce tableau décrit aussi l'un des 16 modèles d'UP mentionnés précédemment. On parle ainsi du modèle 10 qui est basé sur l'apprentissage des concepts théoriques.

	Modèle d'activité	Modèle d'UP
Nom du modèle	Pyramid	Le modèle 10 : Apprentissage des concepts théorique
Définition du Modèle	<p>Au niveau 1, chaque participant étudie le problème et propose une solution. Les groupes de participants comparent et discutent leurs propositions pour proposer une nouvelle solution. Ces groupes se regroupent pour construire des groupes de plus grande taille afin de générer une nouvelle proposition, c'est le niveau 2.</p> <p>Au niveau 3, tous les participants doivent proposer une solution finale.</p>	<p>Le modèle d'activité est basé sur l'apprentissage des concepts théoriques et propose des activités d'apprentissage individuel comme l'analyse d'un chapitre de cours ou la recherche d'informations pertinentes.</p> <p>Ensuite les étudiants discutent sur les notions apprises puis rédigent une synthèse du groupe sur le sujet de chapitre étudié. Une fois que toutes ces phases sont réalisées et validées par l'enseignant, les étudiants passent à l'étude du chapitre suivant en conservant les mêmes types d'activités et le même enchaînement.</p>

*Tableau1 : la définition du modèle pyramid et le modèle 10 d'UP.*

En effet, ce tableau définit non seulement la manière de collaborer suivie par les participants dans le modèle : Pyramid, mais il détermine aussi comment les individus collaborent dans le

modèle d'UP 10. Dans notre travail, on s'intéresse seulement à la collaboration décrite par les modèles de collaboration et le type du contenu pédagogique décrit par les modèles de ressource. Le schéma suivant explique mieux le modèle de collaboration pyramid constitué de trois niveaux. Le niveau1 représente les groupes initiaux au sein desquels, chaque individu étudie le problème posé et propose une solution, ces individus discutent les solutions proposées par chacun pour aboutir à une seule proposition au niveau 2, les groupes se rejoignent dans des groupes de plus grande taille et proposent une autre solution. Tous les groupes se regroupent encore une fois pour construire toute la classe, c'est le niveau 3.

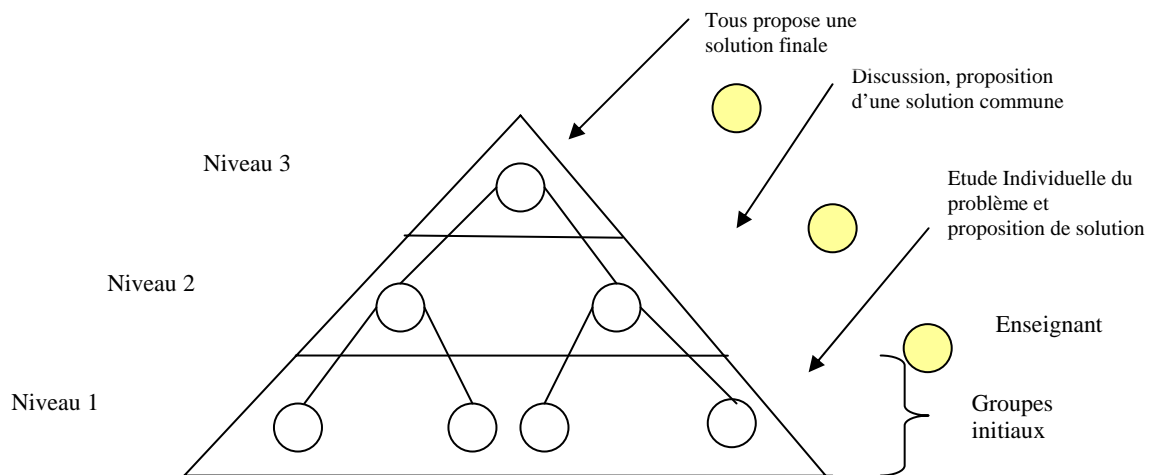


figure13 : Le modèle de collaboration : Pyramid.

Pour décrire le processus d'apprentissage avec plus de rigueur, on a utilisé le diagramme d'activité UML. Ce diagramme représente la première étape de la conception d'IMSLD, qui sert de base pour créer le fichier xml conforme à la spécification IMSLD.

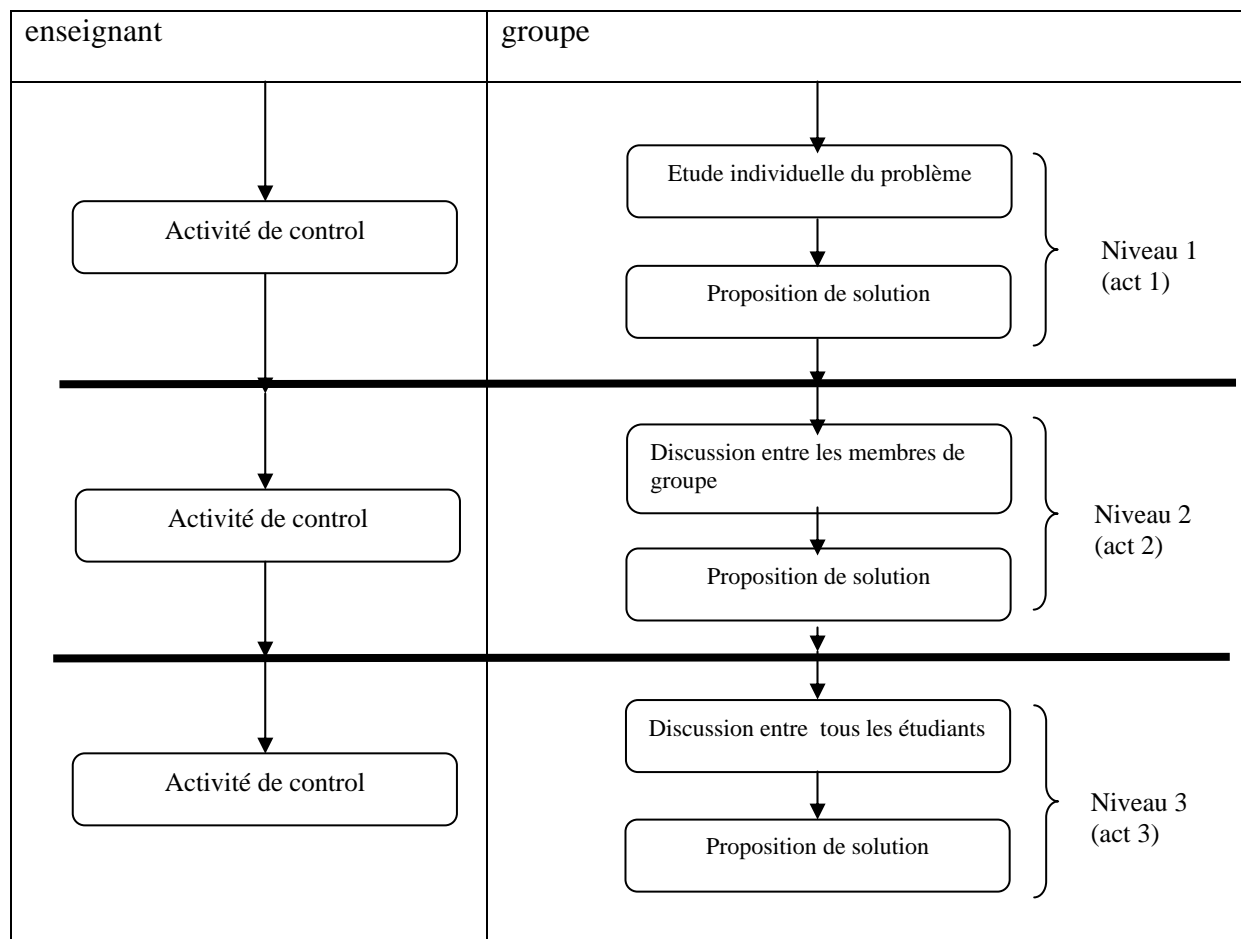


Tableau2 : Play d'un learning design basé sur le modèle : Pyramid

Le Tableau2 représente un play de learning design basé sur le modèle de collaboration pyramid, constitué de trois niveaux, chaque niveau correspond à un act du Play. Au niveau 1, chaque apprenant étudie individuellement le problème et propose une solution. Au niveau 2, les étudiants se regroupent pour former des groupes de plus grande taille, qui discutent et proposent une nouvelle solution commune. Au dernier niveau, tous les étudiants se regroupent pour discuter et élaborer une solution finale. L'enseignant peut intervenir à chaque niveau pour contrôler leurs activités, vérifier leurs réponses et répondre à leurs questions.

Après avoir déterminé ces deux modèles à partir des réponses de l'enseignant, le fichier xml extrait de la base des modèles doit être particularisé en ajoutant des informations concernant par exemple : les descriptions des activités d'apprentissage (lecture d'un document d'architecture), les groupes,....

Dans ce fichier, on a défini les différents rôles participant à l'apprentissage ainsi que les activités qui leur sont associées. Ce manifest contient d'une part des informations constantes, c-à-d qu'elles existent dès le début dans ce manifest telles que : les rôles, les activités réalisées dans chaque niveau..., et d'autre part, on peut remarquer des informations dynamiques, qui ont été rajoutées à partir des réponses données par l'enseignant, ces informations sont, par exemple : l'objectif de l'apprentissage, la description de chaque activité, ....

### L'exemple du modèle Jigsaw :

Le deuxième modèle d'activité que nous avons choisi d'utiliser est le modèle : jigsaw. Ce modèle est basé sur la division du problème posé en sous problèmes, d'où chaque individu ou groupe initial étudie un sous problème, puis ces individus (groupes initiaux) se regroupent afin de proposer une solution au problème globale.

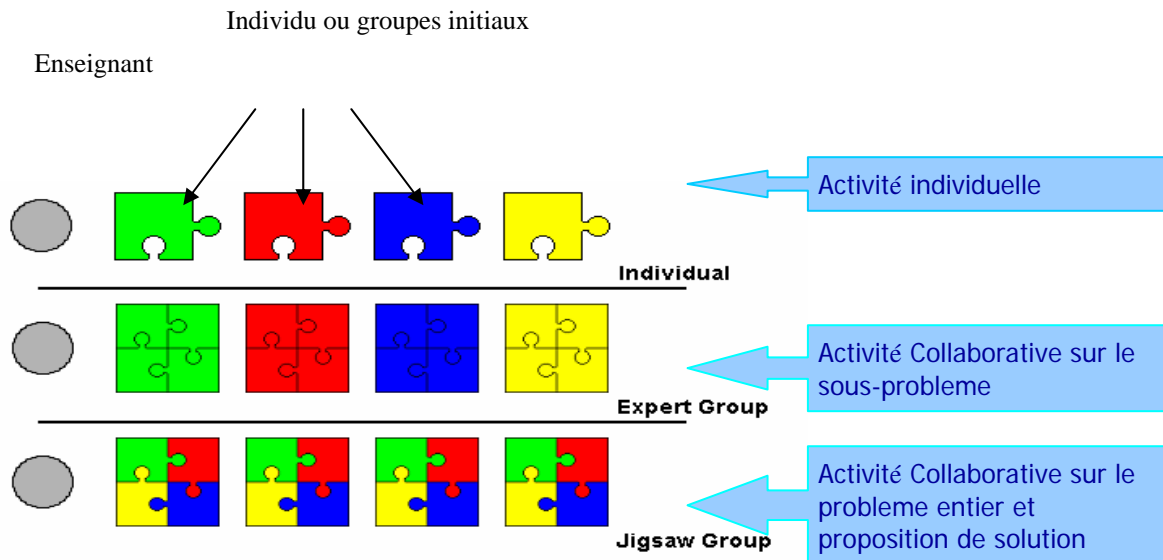


Figure14 : Le modèle de collaboration jigsaw.

Ce schéma décrit le modèle d'activité jigsaw qui est constitué de trois niveaux. Dans le niveau 1, chaque participant (individu ou groupe initial) étudie un sous problème. Dans le niveau 2, les participants des différents groupes qui étudient le même sous problème se regroupent dans des groupes appelés : "*groupes experts*" pour échanger leurs idées. Dans le niveau 3, les groupes d'étudiants des différents sous problèmes, se rejoignent encore une fois pour approfondir leurs réflexions sur tout le problème et proposer une solution (Groupes Jigsaw). Le diagramme d'activité UML suivant décrit le processus d'apprentissage en utilisant le modèle jigsaw.



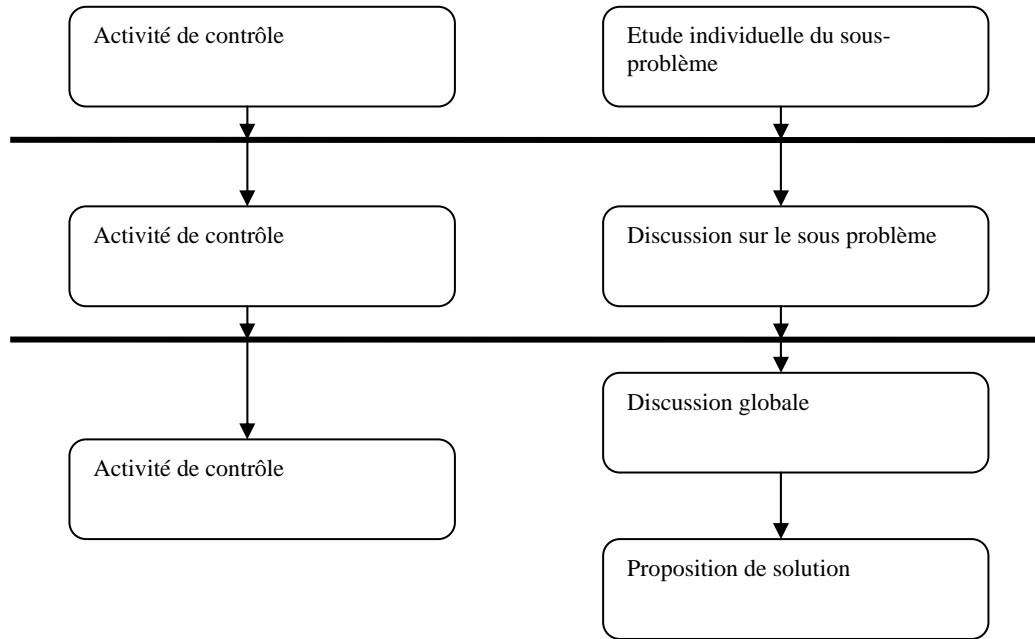


Figure15 : Diagramme d'activité UML représentant le modèle de collaboration : Jigsaw.

Ce diagramme représente un learning design basé sur le modèle de collaboration jigsaw, constitué de trois niveaux. Au niveau 1, chaque participant étudie individuellement un sous problème. Au niveau 2, les participants étudient le même sous problème se rejoignent pour construire des groupes experts. Finalement, les groupes d'étudiants des différents sous problèmes, se regroupent pour former ce qu'on appelle 'groupes Jigsaw', discutent et proposent une solution, c'est le dernier niveau. Comme dans le modèle Pyramid, l'enseignant intervient pour guider, contrôler et répondre aux questions des participants.

**3.1.3. Création d'unité d'apprentissage :**

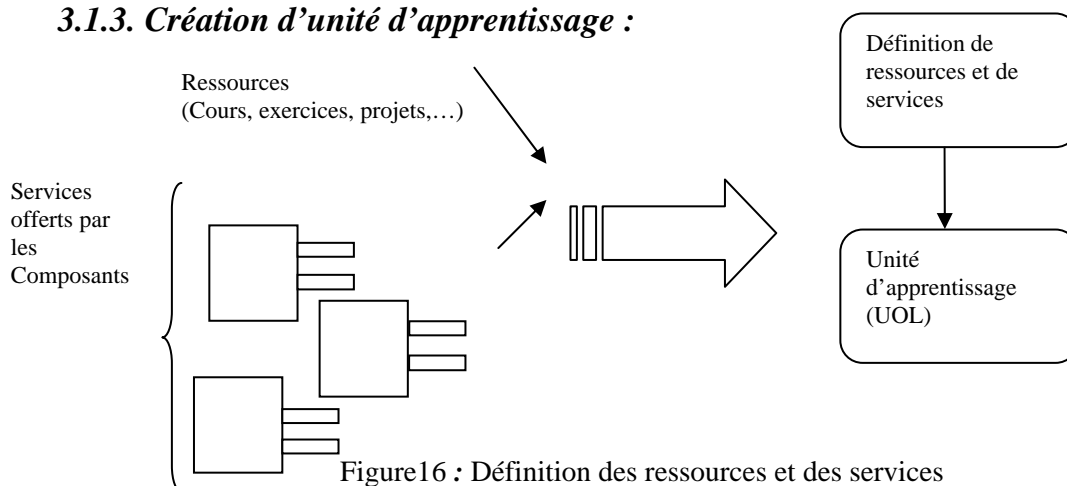


Figure16 : Définition des ressources et des services

Les apprenants ont besoin de certains nombre de ressources pour les utiliser lors du processus d'apprentissage comme : cours, exercices, cours et exercices,... Ces ressources sont représentées par les modèles de ressources qui se trouvent dans la base, le système choisit parmi ces

ressources celles nécessaires pour l'apprentissage à partir des réponses de l'enseignant au questionnaire. Lors de la réalisation des activités, les apprenants ont besoin aussi de certains services tels que : chat, forums, simulateur..., qui leur permettent de communiquer entre eux pour discuter, échanger des idées,... Ces services sont implémentés en utilisant les composants logiciels qui offrent pas mal d'avantages, on cite parmi ces avantages la réutilisation. Ces composants vont être appelés lors de l'exécution de l'unité d'apprentissage.

Les composants logiciels sont des entités réutilisables et déployables, permettant d'offrir plusieurs fonctionnalités. En effet, il existe aujourd'hui un grand nombre de modèles de composant, nous avons choisi parmi eux le modèle Fractal car son implémentation est plus simple. Pour la conception d'un composant logiciel selon le modèle Fractal, il faut tout d'abord déterminer s'il est composite ou primitif, puis définir ses interfaces et les interfaces de ses sous composants, en d'autre terme, la définition de son architecture.

Nous présentons par la suite l'architecture du composant chat, qui présente un exemple de services de communication. Ce service permet aux apprenants de communiquer en envoyant des idées, des solutions, des fichiers,... Ils peuvent aussi envoyer leurs questions à l'enseignant et il leur envoie des réponses.

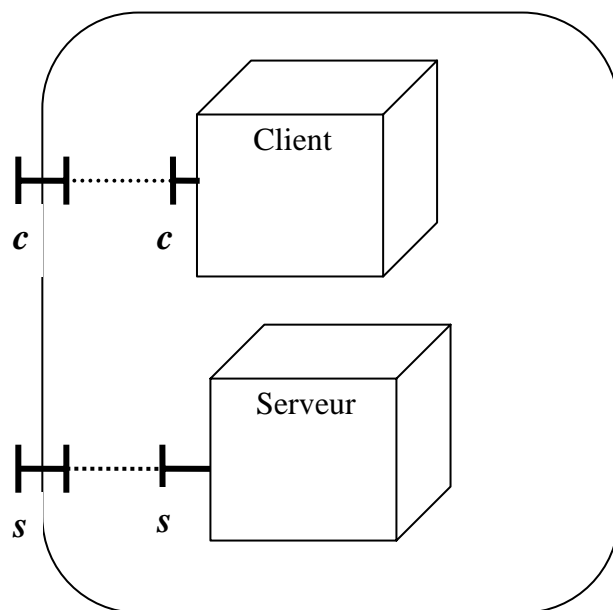


Figure17 : Architecture du composant chat.

L'architecture que nous avons choisi pour le composant chat est une architecture client/serveur, car il s'agit d'un principe de requête/réponse : l'apprenant envoie une question et l'enseignant ou un autre apprenant répond à cette question. Cette architecture est constituée de deux sous composants : client et serveur, encapsulés par un troisième composant appelé : composant racine.

- **Composant serveur :**

Ce composant possède une interface serveur qui offre un service permettant d'établir une connexion avec le client et attendre ses messages, puis il envoie ses réponses. Cette interface interne est connectée avec une interface externe (possède le même nom que celui de l'interface interne's') du composant racine pour pouvoir être accessible de l'extérieur du composant racine (autres composants).

- **Composant client :**

Il possède une interface cliente permettant d'établir une connexion avec le serveur et envoyer des messages, dans le cas où le serveur est lancé. Cette interface est connectée avec l'interface externe : 'c'.

- **Composant racine :**

C'est le composant globale et possède deux interfaces externes. La première est connectée avec l'interface cliente et la deuxième est liée avec l'interface serveur.

Lorsque les ressources et les services nécessaires sont déterminés et regroupés avec le manifest (fichier xml décrivant le scénario pédagogique) dans un package compressé, une unité d'apprentissage est achevée et peut être exécutée.

### 3.2. La structure d'une unité d'apprentissage en IMSLD :

Le terme : unité d'apprentissage (Unit Of Learnig) est utilisé pour indiquer un cours, un module d'enseignement, une leçon, etc. Elle est composée selon le point de vue IMS, d'une partie « organisation du contenu » et une partie « ressources » (figure17). Le formalisme IMSLD utilise une structure arborescente pour organiser les éléments pédagogiques tels que : scénario, acte, activité d'apprentissage, ressources (des liens vers les ressources physiques),.... L'organisation du contenu (éléments pédagogiques) est décrite par un fichier XML appelé : manifest, on va détailler sa structure dans la section suivante. La deuxième partie désigne les ressources réelles (cours et services).

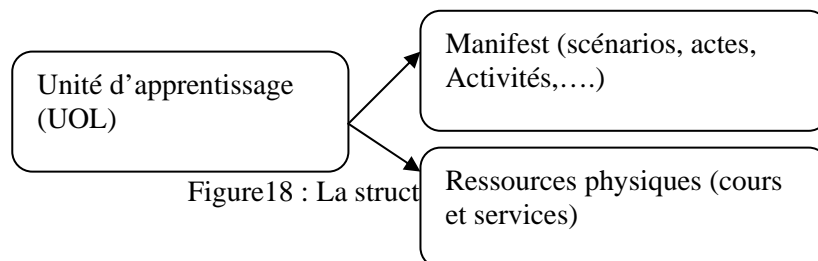


Figure18 : La struct

#### 3.2.1. Les éléments d'un manifest :

Comme nous l'avons déjà mentionné, un manifest sert à décrire les différents éléments d'un scénario pédagogique. Ces éléments ne sont que les concepts définis dans le modèle conceptuel de la spécification IMSLD. En effet, seuls les concepts utilisés dans le niveau A sont pris en compte. Les éléments principaux d'un manifest sont : les rôles, les activités, la méthode et

l'environnement. On appelle le dernier élément : ressources. En plus de ces éléments, le manifest contient un élément : *title*, pour exprimer le titre d'apprentissage et un élément : *learning objectives*, cet élément permet d'indiquer l'objectif de l'apprentissage. Les différents composants d'un manifest sont décrits par la figure suivante :

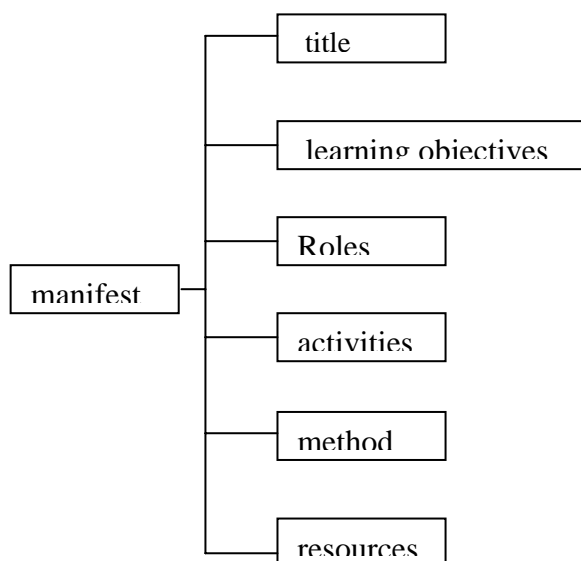


Figure19 : Les éléments d'un manifest.

Les différentes structures que nous avons conçues pour ces éléments seront détaillées par la suite.

### 3.2.1.1. Les rôles : *roles* :

Cet élément permet de définir les différents rôles qui peuvent être joués par les personnes qui peuvent participer dans le processus de l'apprentissage. En effet, il existe deux types de rôle : enseignant (*stuff*) et étudiant (*learner*). On a choisi une structure très simple pour l'élément rôles, constituée d'un ou plusieurs sous éléments appelés : *learner* et un sous élément *teacher*, chacun de ces éléments est constitué à son tour d'un titre permettant de décrire son nom. Cette structure est présentée par le diagramme en arbre suivant :

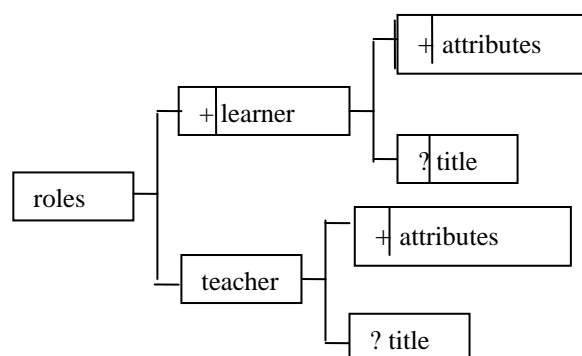


Figure20 : Diagramme en arbre dérivant l'élément *rôles*.

### 3.2.1.2. Les activités : *activities* :

Comme son nom l'indique, cet élément sert à présenter les différentes activités réalisées par les différents rôles définis par l'élément précédent. Il comporte un ou plusieurs : *learning*

*activity*, *structure activity* ou *support activity*. *Learning activity* est une activité simple qui contient les sous éléments : *title* et une *description* pour donner plus de détail sur cette activité, alors que *structure activity* désigne une activité structurée, composée de deux ou plusieurs *learning activity*. Le troisième type d'activité qui est : *support activity*, est l'activité déterminée pour être effectuée par l'enseignant, les sous éléments qui la composent sont identiques à ceux de l'élément *learning activity*. Le diagramme suivant illustre la structure de l'élément *activities*.

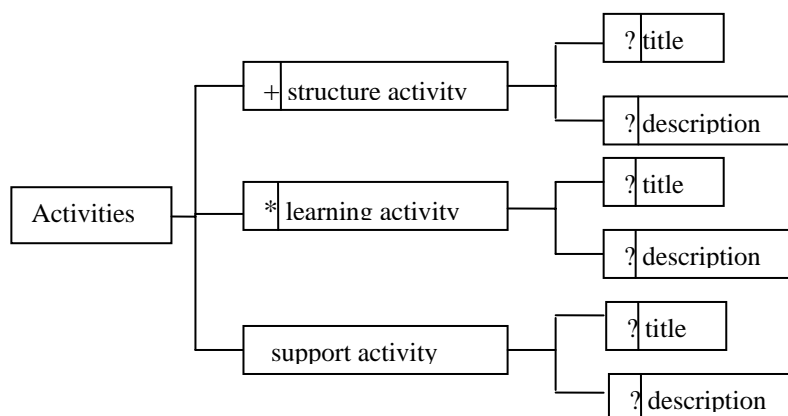


Figure21 : Diagramme en arbre dérivant l'élément *activities*.

### 3.2.1.3. La méthode : *method* :

C'est l'élément le plus compliqué d'un manifest, car il contient beaucoup d'éléments. L'élément *method* est constitué d'un ou plusieurs sous élément nommés : *play*, qui est composé d'un *title* et d'un ou plusieurs *act*. L'*act* possède lui aussi un *title*, et fait référence à un ou plusieurs *role part*. Les sous éléments : *role-ref* et l'un des sous éléments : *structure-activity-ref*, *support-activity-ref*, *learning-activity-ref*, qui sont regroupés dans l'élément : *activity-ref*. Dans ce qui suit un diagramme qui présente cet élément.

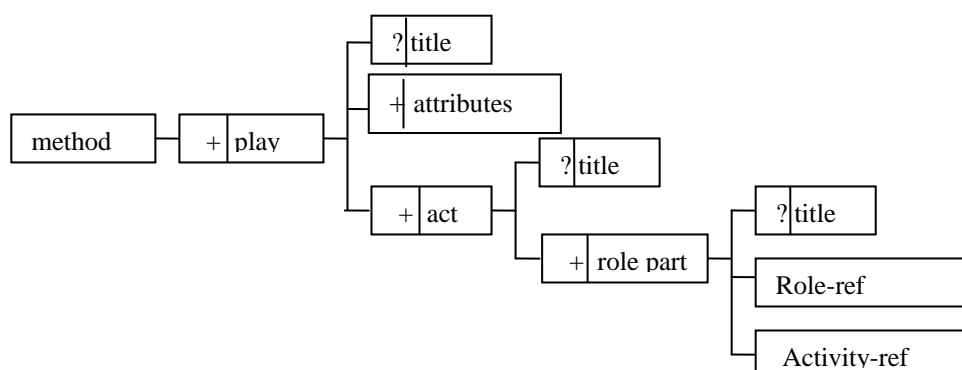


Figure 22: Diagramme en arbre dérivant l'élément *method*.

### 3.2.1.4. Les ressources : *resources* :

Les cours et les services de communication qui peuvent être utilisés lors de l'apprentissage, on les appelle : ressources. L'élément *resources* est structuré d'un ou plusieurs sous éléments nommés : resource. La structure de cet élément est présentée par la figure suivante :

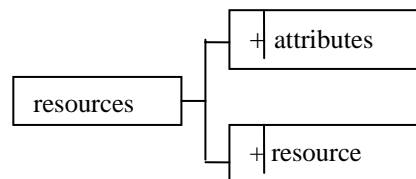


Figure 23: Diagramme en arbre dérivant l'élément *resources*.

#### 4. Conclusion:

Dans ce chapitre, on a présenté les deux modèles de situations d'apprentissage collaboratif : pyramid et jigsaw, l'ingénierie basée sur la scénarisation, la relation entre l'IMSLD et CSCL, et notre démarche de conception qui commence par l'extraction du fichier xml décrivant le modèle de collaboration déterminé à partir des réponses du concepteur pédagogique à un questionnaire concernant : les objectifs de l'apprentissage et le type du contenu pédagogique. Ce questionnaire permet aussi de déterminer le modèle décrivant le contenu pédagogique qui va être utilisé comme ressource. La deuxième étape consiste en la particularisation du fichier xml en fonction des deux modèles déterminés précédemment.

Puis, la définition de toutes les ressources et les services nécessaires pour le processus d'apprentissage. Ces services permettent d'assurer la collaboration entre les apprenants,..., sont implémentés sous forme de composants logiciels et on a présenté la structure d'un exemple de composant : le composant chat. Nous avons terminé par la description de la structure proposée pour les manifests utilisés.

# **Chapitre 5**

## **Implémentation d'un scénario de collaboration fondé sur les composants logiciels**

### 1. Introduction :

Après avoir défini les différentes étapes de l'approche conceptuelle, nous allons présenter dans ce chapitre, le schéma fonctionnel du système, le prototype réalisé, les outils utilisés pour l'implémentation, les parties de l'un des manifests qu'on a implémenté, on parle ainsi du manifest qui décrit un scénario d'apprentissage avec le modèle pyramid disposant de trois niveaux, la réalisation d'un exemple de services de communication chat ainsi que les différentes étapes suivies pour l'exécution d'une unité d'apprentissage (UOL) par copperCore.

### 2. Schéma fonctionnel du système :

Nous présentons par la suite le schéma fonctionnel de notre système:

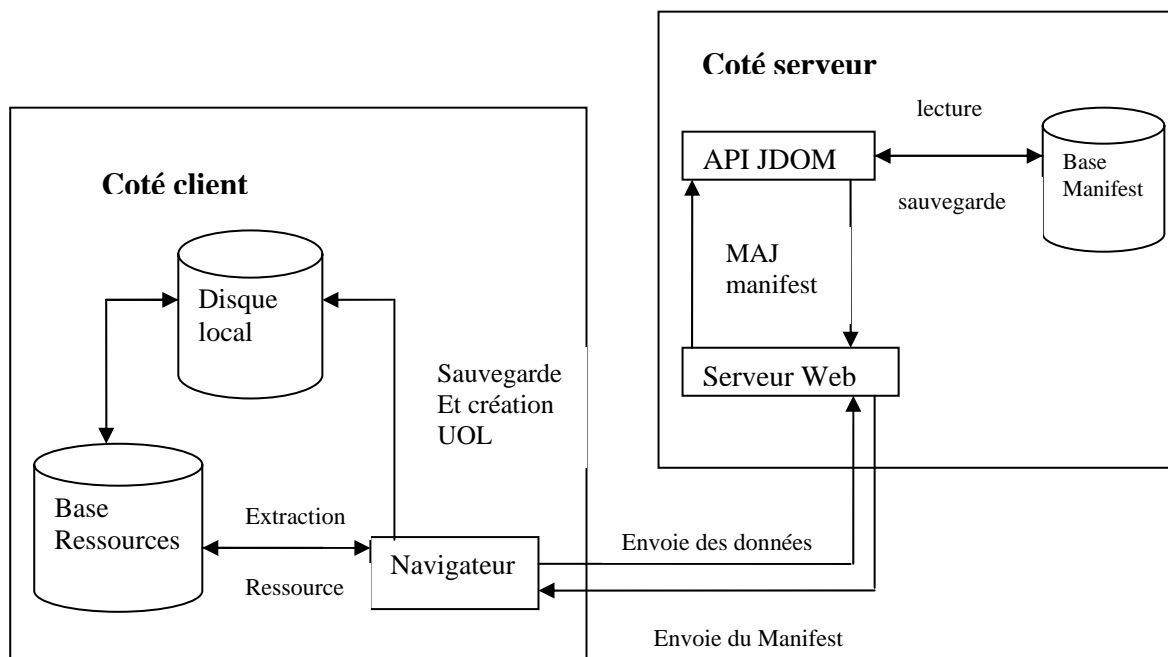


Figure24 : schéma fonctionnel du système.

### 3. Les outils utilisés :

Les exigences du domaine de l'apprentissage nécessitent que notre outil soit fondé sur le Web, plus précisément sur une architecture client/serveur, ce qui signifie qu'on a besoin d'un serveur http (repose sur le protocole http qui a une particularité de maintenir la connexion entre le client et le serveur uniquement durant l'échange de la requête et la réponse).

Notre outil doit donc disposer de deux applications : la première est une application cliente (applet) avec une interface graphique, et la deuxième est une application serveur (servlet) qui



serve à traiter les requêtes envoyées par le client (navigateur) et renvoyer une réponse à ce dernier.

### **3.1 Serveur Web :**

En effet, le serveur Web Apache est développé pour tout type de plate forme, pour cela on a choisi de travailler avec le serveur Tomcat Apache développé par le projet Jakarta du groupe Apache, ainsi ce serveur a pour avantage d'être capable de déployer les pages Web classiques (HTML) et toutes les technologies propres à java comme : les applets, les servlet, les JSP (Java Server Page),.....

### **3.2. Le choix de langage de programmation :**

Nous avons choisi pour implémenter notre outil, le langage le plus utilisé aujourd'hui : Java, car il présente un certain nombre d'avantages ce qui le rend approprié à nos besoins. Comme nous avons adopté la spécification IMSLD pour la modélisation du processus d'apprentissage, on est obligé de travailler avec le langage XML pour implémenter le modèle d'information. Ce langage permet le traitement automatique par différents systèmes informatisés (interopérabilité).

#### ***Pourquoi Java :***

L'outil de conception des situations d'apprentissage collaboratives que nous visons doit pouvoir être consultable depuis n'importe quel poste appartenant à un réseau ou depuis Internet, par conséquent, il doit être portable sur tout type de système d'exploitation. Vue les contraintes mentionnées précédemment, notre choix s'est dirigé vers le langage Java, car il s'agit d'un langage objet, portable, interprété architecturalement neutre, disposant d'une bibliothèque très riche et développé pour supporter les technologies Web. Toutes ces caractéristiques indiquent qu'il sera utile pour implémenter notre application que nous voyons répartie sur deux postes client et serveur.

### **3.3 Environnement de programmation :**

Nous avons choisi pour le développement de notre projet l'environnement de programmation : **Eclipse** (version 3.2), qui est l'un des derniers outils de développement en java développé par IBM. Cet environnement possède de nombreux points forts qui sont à l'origine de son énorme succès, parmi lesquels on cite :

- Un support multi langage grâce à des plug-in dédiés : Cobol, C, PHP, C #,....
- Supporté par plusieurs plates formes d'exécution : Windows, Linus,....
- Malgré son écriture en Java, l'exécution de l'eclipse est très rapide grâce à l'utilisation de la bibliothèque SWT.

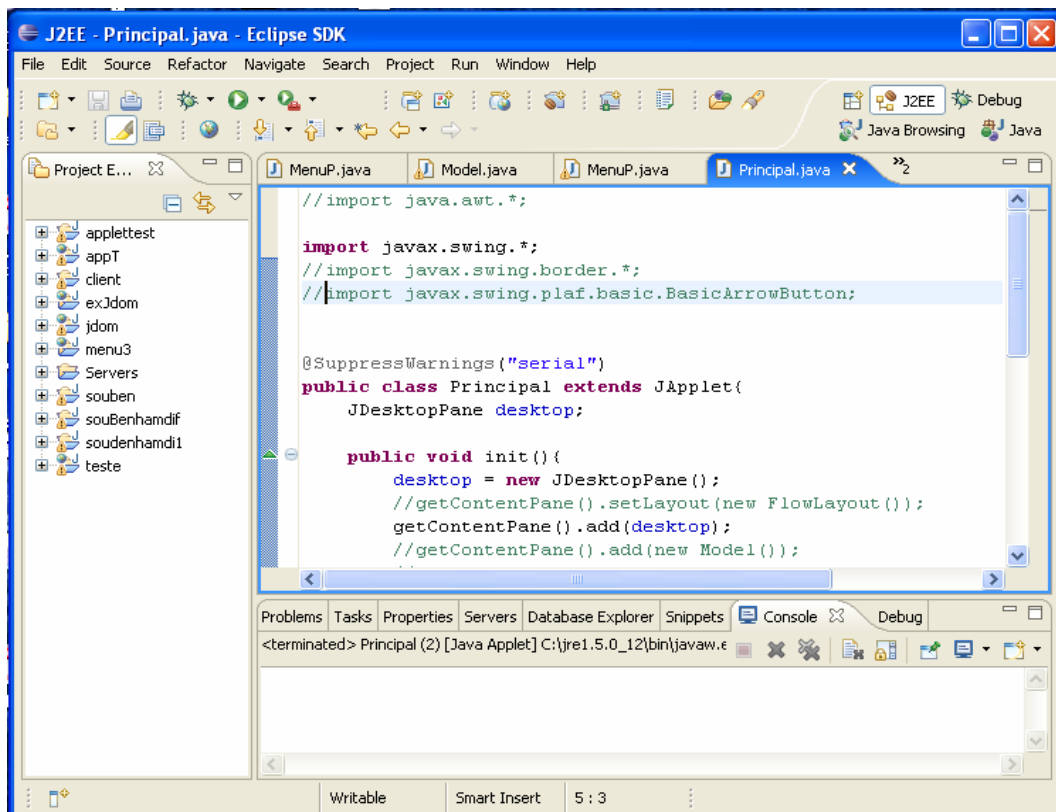


Figure25 : l'Interface de l'environnement eclipse.

#### 4. Implémentation du prototype :

Après avoir déterminé les exigences de notre outil, nous nous intéressons maintenant à son implémentation complète. Cet outil est une Applet java qui s'exécute côté client «enseignant», lui permettant de concevoir des situations d'apprentissage collaboratifs. La figure '25' permet de présenter l'interface principale de l'outil réalisé. Cette interface comporte deux parties :

- Barre de menu « 1 ».
- Fenêtre d'affichage « 2 ».

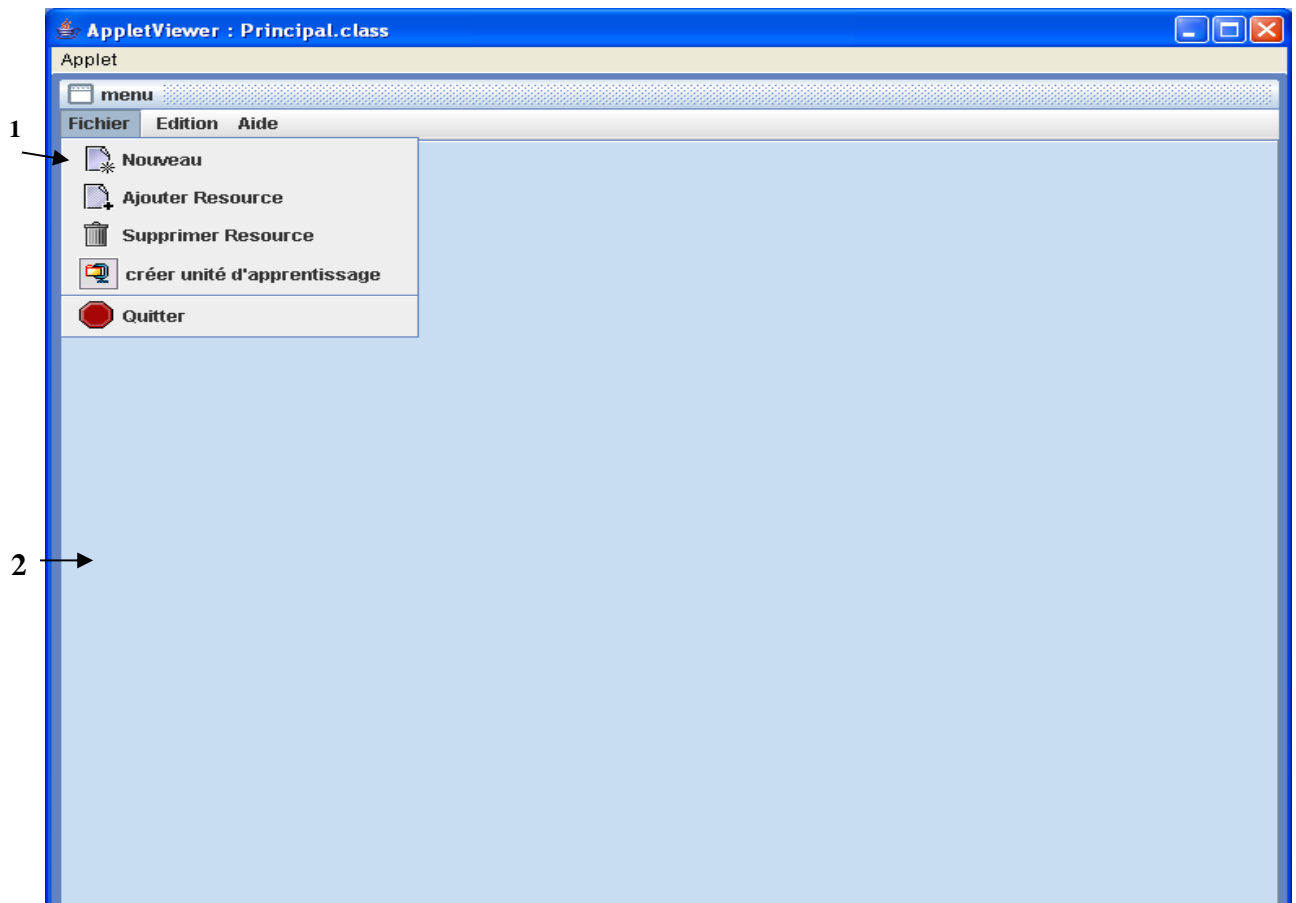


Figure26 : interface principale de l'outil.

#### 4.1. Barre de menu :

Elle contient les menus suivants :

- Fichier, avec les sous menus :
  - nouveau.
  - ajouter ressource.
  - supprimer ressource.
  - créer unité d'apprentissage.
  - quitter.
- Edition :
- .....
- Aide, qui contient :
  - Aide.
  - A propos.
  - chercher.

## **4.2. Fenêtre d'affichage :**

Cette zone est consacrée à la visualisation des différentes interfaces implémentées, grâce auxquelles l'enseignant peut créer des unités d'apprentissage, parmi lesquelles on cite :

- L'interface du questionnaire, qui est composée de deux questions, le premier concerne l'objectif de l'apprentissage, et le deuxième concerne le type du contenu pédagogique.
- L'interface qui présente le help du modèle de collaboration choisis.
- L'interface des activités, des ressources et services utilisés.
- Etc.

## **4.3. Les interface de l'outil :**

L'outil déjà développé est composé de plusieurs interfaces qui permettent à l'enseignant d'assurer plusieurs tâches, chaque tâche constitue l'une des étapes de conception et de création d'une unité d'apprentissage.

### **4.3.1. Interface Questionnaire :**

En effet, cette interface est composée de deux questions, chaque question possède plusieurs choix. La première question permet à l'enseignant de spécifier son objectif d'apprentissage « 1 », alors que la deuxième sert à déterminer le type du contenu pédagogique où le type des cours qui seront utilisés lors de l'apprentissage « 2 ».

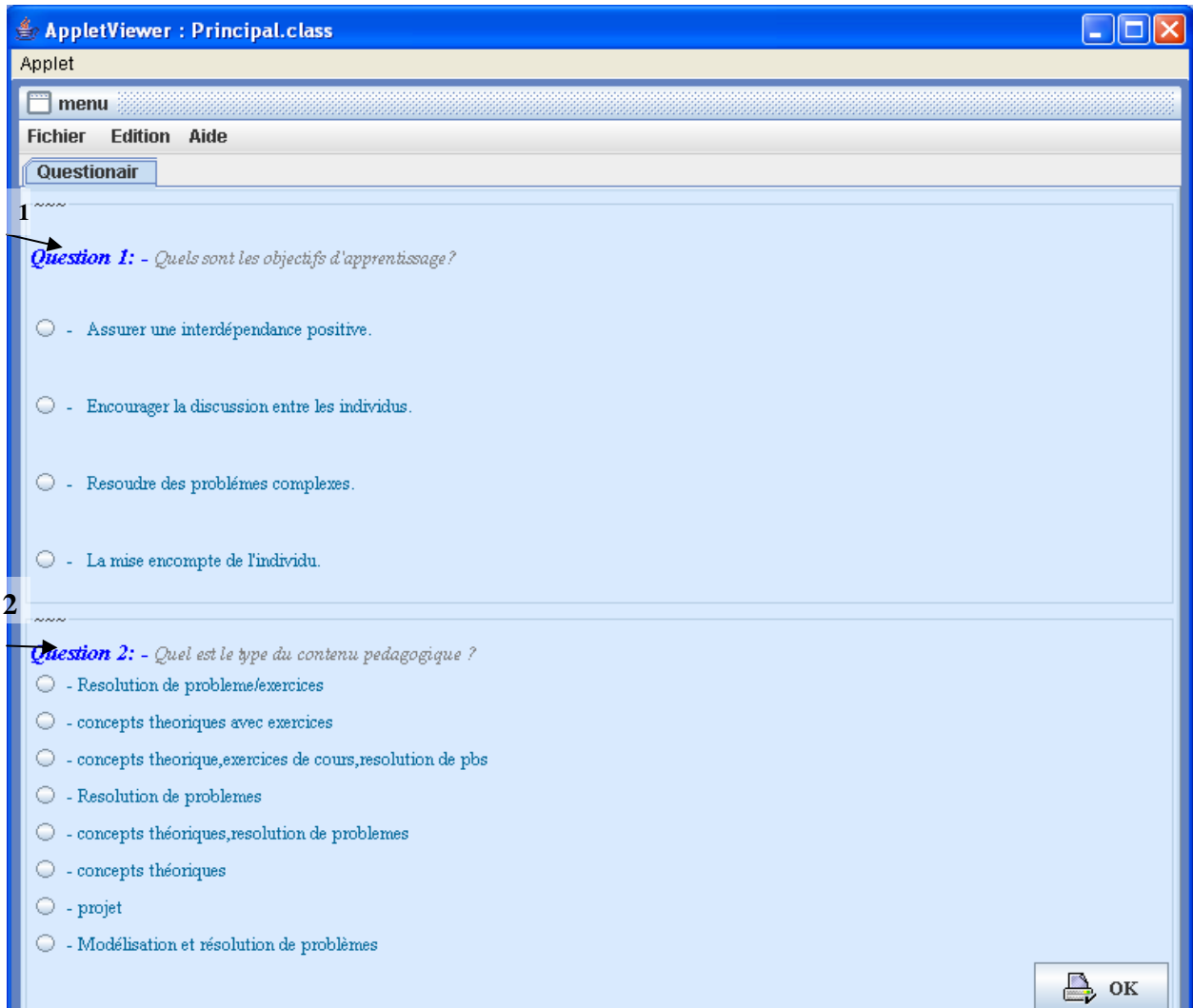


Figure27 : Interface du questionnaire.

#### 4.3.2. Interface help :

Après avoir répondu au questionnaire, l'enseignant peut voir une brève description du modèle choisi par le système suivant ses réponses, donc cette interface sert de help pour que l'enseignant connaisse le principe du modèle de collaboration. Il existe également deux help, le premier pour le modèle 'pyramid' et le deuxième pour le modèle 'jigsaw'.

La figure suivante présente le help du modèle pyramid, ce help contient une définition de ce modèle ainsi que son diagramme.

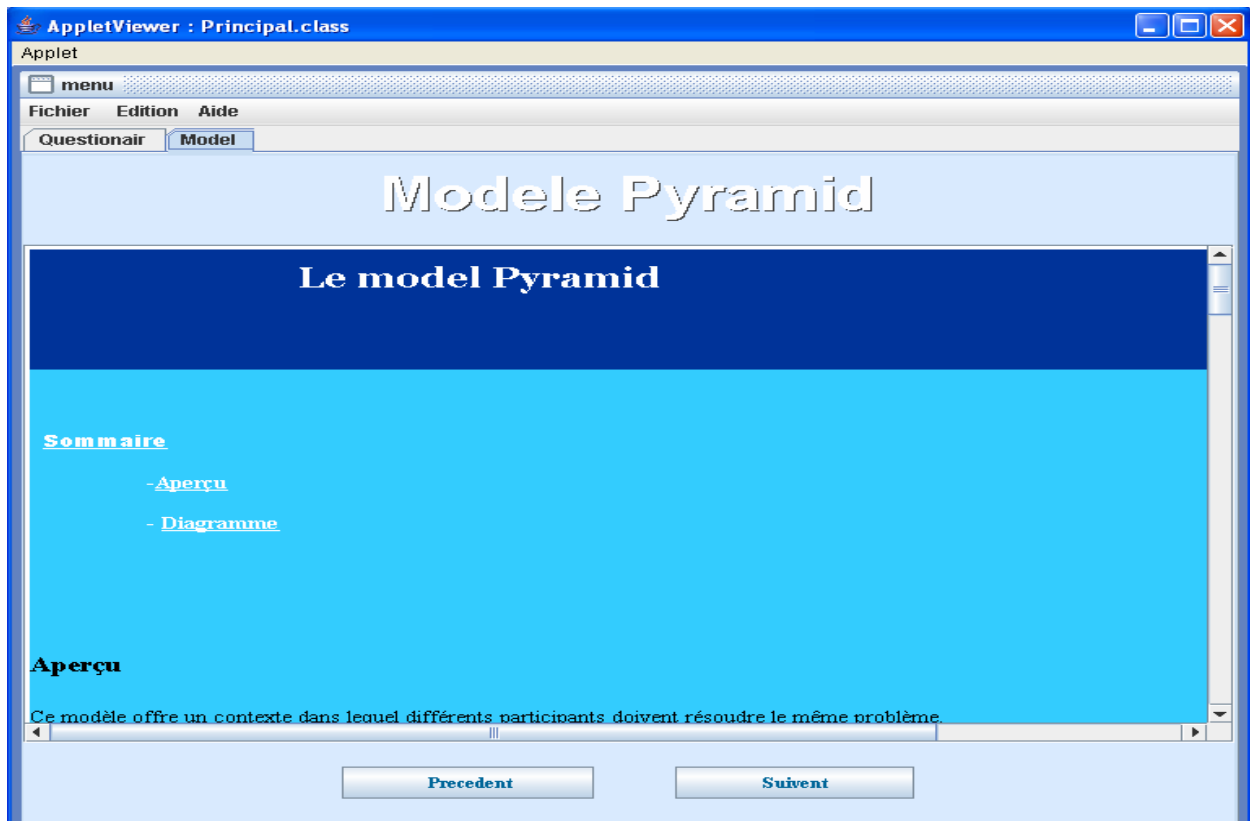


Figure28 : Interface du modèle pyramid.

Dans ce qui suit une figure illustre le help du modèle de collaboration jigsaw avec une présentation de : son principe, ses objectifs éducatifs ainsi que le diagramme qui explique mieux son principe qui est basé sur la division du problème global en sous problème, ce qui simplifie la tâche pour les étudiants.



Figure29: Interface du modèle jigsaw.

#### 4.3.3. Interface du nombre de niveaux :

Dans le cas où le modèle choisi est le modèle pyramid, l'enseignant doit ainsi choisir le nombre de niveaux, s'il s'agit de deux, trois ou quatre niveaux. Le modèle 'Jigsaw' possède un nombre constant de niveaux (trois niveaux).

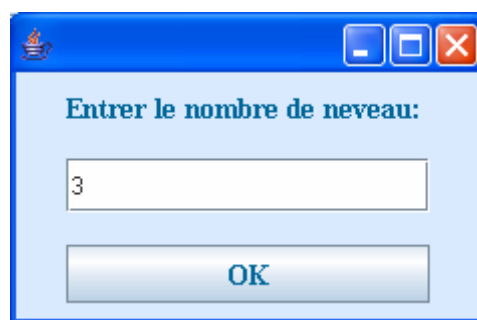


Figure30 : Interface du nombre de niveaux

#### 4.3.4. Interface d'activité :

Elle permet à l'enseignant de : donner une description complète de chaque activité, soit son activité ou l'activité de l'apprenant « 1 », de choisir les ressources « 2 » (les cours créés précédemment) et les services « 3 » nécessaires pour l'apprentissage. On a crée trois interfaces

d'activité pour le modèle 'pyramid', chacune correspond à un niveau précis, et une interface identique à celle du pyramid avec trois niveaux pour le modèle 'Jigsaw'. Dans ce qui suit une figure qui décrit un pyramid avec deux niveaux.

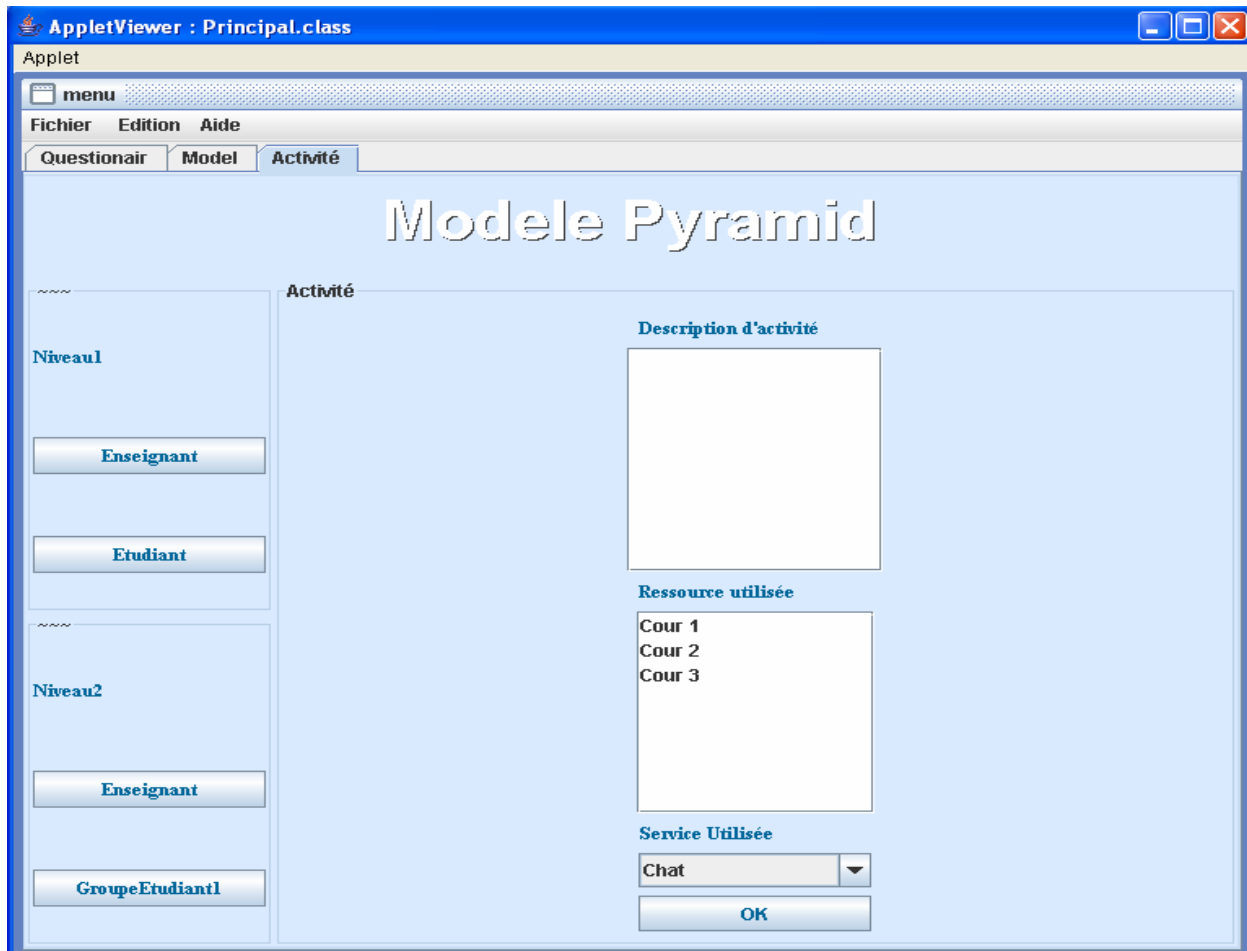


Figure 31: Interface d'activité ' pyramid avec deux niveaux'.

La figure suivante illustre le modèle pyramid avec trois niveaux, toujours deux rôles qui sont associés à chaque niveau. Au niveau 1, on trouve l'enseignant et l'apprenant, le niveau 2 comporte les rôles : enseignant et groupe d'étudiant1, les rôles présentés au dernier niveau sont : l'enseignant et groupe étudiant 2 (toute la classe).



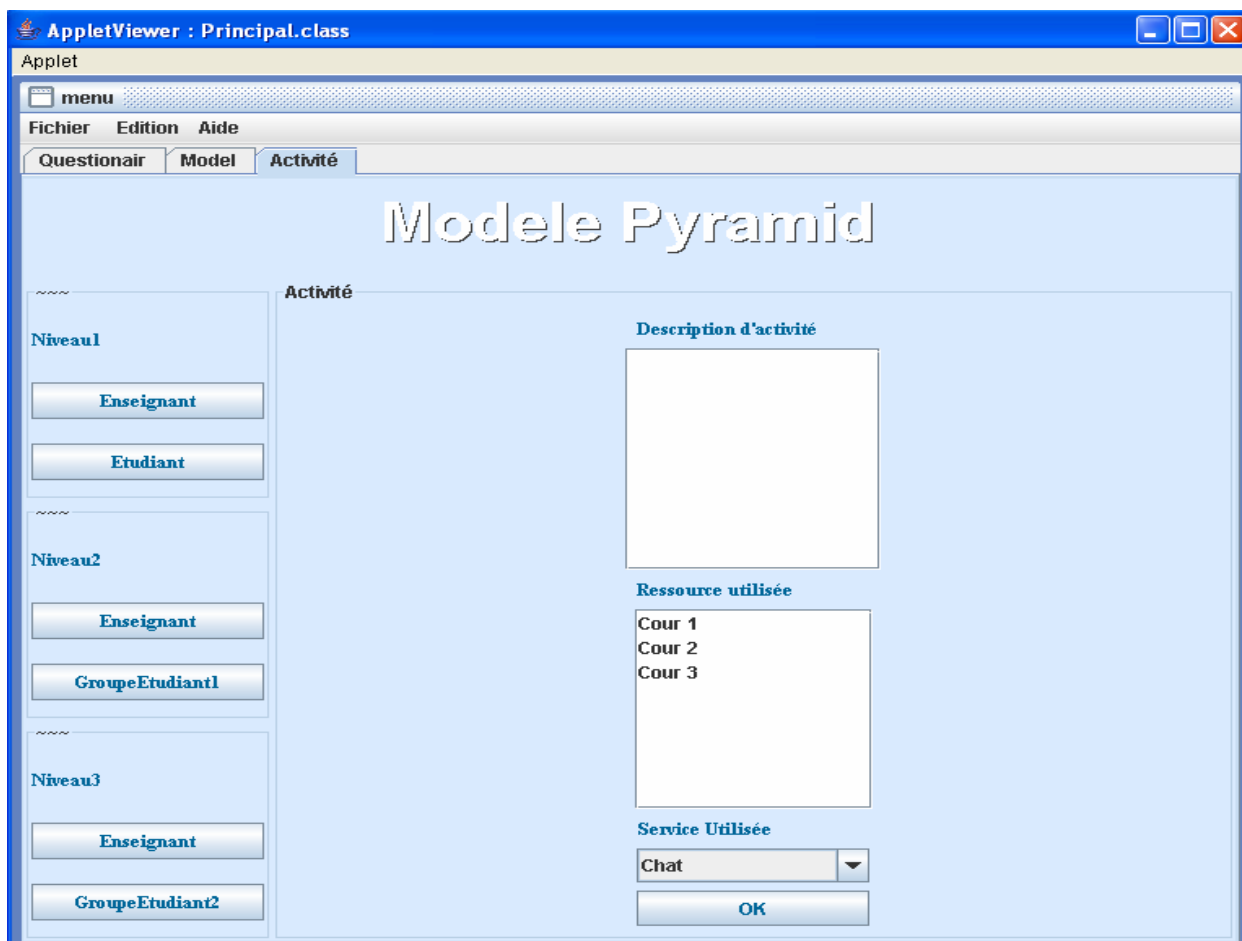


Figure32 : Interface d'activité ' pyramid avec trois niveaux'.

La figure '33' présente un Pyramid avec quatre niveaux. Pour le premier niveau, il existe deux rôles : l'enseignant et l'étudiant. Il y'a deux rôles aussi pour le deuxième niveau : l'enseignant et groupe d'étudiant1. Dans le troisième, on trouve l'enseignant et le groupe d'étudiant2. Les rôles : enseignant et groupe d'étudiant3 qui désigne toute la classe, sont associés au dernier niveau. Nous remarquons que tous les apprenants se regroupent pour construire toute la classe au dernier niveau.

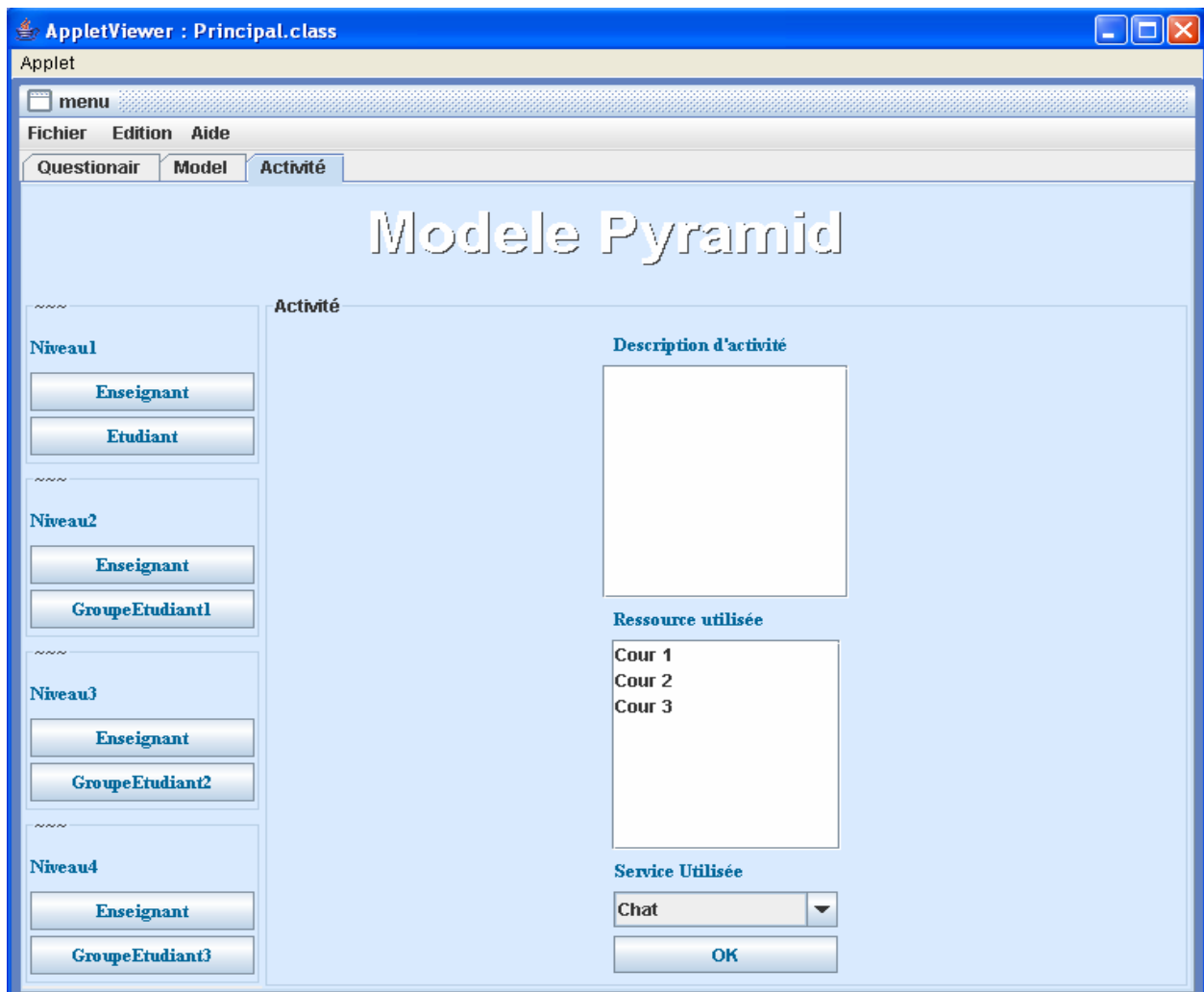


Figure33 : Interface d'activité ' pyramid avec quatre niveaux'.

#### 4.3.5. Interface Ressource :

Grâce à cette interface l'enseignant peut créer les cours qu'il veut présenter aux apprenants. . Il s'agit ici d'un certains nombre de squelettes des cours de différents types, qui sont enregistrées dans une base. Le type du squelette dans laquelle l'enseignant va mettre son cours dépend du type du contenu pédagogique (concepts théoriques, exercices,...) qu'il a choisi auparavant, il doit :

- spécifier le type du fichier qui va être crée : HTML,... « 1 »
- mettre le contenu du cours dans la zone spécifiée « 2 ».
- enregistrer le cours dans la base des modèles d'activités et de ressources « 3 ».

Le chemin de cette base est indiqué par une zone « 4 » pour que l'enseignant soit au courant de l'emplacement de ces cours, et il peut les consulter.

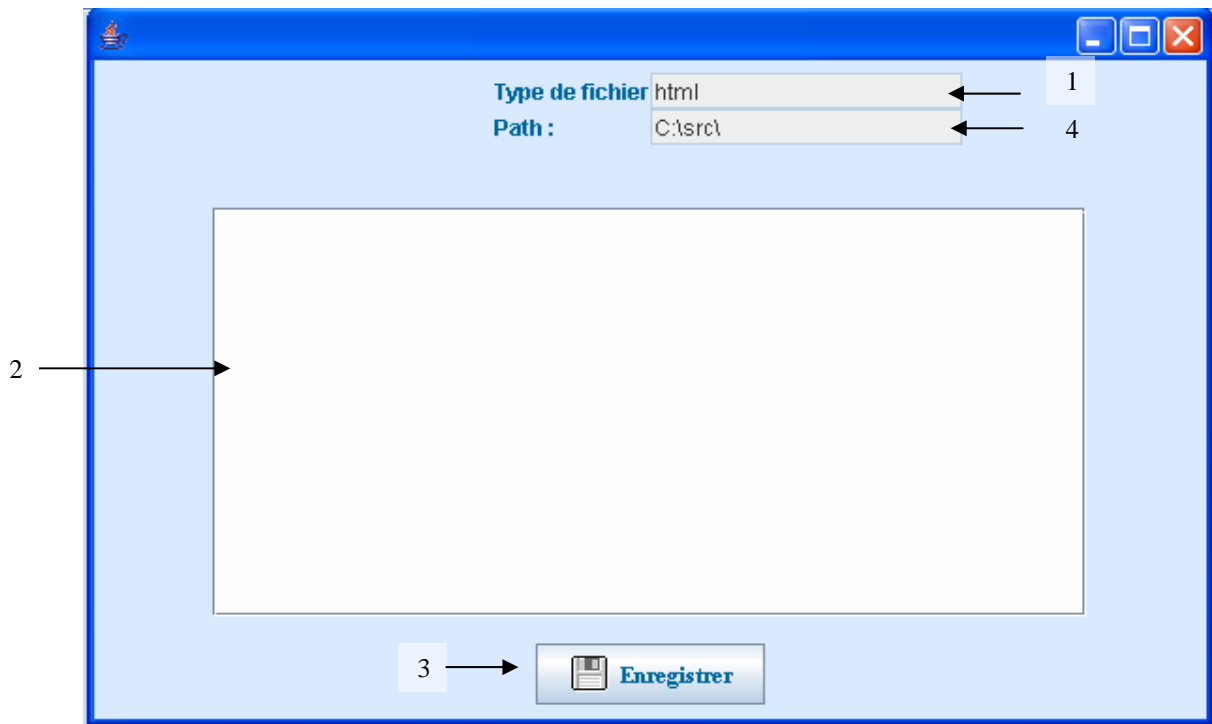


Figure34: Interface de création de ressources.

## 5. Implémentation des manifests :

Dans le chapitre précédent, on a parlé d'une base qui contient des fichiers xml, chacune décrit un scénario précis. De ce fait, on a implémenté quatre manifests, le premier sert à définir un scénario d'apprentissage qui décrit la collaboration selon le modèle Pyramid avec deux niveaux, le deuxième correspond à un scénario avec trois niveaux, le troisième avec quatre niveaux, et le dernier manifest décrit un scénario qui adopte le modèle : « jigsaw ».

Nous expliquons ainsi les différentes parties (définies par IMSLD) du manifest avec trois niveaux.

### Rôles :

Dans le cas d'un pyramid avec trois niveaux, on a quatre rôles : l'étudiant, le groupe d'étudiant, la classe et l'enseignant.

<rôles>

< learner identifier="R-etud">

    <title>etudiant</title>

</ learner >

< learner identifier="R-groupe">

    < title> groupe etudiant</title>

</ learner >

< learner identifier="R-clase">

    <title>classe</title>

```
</ learner >
  <teacher identifier="R-profesor">
    <title>enseignant</title>
```

```
</teacher>
```

```
</rôles>
```

### Activités:

En effet, il existe trois types d'activité : activité d'apprentissage, activité structurée qui peut contenir plusieurs activités d'apprentissage et le dernier type présente l'activité de support.

Les activités des deux premiers types sont réalisées par l'étudiant alors que l'activité de support est pour l'enseignant.

```
<activities >
```

```
  < structure-activity identifier="activite-N1">
    < learning-activity identifier="Etude individuelle">
      < title>Etude individuelle</ title>
      </ learning-activity>
      < learning-activity identifier="Proposition solution1">
        < title>Proposition solution 1</ title>
      </ learning-activity>
    </ structure-activity>
```

```
<structure-activity identifier="activite-N2">
  <learning-activity identifier="Discussion">
    < title>Discussion</title>
    <imsld:activity-description> </imsld:activity-description>
  </imsld:learning-activity>
  <imsld:learning-activity identifier="Proposition solution2 ">
    <imsld:title>Proposition solution2 </imsld:title>
    <imsld:activity-description> </imsld:activity-description>
  </imsld:learning-activity>
```

```
</imsld:structure-activity>
```

```
<structure-activity identifier="activite-N3">
  <learning-activity identifier="Discussion Finale">
    <title>Discussion Finale</title>
    <activity-description></activity-description>
  </learning-activity>
  <learning-activity identifier="Proposition solution Finale">
```

```

    <title>Proposition solution Finale</title>
    <activity-description></activity-description>
  </learning-activity>
</structure-activity>
< support-activity identifier="activite-control">
  < title>Activite control</ title>
  <activity-description></activity-description>
</ support-activity>
</activities>

```

### Méthodes :

La méthode de la spécification IMSLD (voir chap 3 section 3.4) est constitué d'un ou plusieurs play. Un play est constitué d'un ou plusieurs acts. Un act fait références à un ou plusieurs rôle-part.

```

<method >
<play identifier="PLAY-Pyramid3N">
  <title>Play</title>
  <act identifier="act-1">
    <title>Level 1</title>
    <role-part identifier="RP-S1">
      <title>RP_S</title>
      <role-ref ref="R-etud" />
      <structure-activity-ref ref="activite-N1" />
    </role-part>
    <role-part identifier="RP-T">
      <title>RP_T</title>
      <role-ref ref="R-profesor" />
      <support-activity-ref ref="activite-control" />
    </role-part>
  </act>
  ...
</play>
</method >

```

**Ressource :**

Deux types de ressources sont utilisées : le premier présente les documents qui peuvent être consultés par l'étudiant, ces documents peuvent contenir des concepts théoriques, des exercices,... le deuxième type correspond aux services utilisés pour la communication.

```
<resources>
```

```
<resource type=" webContent" id="cour1" href="cour1.html" />
```

```
<resource type=" service" id="chat" href="chat.jar" />
```

```
...
```

```
</resources>
```

**6. L'implémentation du service chat :**

Le service chat est un exemple de service servant à établir une communication entre les apprenants, ou entre les apprenants et l'enseignant. Nous avons adopté le modèle des composants fractals pour implémenter ce service.

Malgré la richesse de la bibliothèque java, elle ne supporte pas la programmation des composants Fractal. Pour implémenter ces composants en utilisant le langage de programmation java, on est obligé d'intégrer une API nommée : « **Julia** ». Cette API contient des classes permettant de : créer, lancer, stopper, détruire,...un composant Fractal.

Dans le chapitre de conception, nous avons présenté l'architecture du composant développé : chat, et nous avons dit que le composant chat est un composant composite qui contient deux sous composants, le premier est un client et le deuxième est un serveur. L'implémentation de ce composant se traduit par :

- La définition du type du composant racine en spécifiant ses interfaces. Il possède deux interfaces externes. Pour chaque interface, on doit déterminer : son nom, son rôle (client ou serveur), sa signature, sa contingence ou si ses fonctionnalités doivent être disponibles ou non et sa cardinalité, c-à-d si elle est liée avec une ou plusieurs autres interfaces.

```
ComponentType rType = tf.createFcType (new InterfaceType [] {
    tf.createFcItfType("c", "Client", false, false, false),
    tf.createFcItfType("s", "Serveur", false, false, false)
});
```

- La définition du type du sous composant client avec son interface qui possède : le nom 'c', la signature ou le nom java associé à cette interface 'Client', le rôle 'false' qui indique le rôle

client,contingence'false',c'est-à-dire cette interface est optionnelle, et la cardinalité 'false' qui indique la liaison avec une seule interface ou le type singleton.

```
ComponentType cType = tf.createFcType(new InterfaceType[] {
    tf.createFcItfType("c", "Client", false, false, false)});
```

- La définition du sous composant serveur qui signifie la description de son interface interne.

```
ComponentType sType = tf.createFcType(new InterfaceType[] {
    tf.createFcItfType("s", "Serveur", false, false, false)});
```

- La création du composant racine en définissant : son nom'rComp', son type'composite', et le nom de la classe qui implémente les services offerts par ses interface (dans ce cas 'null', car il n'existe pas une classe d'implémentation).

```
rComp =cf.newFcInstance(«rComp », « composite », « null ») ;
```

- La création des deux sous composants : client et serveur, de la façon suivante :

```
Component cComp=cf.newFcInstance(cType,"primitive", "ClientImplement");
Component sComp = cf.newFcInstance(sType, "primitive", "ServeurImplement");
```

- L'ajout des deux sous composant dans le composant racine en utilisant l'instruction suivante :

```
Fractal.getContentController(rComp).addFcSubComponent(cComp);
Fractal.getContentController(rComp).addFcSubComponent(sComp);
```

- L'établissement de la liaison entre les interfaces du composant racine et celles des sous composants :

```
Fractal.getBindingController(rComp).bindFc("c", cComp.getFcInterface("c"));
Fractal.getBindingController(rComp).bindFc("s", sComp.getFcInterface("s"));
```

- Le lancement du composant racine par l'instruction :

```
Fractal.getLifeCycleController(rComp).startFc() ;
```

La fenêtre principale de ce composant est illustrée par la figure suivante :

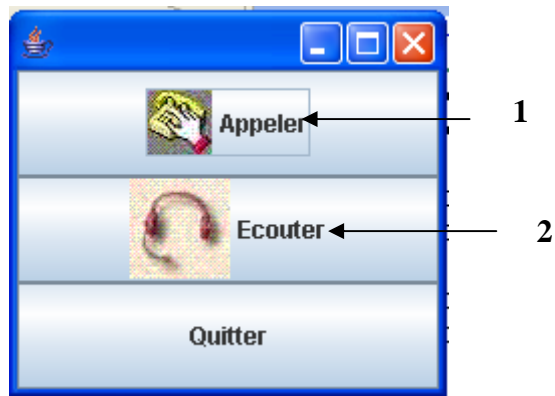


Figure35 : la fenêtre principale du composant chat.

-« 1 », pour afficher la fenêtre du client. La zone d'affichage permet d'afficher les messages échangés entre le serveur et le client « 3 ». Le client met le message à envoyer dans un champ spécifique « 4 » :

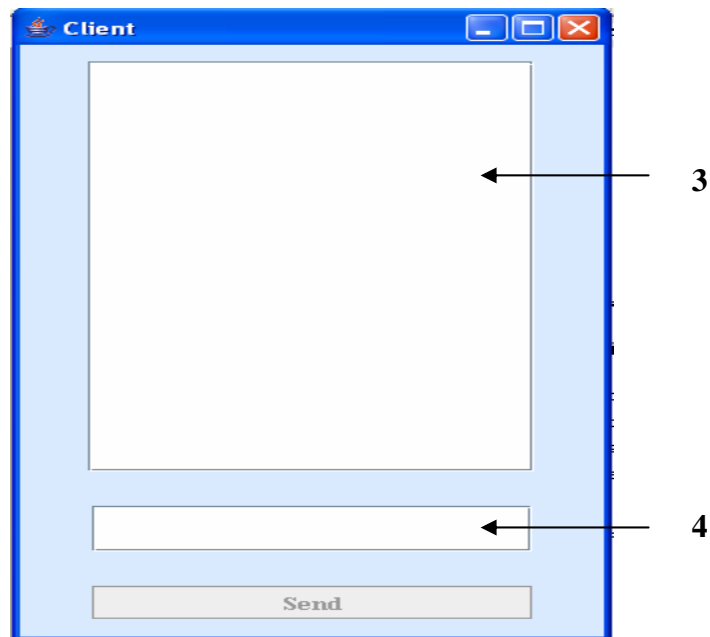


Figure36 : fenêtre du client.

-« 2 » sert à afficher la fenêtre du serveur qui possède la même structure de la fenêtre client. Cette fenêtre doit être lancée avant la fenêtre du client, pour que ce dernier puisse envoyer ses messages au serveur:



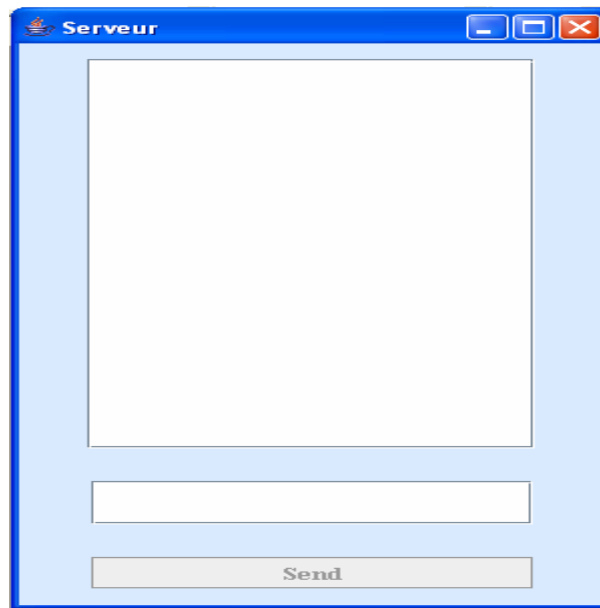


Figure37 : Fenêtre du serveur.

## 7. Exécution d'une unité d'apprentissage par CopperCore :

Le résultat de notre outil est une unité d'apprentissage créée pour être utilisée par l'apprenant, pour cela il doit utiliser l'un des environnements qui permettent l'exécution des unités d'apprentissage, on prend comme exemple l'environnement : copperCore, et un exemple d'une unité d'apprentissage qui décrit un scénario d'apprentissage collaboratif en utilisant le modèle d'activité pyramid constitué de deux niveaux. On a utilisé pour créer cette unité, l'outil collage car les UOL créés par notre outil ne peuvent pas être encore exécutés. L'exécution de cette unité d'apprentissage par l'environnement copperCore se fait en suivant plusieurs étapes. Ces étapes seront détaillées par la suite :

-la première étape consiste à choisir une unité d'apprentissage (choisir son chemin) « 1 », puis faire publier et valider cette UOL « 2 » dans le but de détecter les éventuelles erreurs, comme le montre la figure ci-dessous :

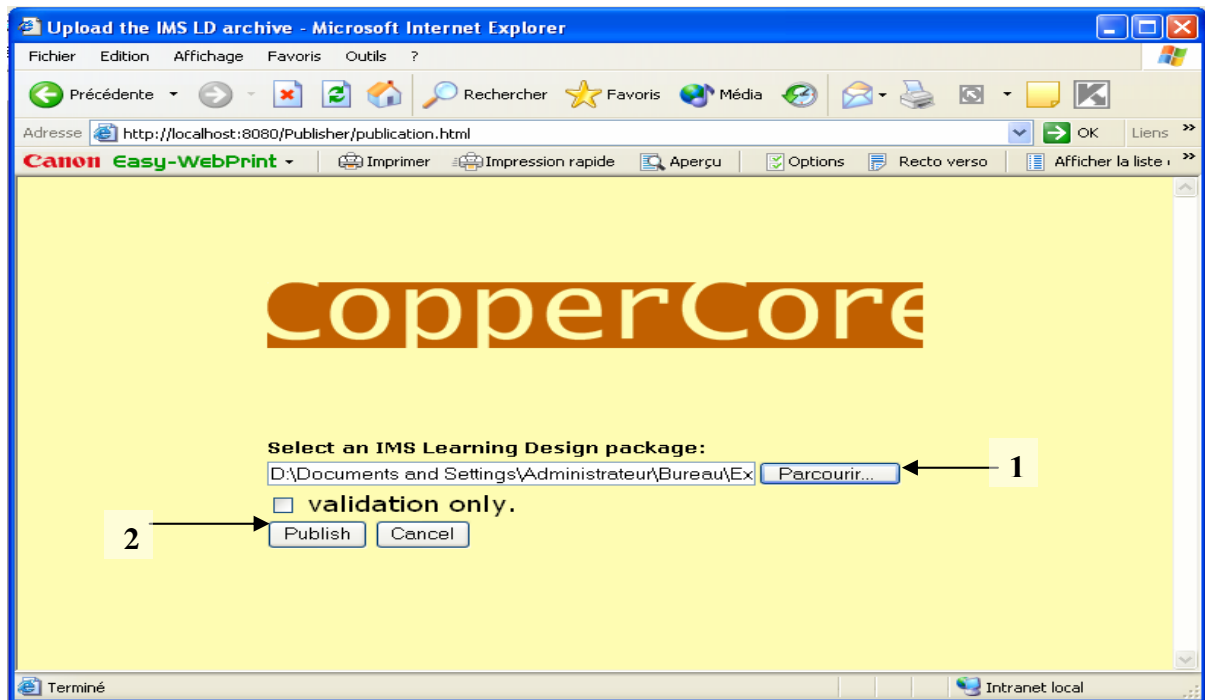


Figure38 : la publication de l'UOL.

-Dans la deuxième étape, on doit créer des utilisateurs et leurs associer les rôles définis dans l'unité d'apprentissage. La figure suivante montre la fenêtre qui contient une liste des utilisateurs, et l'utilisateur sélectionné : 'etud', auquel l'UOL qu'on a crée est associée :

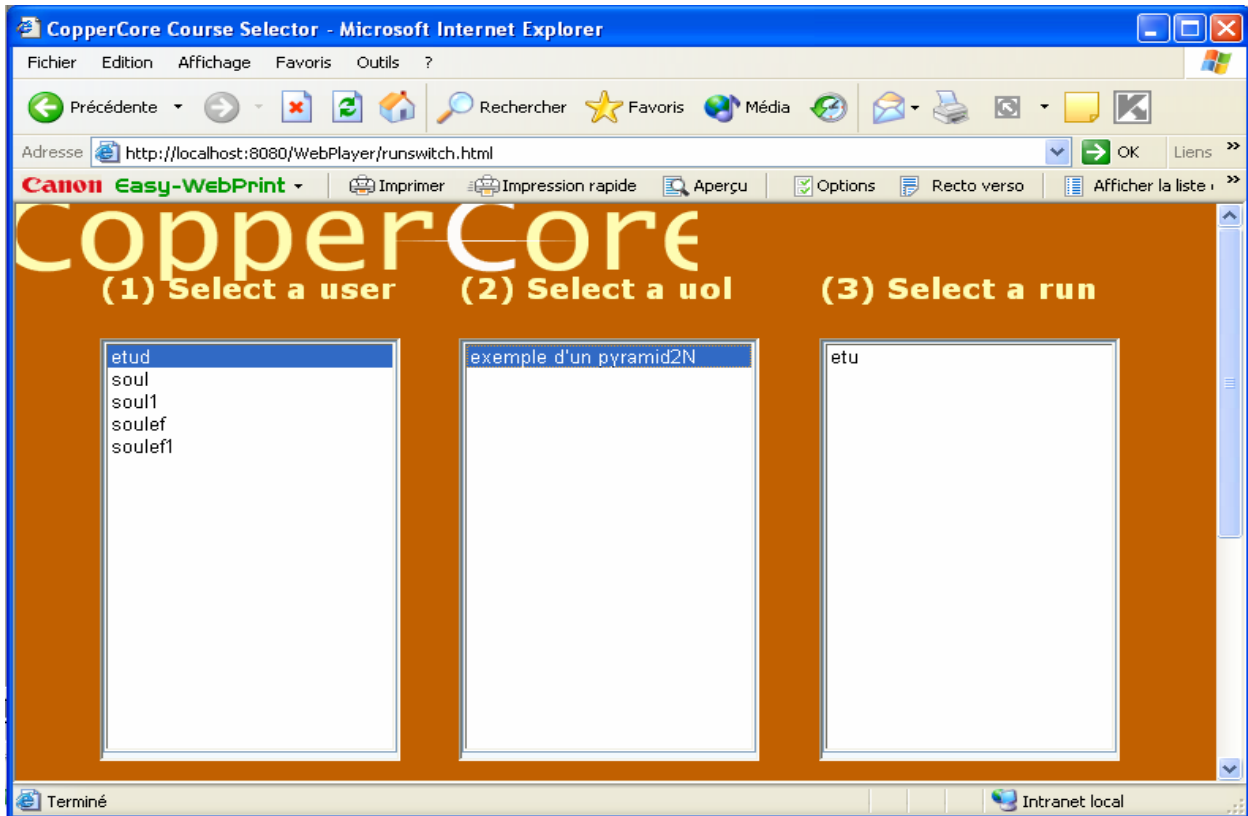


Figure39 : interface des utilisateurs.

-l'apprenant peut ainsi voir les différentes activités « 1 », les ressources et les services « 2 » décrits par l'enseignant, comme il peut les consulter à partir de la zone créée pour ce but « 3 ». La figure suivante montre le choix de l'apprenant d'utiliser le service chat :

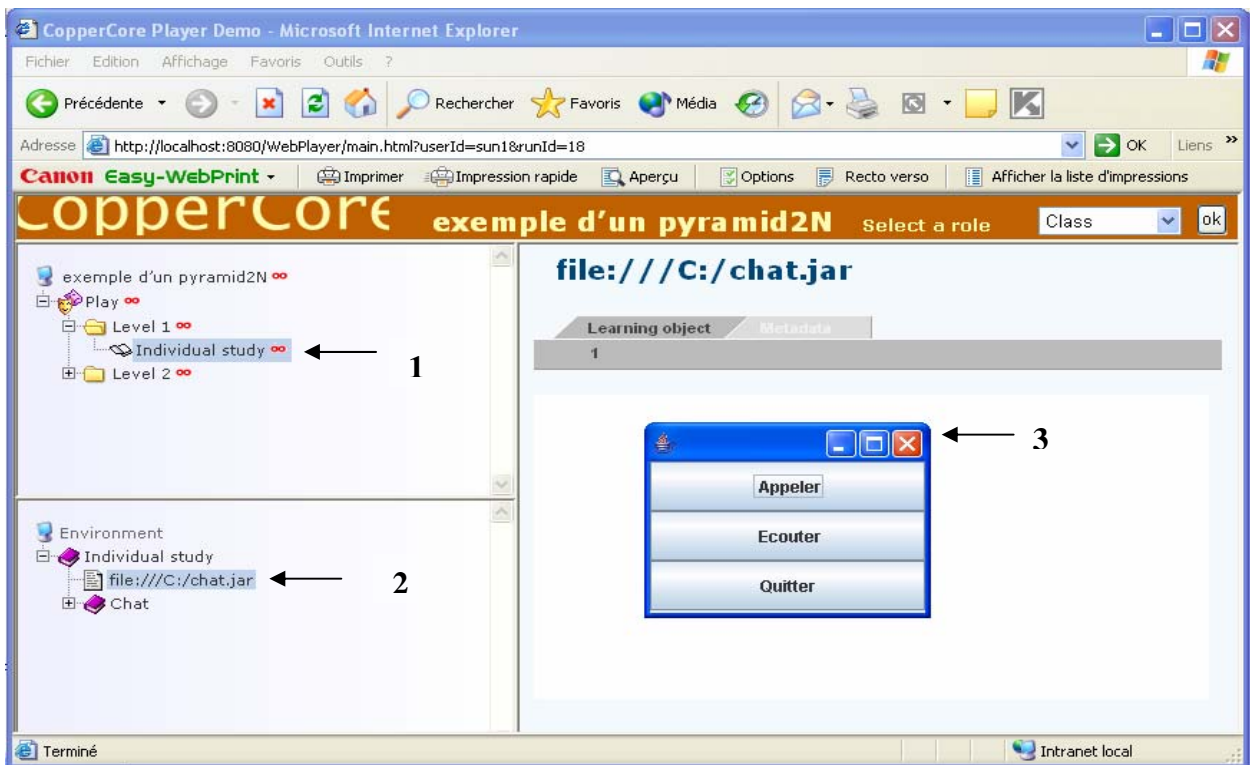


Figure40 : interface d'exécution de l'UOL.

-Après le lancement du service chat l'apprenant peut ainsi l'utiliser en envoyant ses différentes questions à l'enseignant ou discuter avec un autre apprenant, comme l'illustre la figure suivante :

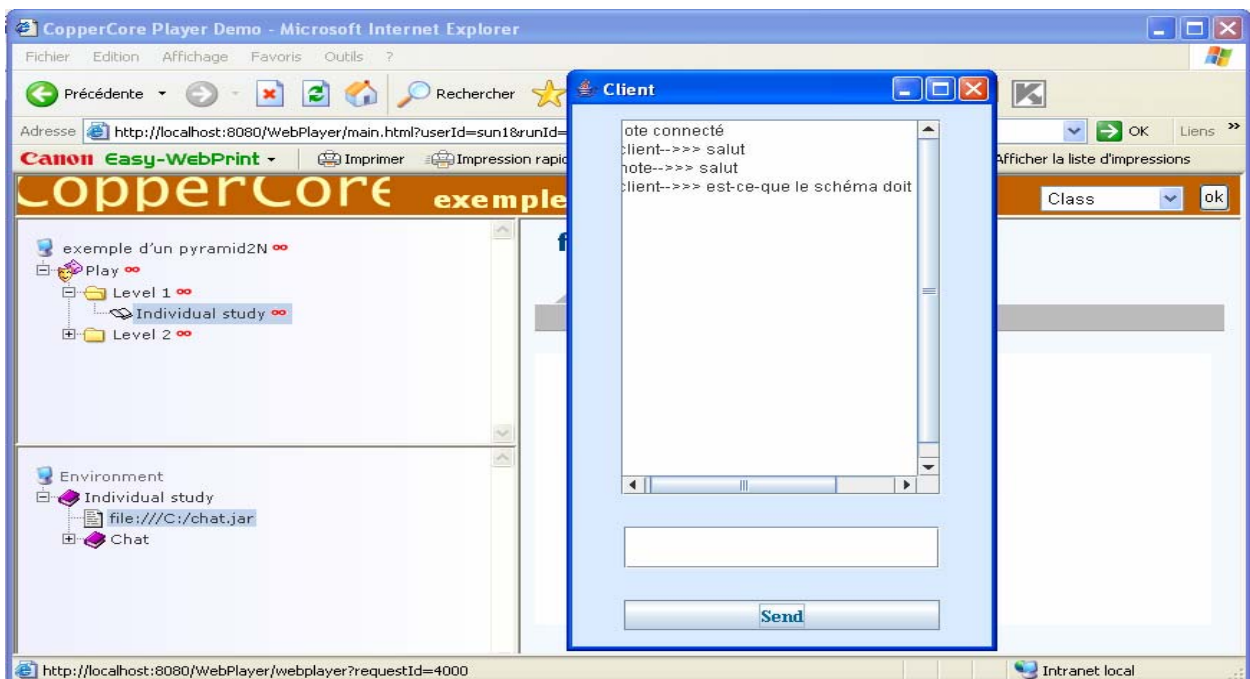


Figure 41: L'utilisation du service Chat.

## **8. Conclusion :**

Dans le but de valider l'approche conceptuelle que nous avons proposée, nous avons implémenté un outil permettant de guider les enseignants dans la conception des situations d'apprentissage collaboratives qui se traduit par la création des unités d'apprentissage utilisables par les apprenants. Dans ce contexte, nous avons présenté dans ce chapitre le schéma fonctionnel de l'outil, les outils adoptés pour son implémentation, la syntaxe proposée pour les manifests utilisés, les différentes interfaces qui constituent l'outil développé, la mise en œuvre du composant chat (un exemple de service de communication), et les différentes étapes suivies pour l'exécution d'un exemple d'une unité d'apprentissage via l'environnement d'exécution : CopperCore.

## **Chapitre 6**

### **Exemple d'une unité d'apprentissage dédiée à la collaboration**

## 1. Introduction :

Notre travail a pour but de concevoir et modéliser des situations d'apprentissage collaboratif. Le résultat de ce travail est effectivement une entité qui peut être utilisable par les apprenants en l'exécutant par un environnement d'exécution, cette entité est appelée : unité d'apprentissage (Unit Of Learning). Dans ce chapitre, nous présentons les différentes étapes de création d'un exemple réel d'une unité d'apprentissage en utilisant l'outil que nous avons développé.

## 2. Un exemple d'unité d'apprentissage :

Le texte suivant permet de décrire un exemple d'un scénario simplifié :

- L'objectif recherché par l'enseignant est d'encourager la discussion entre les étudiants.
- Un groupe d'apprenants doivent lire attentivement un cours sur l'architecture des ordinateurs, puis ils doivent proposer un schéma qui présente la structure d'un ordinateur.
- Le travail de l'enseignant consiste à répondre aux questions éventuellement posées par les apprenants et vérifier la solution finale.

Le modèle d'activité approprié d'après l'objectif de l'enseignant est le modèle : Pyramid. Le diagramme d'activité UML suivant permet de représenter ce scénario en utilisant un pyramid avec deux niveaux au titre d'exemple.

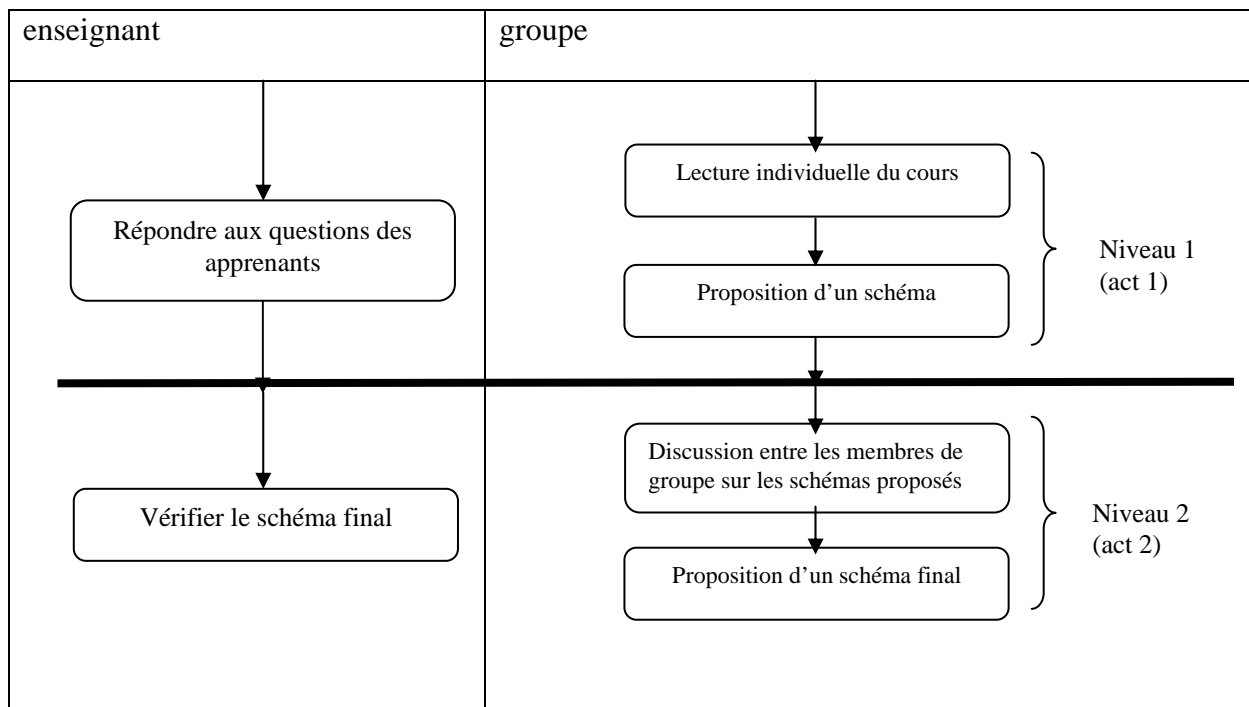


Tableau3: Diagramme d'activité UML représentant un exemple de scénario.

Il s'agit ici d'un seul *play* constitué de deux *acts*. Chaque *act* représente un niveau. Nous détaillons maintenant les étapes que doivent être suivies par l'enseignant afin de créer une unité d'apprentissage représentant le scénario défini précédemment par l'outil implémenté :

- La première étape consiste à répondre au questionnaire proposé à l'enseignant dans le but de choisir l'objectif qui cherche à atteindre ainsi que le modèle du cours que doivent être présentés aux étudiants. Selon l'exemple précédent, l'objectif de l'enseignant est : encourager la discussion entre les apprenants et le type du cours est : concepts théoriques.

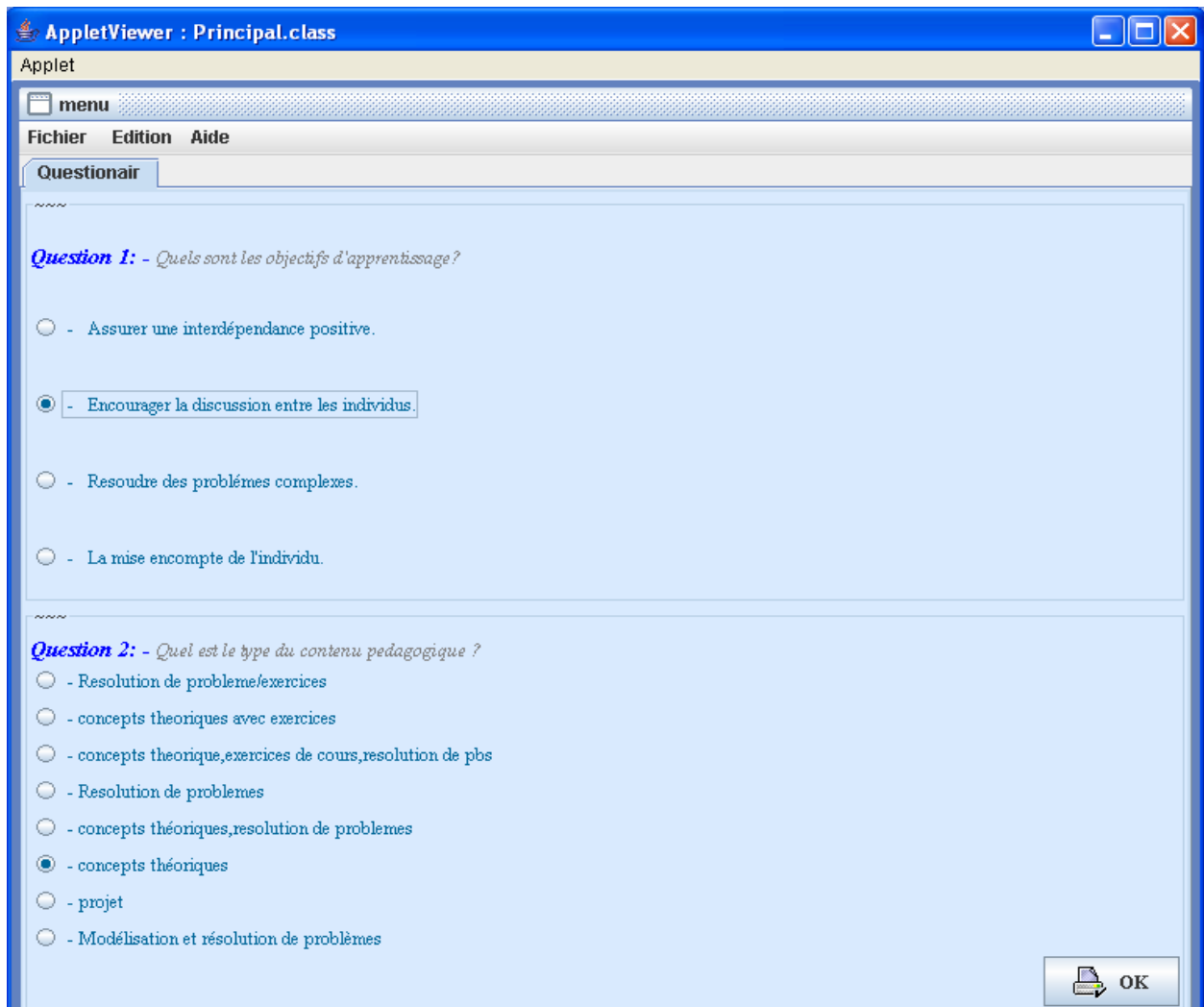


Figure42 : Les réponses choisies par l'enseignant.

-Après avoir répondu au questionnaire, un help décrivant le modèle sélectionné s'affiche dans l'interface, le help affiché est celui du pyramid.

-Comme le modèle choisi ici est le modèle pyramid, la deuxième étape concerne le choix du nombre de niveaux de ce pyramid, par exemple : deux niveaux.

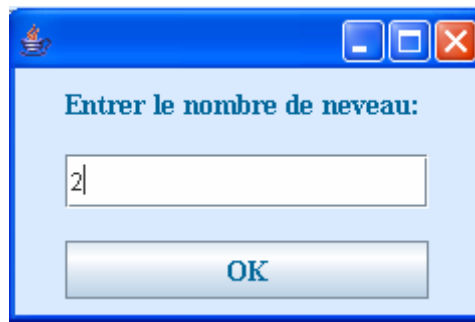


Figure43 : choix du nombre de niveaux.

-Dans la troisième étape, l'enseignant doit définir plusieurs informations importantes telles que :

- Une description détaillée de l'activité que doit être achevée par chaque rôle.
- Le ou les cours que doivent être consultés par les apprenants lors de l'apprentissage.
- Les services qui doivent être utilisés par les étudiants pour communiquer.

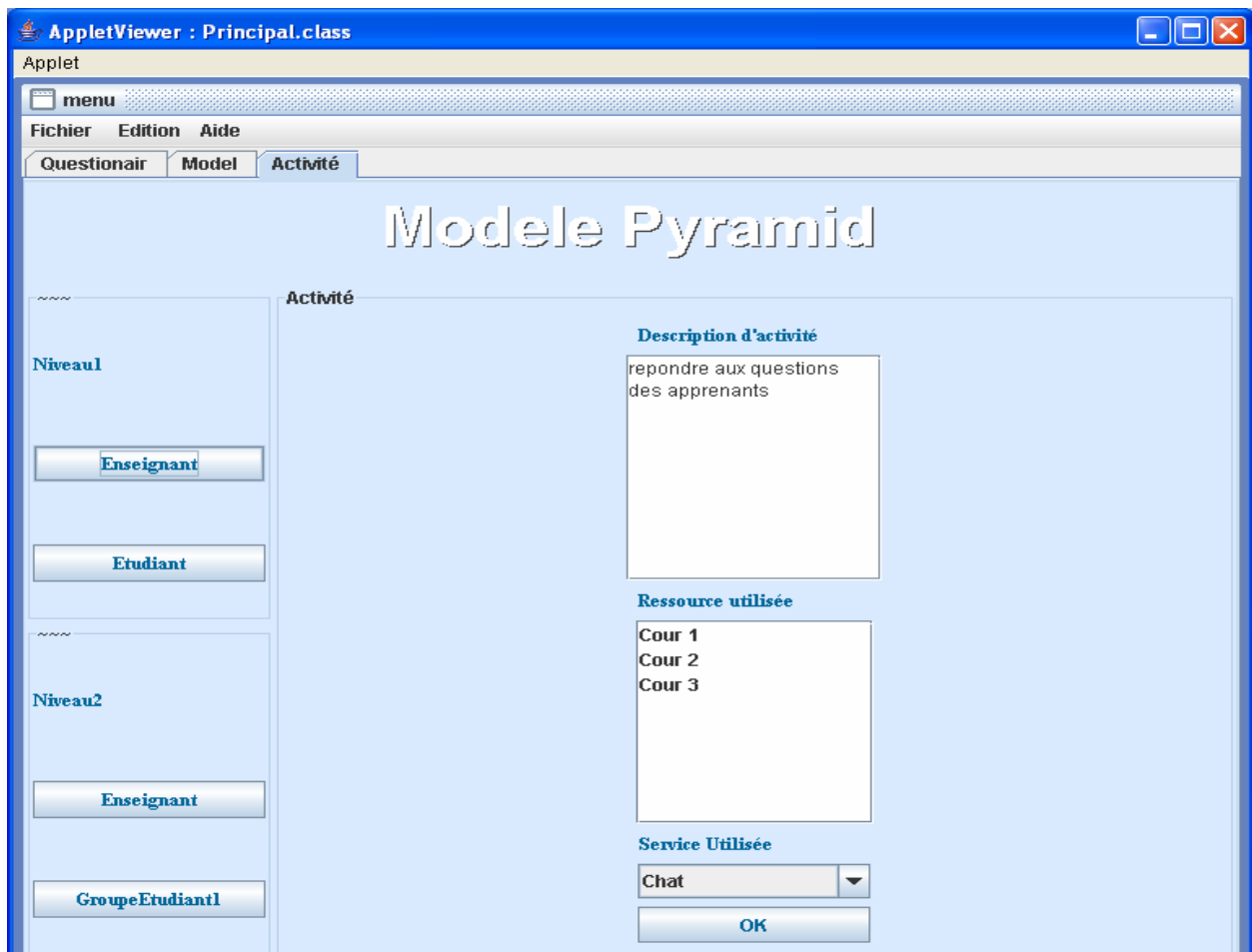


Figure44 : Définition des informations nécessaires pour le rôle : enseignant (niveau1)

Cette figure montre la description de l'activité de l'enseignant au niveau1, cette description est la suivante : répondre aux questions des apprenants. On remarque ici que l'enseignant n'a pas



choisi une ressource ou cours, car les ressources prises comme exemple sont seulement dédiées aux étudiants, mais il a choisi le service chat pour communiquer avec les apprenants.

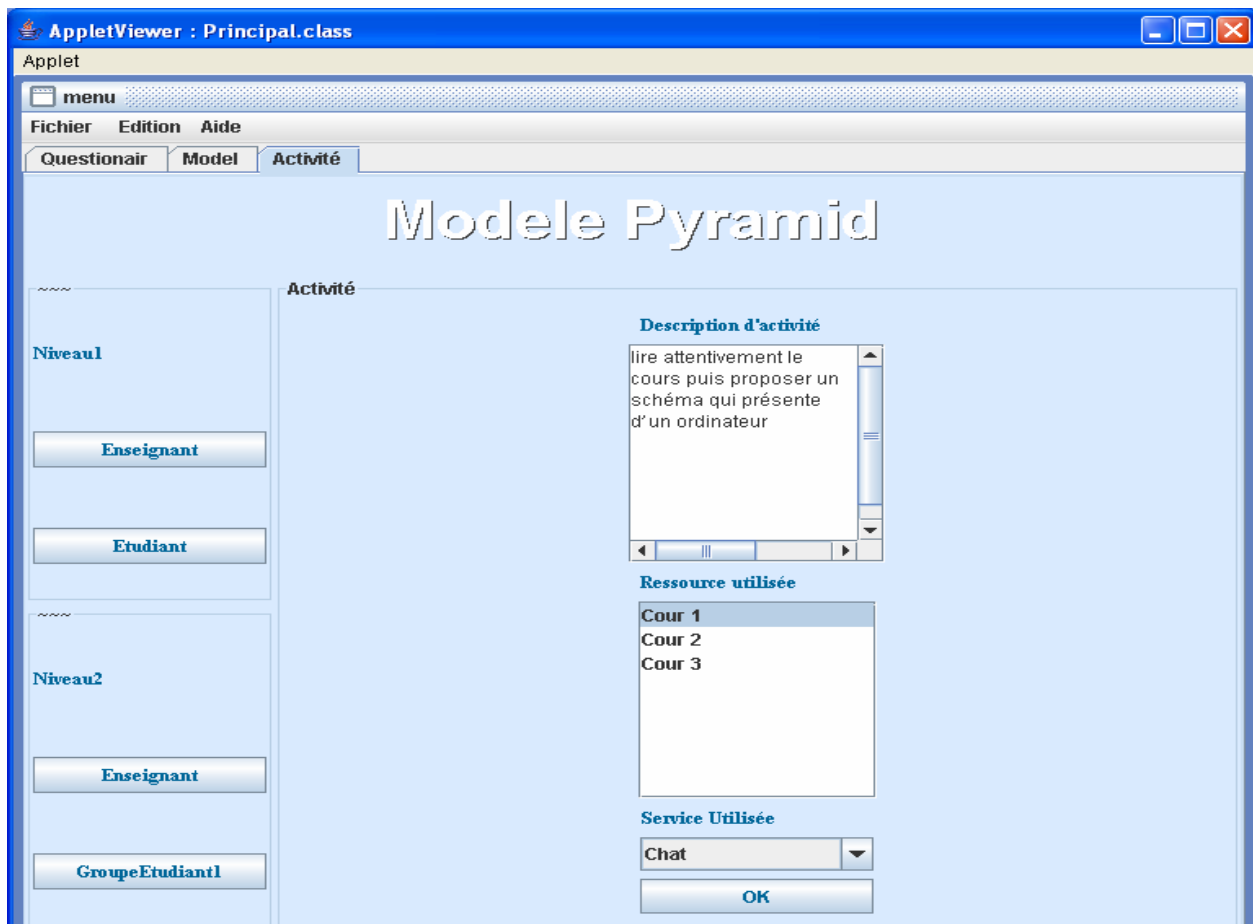


Figure45: définition des informations nécessaires pour le rôle : étudiant (niveau1).

Cette figure montre la description donnée par l'enseignant pour l'activité de l'étudiant au niveau 1 qui est : lire attentivement le cours puis proposer un schéma qui présente l'architecture d'un ordinateur, le choix du cours 1 qui sera utilisé comme ressource et le service chat.

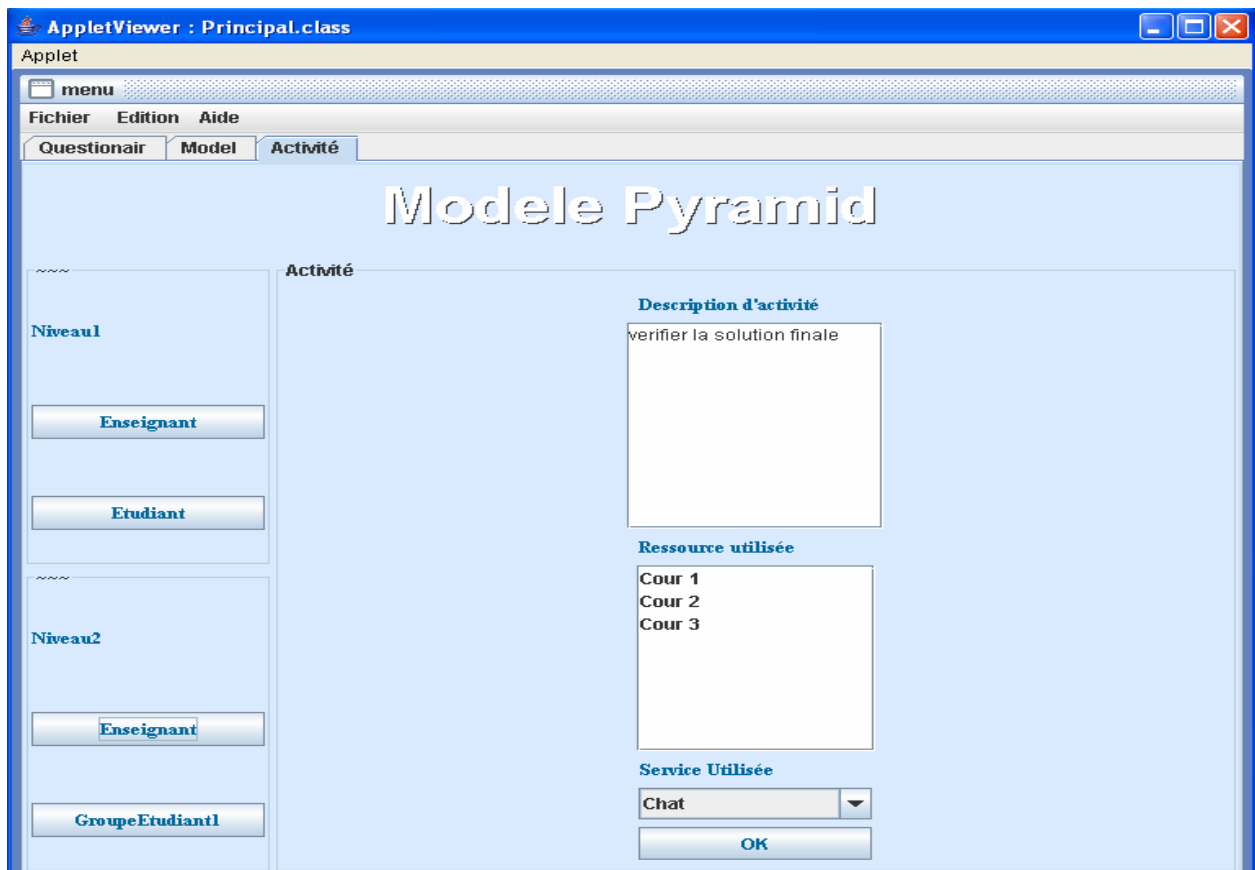


Figure46: définition des informations nécessaires pour le rôle : enseignant (niveau2)

Cette fenêtre présente l'association de la définition : vérifier la solution finale, à l'activité de l'enseignant au niveau 2.

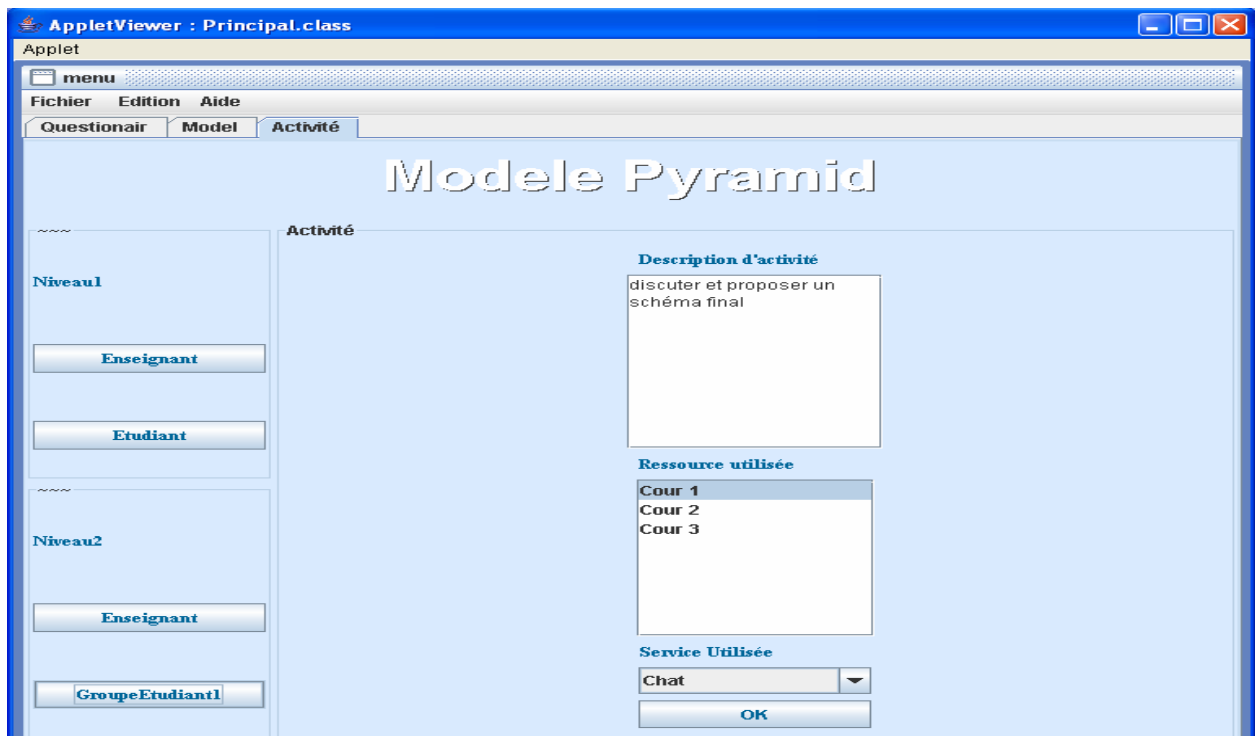


Figure 47: définition des informations nécessaires pour le rôle : groupe étudiant (niveau2)

La 'figure48' illustre la description : discuter et proposer un schéma final, donné par l'enseignant à l'activité de tous les étudiants qui constituent la classe entière au niveau 2.

Ces informations seront ajoutées dans le manifest préexistant dans la base des manifests et qui correspond au modèle pyramid, c'est ce qu'on appelle particularisation du manifest.

-Après la définition des informations mentionnées précédemment, la quatrième étape présente la création du cours nécessaire pour l'apprentissage. Pour cela, une zone de saisie s'affiche afin de permettre à l'enseignant de mettre son cours dedans. La figure suivante illustre la création d'un cours d'architecture des ordinateurs.

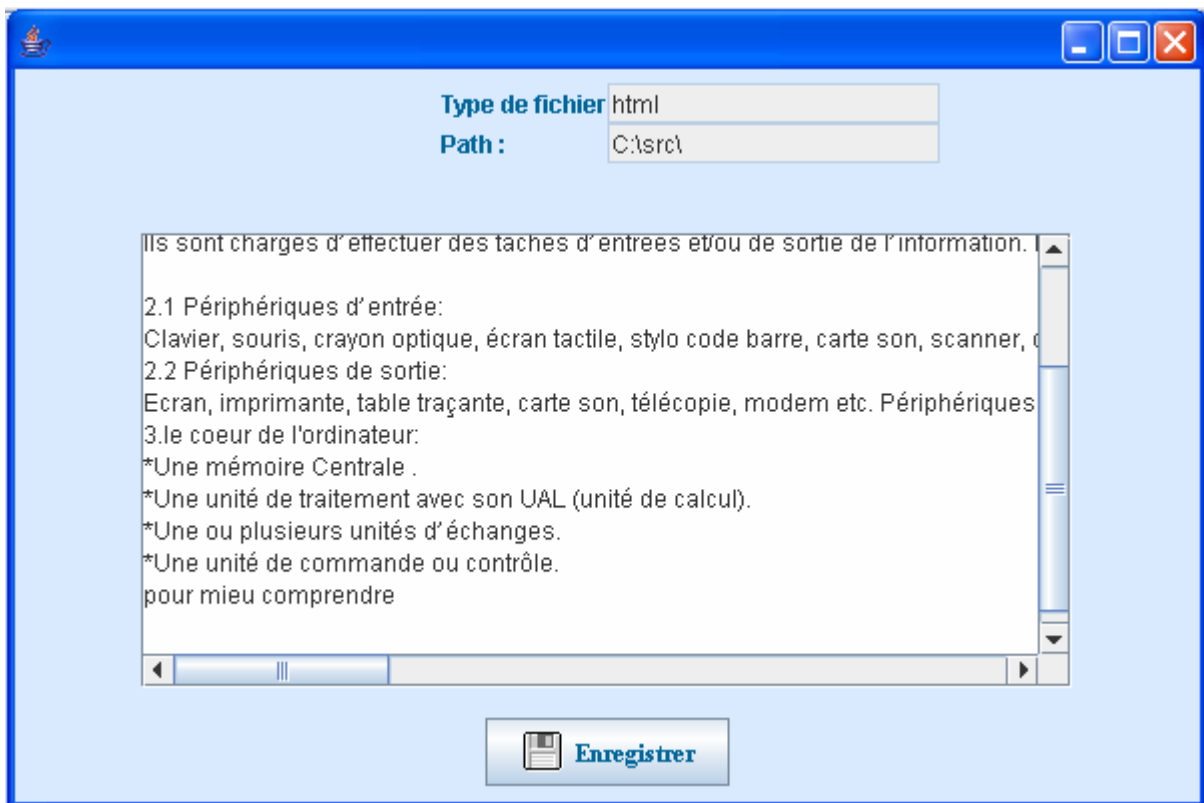


Figure48 : Création du cours d'architecture.

- La dernière étape présente la création de l'unité d'apprentissage. Lorsqu'on clique sur le menu 'créer unité d'apprentissage', le manifest définissant ce scénario sera enregistré avec les ressources et les services identifiés par l'enseignant (dans ce cas : cours 1 et le service chat) dans un répertoire compressé.

En effet, la création d'une unité d'apprentissage n'est pas la dernière étape dans le processus de l'apprentissage, car pour que les étudiants peuvent apprendre et réalisent les activités qui leurs sont affectées par l'enseignant, ils doivent exploiter cette unité. L'exploitation d'une unité d'apprentissage se traduit par son exécution en utilisant un environnement d'exécution tel que :

Coppercore (les étapes d'exécution de cette unité d'apprentissage sont présentées dans le chapitre de l'implémentation).

Après la particularisation du manifest représentant le scénario de l'exemple défini ci-dessus, nous obtenons le manifest suivant :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<manifest>
  <learning-design identifier="LD-Pyramid2N" uri="" level="A"/>
  <titleLD>exemple d'un pyramid2N</titleLD>
  <learning-objective>Encourager la discussion </learning-objective>
<rôles>
< learner identifier="R-etud">
  <title>etudiant</title>
</ learner >
< learner identifier="R-clase">
  <title>classe</title>
</ learner >
<teacher identifier="R-profesor">
  <title>enseignant</title>
</teacher>
</rôles>
<activities>
  <structure-activity identifier="activite-N1">
    <learning-activity identifier="Etude individuelle">
      <title>Etude individuelle</title>
      <activity-description>Lire attentivement le cours </activity-description>
      <resref identifierref="cour1"/>
    </learning-activity>
    <learning-activity identifier="Proposition solution ">
      <title>Proposition solution </title>
      <activity-description> proposer un schéma qui présente
        I 'architecture d'un ordinateur</activity-description>
    </learning-activity>
  </structure-activity>
  <structure-activity identifier="activite-N2">
    <learning-activity identifier="Discussion Finale">
      <title>Discussion Finale</title>
      <activity-description>discuter </activity-description>
    </learning-activity>
    <learning-activity identifier="Proposition solution Finale">
      <title>Proposition solution Finale</title>
      <activity-description> proposer un schéma final </activity-description>
      <item identifierref="RES-DF0DA6E3-AB89-30A4-FD77-653FDAE72467" />
    </learning-activity>
  </structure-activity>
```

```

<support-activity identifier="activite-control">
  <title>Activite control</title>
  <activity-description>répondre aux questions des apprenants puis vérifier
  la solution finale</activity-description>
</support-activity>
</activities>
<method>
  <play identifier="PLAY-Pyramid2N">
    <title>Play</title>
    <act identifier="act-1">
      <title>Level 1</title>
      <role-part identifier="RP-S1">
        <title>RP_S</title>
        <role-ref ref="R-etud" />
        <structure-activity-ref ref="activite-N1" />
      </role-part>
      <role-part identifier="RP-T">
        <title>RP_T</title>
        <role-ref ref="R-profesor" />
        <support-activity-ref ref="activite-control" />
      </role-part>
    </act>
    <act identifier="act-2">
      <title>Level 2</title>
      <role-part identifier="RP-part11">
        <title>RP11</title>
        <role-ref ref="R-clase" />
        <structure-activity-ref ref="activite-N2" />
      </role-part>
      <role-part identifier="RP-part12">
        <title>RP12</title>
        <role-ref ref="R-profesor" />
        <support-activity-ref ref="activite-control" />
      </role-part>
    </act>
  </play>
</method>
<resources>
<resource type=" webContent" id="cour1" href="cour1.html" />
<resource type=" service" id="chat" href="chat.exe" />
</resources>
</manifest>

```

### **3. Conclusion :**

La création des situations apprentissage correspond réellement à la création des unités d'apprentissage qui permettent aux individus de vivre des situations d'apprentissage réelles. Cette exécution n'est possible que par l'usage de l'un des environnements d'exécution. Nous avons présenté dans ce chapitre un exemple réel d'un scénario pédagogique décrivant d'une part, l'objectif de l'apprentissage que l'enseignant veut atteindre, et d'autre part, les différentes activités qui doivent être achevées par les apprenant et l'enseignant. Finalement, nous avons exposé les différentes étapes nécessaires pour la création d'une unité d'apprentissage par le biais de notre outil.

# **Conclusion et Perspectives**

## Conclusion et perspectives

L'introduction de l'informatique dans l'éducation a conduit à l'évolution de ce domaine en le dotant des différents outils et plate forme qui permettent d'organiser le processus de la formation des apprenant pour créer les contenus pédagogique indispensables pour ce processus. Malgré la multitude des environnements développés, leur usage reste un peu plus difficile pour les non informaticiens (enseignant, ...).

Pour y remédier, nous avons vu la nécessité de faciliter le travail de l'enseignant qui joue un rôle déterminant dans le processus d'apprentissage, et l'aider à mettre en place des scénarios pédagogiques, en lui proposant un outil lui permettant de concevoir des situations d'apprentissage collaboratives sans difficultés.

Ainsi, dans l'objectif d'assurer la collaboration entre les apprenants, nous avons pensé que le développement des services permettant la communication entre les individus d'une part, et d'autre part, entre les individus et l'enseignant, est nécessaire.

A partir des problèmes abordés durant notre étude, nous avons essayé d'élaborer une solution pour répondre à ces problèmes. Cette solution consiste, en premier lieu, en la mise en place d'une approche conceptuelle pour la modélisation des scénarios pédagogiques, constituée de trois étapes : le choix du modèle de collaboration, la particularisation du scénario et la création d'une unité d'apprentissage, puis le développement d'un outil permettant de valider cette approche, dans le but d'être utilisé par les enseignants pour la création des unités d'apprentissages.

Dans des travaux futurs, nous souhaitons accomplir plusieurs tâches pour compléter le travail que nous avons fait, ces tâches peuvent être exprimées par les points suivants :

\_ Prendre en compte les deux autres niveaux B et C décrits par la spécification IMSLD, car nous avons travaillé seulement avec le niveau A.

\_ Comme nous n'avons expérimenté que sur les deux modèles d'activité : Pyramid et Jigsaw, nous espérons mettre en considération d'autres modèles comme par exemple : brainstorming, simulation, Think Pair Share (TPS),....

\_ En effet, nous avons crée un seul service de communication qui est : le service Chat, nous souhaitons mettre en œuvre d'autres services comme par exemple : Forum, messagerie,....



\_ Rajouter d'autres niveaux au modèle pyramid : pyramid avec cinq niveaux, six niveaux, sept niveaux, huit niveaux,..., car on a utilisé seulement : un pyramid avec deux niveaux, trois niveaux et quatre niveaux.

\_ Rendre les unités d'apprentissage résultantes de notre outil exécutable par les différents environnements d'exécution (copperCore,...).

# **Bibliographie**

# Bibliographie

[**All & Fro, 98**] ALLEN P, FROST S. (1998). Component-Based Development for Enterprise Systems: Apply the Select Perspective TM, SIGS Books/Cambridge University Press. ISBN 0521649994

[**AMB, 98**]. S.W Ambler, An Introduction to Process Pattern, Ambysoft white paper, 1998.  
<http://www.Ambysoft.com>.

[**Ant, 01**] Antoine Chalubert, 2001 : « Formalisation des données anatomique. Realisation d'un outil de génération de sites Web d'exploration anatomique ». JOBIM Comité organisateur.

[**Bag & Rét ,03**] : Baggetun, R., Retalis, S., et.al. (2003). Design patterns for CSCL. Workshop proposal. Bergen: CSCL, 2003.

[**Bar, 91**] Monique Baron, Régis Gras, Jean-François Nicaud, Eds. « Actes des deuxièmes journées d'EIAO de Cachan », les éditions de l'école normale de Cachan, 1991

[**Bet, 03**] Marie-Laure Betbeder-Matibet, 2003 : Symba : un environnement malléable support d'activités collectives en contexte d'apprentissage. Thèse de doctorat de l'Université du Maine.

[**BCL<sup>+</sup>, 04**] E.Bruneton, T.Coupaye, M.Leclercq, V.Quemu, and J.B.Stefani, An open component model. Internationnal Symposium on component based Software Engineering, May 2004.

[**BCS, 04**] E.Bruneton, T.Coupaye, and J.B.Stejani. The Fractal component model: specification, Technical report, France Telecom R&D-INRIA, Februnany 2004.

[**Bla, 05**] BLANC X., (2005) : MDA en action - Ingénierie logicielle guidée par les modèles, Édition Eyrolles, ISBN : 2-212-11539-3.

[**BRO & WAL, 98**] BROWN A.W., WALLNAU K.C. (1998), The Current State of CBSE, IEEE Software September/October 1998.

[**Car, 06**] Pierre-André Caron, Implantation de scénarios pédagogiques selon une approche orientée modèle : Une application de l'IDM aux EIAH, 1ères Rencontres Jeunes Chercheurs en EIAH, RJC-EIAH'2006, pages 1 à 8.

[**Cla, 88**] Gérard Claës, « Contribution à l'application de l'intelligence artificielle pour l'enseignement assisté par ordinateur », Thèse de Doctorat, Université de Paris-Sud, Octobre 1988.

[**Cyr, 06**] Cyril Vachet, « Apports de l'ingenierie dirigée par les modèles au domaine des EIAH », Université de Savoie, 2006. 1ères Rencontres Jeunes Chercheurs en EIAH, RJC-EIAH'2006, pages 56 à 64.

[Etu ,99] « Etude comparative technique et pédagogique des plates-formes pour la formation ouverte et à distance » : Une étude de la Direction de la Technologie, sous-direction des technologies éducatives, des technologies de l'information et de la Communication (DT/SDTETIC), Septembre 1999.

[Fer et al, 05] Christine Ferraris, Anne Lejeune, Laurence Vignollet, Jean-Pierre David, « Modélisation de scénarios d'apprentissage collaboratif», *Université de savoie, EIAH'2005*(Environnements Informatiques pour l'Apprentissage Humain), Montpellier, p. 285-296.

[FOW, 97] M.Fowler. Analysis patterns: reusable object models, Addison Wessley, 1997.

[Gia, 06] Ecaterina Pacurar Giacomini, 2006 : Une plate forme basée sur IMSLD pour la conception de cours en ligne dans le cadre de projet CEPIAH. Thèse de doctorat de l'université de technologie de Compiègne.

[Gen, 00] Geneviève LAMEUL, « Former et échanger par les réseaux, ingénierie de formation a distance », IUFM de Bretagne, Intervention au séminaire inter-iufm de Nantes 6 et 7/4/00 (adaptation pour RESAFAD).

[Geo, 01] Sébastien George, « Apprentissage collectif à distance, SPLASH : un environnement informatique support d'une pédagogie de projet », thèse de doctorat, 11/07/2001, université du Maine.

[Geo, 01] George.S, Apprentissage collaboratif à distance SPLACH : un environnement informatique support d'une pédagogie de projet, Thèse de doctorat en informatique université de Maine, 2001.

[Ger et Del, 99] Jerman, Dillemburg. An analysis of learner arguments in collective learning environment,1999, [second colloque de Guéret](#).

[HAS, 01] vers une meilleure formalisation du pattern : application du patterns, rapport DEA LSR université Joseph Fourier –Grenoble, 2001.

[Her et al, 05] Davinia Hernández-Leo, Juan I. Asensio-Pérez and Yannis Dimitriadis, 2005: Computational Representation of Collaborative Learning Flow Patterns using IMS Learning Design. International Forum of Educational Technology & Society (IFETS).

[IMS, 03a] IMS. IMS Content Packaging Version 1.1.3 Final Speci\_cation. Technical report, June-12 2003.

[IMS,03b] IMS. IMS Learning Design Best Practice and Implementation Guide. Technical report, 2003-January-2003 2003.

[KHA, 01] O. Khayati, Vers des systèmes de patrons formalisés et outillés, Rapport de DEA, LSR-Université Joseph Fourier –Grenoble, 2001.

[Kop, 01] Koper, Rob. Modeling units of study from a pedagogical perspective: the pedagogical metamodel behind eml. first draft, version 2. Technical report, Educational Expertise Technology Centre, Open University of the Netherlands, june 2001.

[Kos, 02] KOSHMANN T. dewey's contribution in the foundations of CSCL research, STAHL G(éd) 2002, p 17-23.

[Kra, 03] S.Krakowiak. Composants logiciels : introduction aux composants .Projet Sardes initia et imag-Ist, Université Loseph Fourie, Grenoble, 2003.

<http://sardes.inrialpes.fr/~krakowia/Enseignement/M2R-SL/CR/Flips/CR3-Composants-1.pdf>

[laf, 04] Pier Laforcade, « Met modelisation pour la conception et la mise en œuvre de situations \_problèmes cooperatives », thèse de doctorat l'université de PAU et des PAYS de l'ADOUR, 2004.

[LOR&PER, 04] LORTEAU Clément et PERALTA Sébastien, « Schema2XMLs ».Rapport de projet, 2004.

[Mac-Ros 03] Macrelle-Rosselle M. *Etude des objectifs d'une plate-forme de coopération pour les EIAH*. Actes de la conférence EIAH 2003, Strasbourg. pp. 379-390.

[Mar et al, 06] Christian Martel, Laurence Vignollet, Christine Ferraris : 'LDL : un langage support à la scénarisation pédagogique'. Actes du Colloque organisé dans le cadre de la 8e Biennale de l'éducation, 2006.

[Mba, 03] Aloys MBALA HIKOLO, 2003 : 'Analyse, conception, spécification et développement d'un système multi-agents pour le soutien des activités en formation à distance'. Thèse de Doctorat en informatique, Université de Franche-Comté. 2003.

[Mon, 95] Monique Baron, « EIAO, quelques repères », 1995.

<http://www.terminal.sgdg.org/articles/65/multimediaron.html>

[Nod, 05] Thierry Nodnot : 'Contribution à l'ingénierie dirigée modèles en EIAH : l e cas de situations problèmes cooperatives'. Thèse de doctorat, université de Pau et des Pays de l'Adour, 2005.

[Oub, 05] Lahcen OUBAHSSI : 'Conception de plates-formes logicielles pour la formation à distance, présentant des propriétés d'adaptabilité à différentes catégories d'usagers et d'interopérabilité avec d'autres environnements logiciels'. Thèse de doctorat de l'Université René Descartes – Paris V ,2005.

[Rass, 04] Dorothee Rasseneur – Coffinet : 'Saafir : un environnement support à l'appropriation d'une formation à distance'. Thèse de doctorat de l'université du Maine, 2004.

[Raw, 02] Rawlings, A. and van Rosmalen, P. and Koper, R. and Rodriguez-Artacho, M. and Lefrere, P. Survey of educational modelling languages (emls). Technical report, September 19st 2002.

[**Reb, 05**] Issam Rebaï, Nicolas Maisonneuve, Jean-Marc Labat, 2005. Un entrepôt pour stocker et rechercher des composants logiciels utiles aux EIAH, Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation (sticef).

[**Roy, 94**] Pea,R.D. seeing what we build together:distributed multimedia learning environments for transformatives communications. Journal of the learning sciences, 1994.

[**Ben et Ser, 07**] Soulef BENHAMDI, Hassina SERIDI. 'Modélisation et standardisation de situations d'apprentissage collaboratif', 4<sup>ème</sup> Symposium International en EAD : Technologies pédagogiques, Web et Réseaux, Sousse, Tunisie, Juin 2007.

[**Ste, 05**]. Sterenn AUBERT, 2005: Le e-learning adaptatif. Rapport de stage. Université de Nice Sophia-Antipolis.

[**Swi 04**] <http://www.swissmedia.ch/qsPortal/Home.asp>

[**SZY, 98**] C.Szyperski.Component Software.ISBN : O-201-17888-S.Addison-wesleyedition. 1998.

[**Szy, 02**] SZYPERSKI C. (2002), Component software beyond object-oriented programming. Addison Wesley, États-Unis.

[**Tch, Bak, 04**] Tchnikine .p,M.Baker et al,2004 : quelques dimensions pour l'analyse des travaux de recherches en conception d'EIAH rapport de l'AS 51 du RTP39.

[**Tch, 02**] Tchounikine, 2002b.conception des environnements informatique d'apprentissages : mieux articuler informatique et sciences humaines et sociales. Actes du GDR I3, p-223-245.

[**Zou, 06**] Katrina Zourou, 2006. Apprentissage collectif médiatisés et didactique des langues : instrumentation, dispositifs et accompagnement pédagogique. Thèse de doctorat, université Stendhal, Grenoble.

# **Annexes**

# Annexe A

## Java, un langage de programmation

### 1. Introduction :

Java est lancé par Sun Microsystems en 1995, permettrait selon l'adage : « write once,run everywhere »,c\_à dire écrire une fois pour toute des applications capable de fonctionner dans tous les environnement.Ce langage permet aux utilisateurs d'accéder aux informations provenant de différentes sources :serveur Web,base de données,...Il a été utilisés par la majorité des fournisseurs car il s'agit d'un langage très performant qui offre plusieurs avantages.

#### 1.1 Les avantages de Java:

- L'avantage le plus important de ce langage est qu'il est indépendant de toute plate forme d'exécution. Les programmes écrits sur la machine virtuelle java (JVM) fonctionnent, ainsi que la réalisation de cette machine, dans tous les environnements.
- la syntaxe de Java est analogue à celle du C++, ce qui le rend économique et professionnel.
- Java, contrairement au langage C++ qui orienté objet(mélange entre programmation à objet et programmation procédurale, est un langage purement objet,tous les élément sauf les types de bases sont des objets.
- Un autre avantage qui rend java plus sécurisée est qu'il est impossible de corrompre la mémoire car il n'y a pas d'accès direct à la mémoire, et pas de pointeurs qui présentent le risque d'écraser toute zone mémoire à cause de compteur erroné.

### 2. Les Applets Java :

Les Applets sont des programmes Java qui présentent un certain nombre de particularités:

- La méthode **main** est remplacée par la méthode **init**, qui n'est pas statique.
- Ils utilisent une interface fenêtrée affichée dans une zone rectangulaire d'une page HTML, qui la référence.
- Elle est téléchargée et exécutée par un navigateur Web lorsque ce document HTML est sollicité.

Les Applets peuvent être utilisable par un navigateur qui dispose d'un JVM (Java Virtuel Machine).



## 2.1 Mise en œuvre :

Une classe dérivée de la classe `java.applet.Applet` hérite de méthodes qu'il faut redéfinir en fonction des besoins et doit être déclarée public pour fonctionner. La méthode `init()` sert à initialiser l'applet, elle est appelée juste après la création de l'applet. La méthode `start()` permet de programmer les traitements, elle est appelée automatiquement après le chargement et l'initialisation de l'applet. La méthode `stop()` interrompt les traitements de tous les processus en cours. Pour implémenter une applet, il faut tout d'abord créer une classe dérivée de la classe *Applet* comme par exemple :

```
import javax.swing.*;

public class PremiereApplet extends JApplet {

    public void init() {

        getContentPane().add(new JLabel("Ça marche !"));

    }

}
```

Et voici le fichier HTML permettant d'exécuter l'applet :

```
<html>
<body>

<applet code="PremiereApplet" width="200" height="150">
</applet>

</body>
</html>
```

## 2.2. Les avantages des Applets :

- Elles sont exécutées sur le post client ce qui constitue un avantage non négligeable, car un grand nombre d'appels aux applets n'influence pas sur le serveur.
- Elle présente une interface utilisateur plus sophistiquée.
- Un autre avantage de l'applet, c'est qu'elle fonctionne constamment "en temps réel", répond aux événements de la souris, tout en restant sur la même page Web. Ce principe est intéressant, puisque si dans l'applet nous utilisons également des formulaires, il est possible de contrôler les valeurs d'entrée avant d'envoyer la requête.

### 3. Les Servlet Java :

Une Servlet est une programme qui s'exécute coté serveur, ou encore elle est chargée et exécutée par un serveur Web. Elle reçoit une requête du client, elle effectue des traitements et renvoie le résultat. La liaison entre la Servlet et le client peut être directe ou passer par un intermédiaire comme par exemple un serveur http.

#### 3.1. Les avantages des Servlet :

- Les Servlet sont indépendantes de toutes les plates formes et les serveurs Web.
- Les Servlet peuvent générer des pages HTML utilisant le protocole http.
- Elles s'appuie sur un langage vraiment standard : Java.
- Les Servlet utilisent tout protocole reposant sur le principe Client/Serveur :http,ftp,...,elles ne sont pas limitées à un seul protocole.
- Certaines requêtes de clients peuvent être satisfaites en renvoyant des documents préconstruits. Ce type de requête est géré par le serveur sans invoquer de Servlet. Cependant, dans un certain nombre de cas, un résultat statique n'est pas suffisant, et il faudra alors générer une page particulière pour chaque requête.

#### 3.2. Un exemple de Servlet :

Nous présentons dans ce qui suit un exemple d'une Servlet qui génère une page HTML affichant le mot : « Bonjour ! ».

```
public class Bonjour extends HttpServlet {  
public void doGet(HttpServletRequest request,HttpServletResponse response)  
throws IOException, ServletException {  
  
response.setContentType("text/html");  
PrintWriter out = response.getWriter();  
out.println("<html><head>");  
out.println("<title>Bonjour</title>");  
out.println("</head><body>");  
out.println("<h1>Bonjour !</h1>");  
out.println("</body></html>");  
}  
}
```

# Annexe B

## XML (eXtensible Markup Langage)

### 1. Introduction :

XML (eXtensible Markup Langage), signifie « langage de balisage extensible », développé par W3C, ce langage est devenu rapidement un standard de description et d'échange de données sur le Web. XML est un langage balisé comme l'HTML. La différence entre les deux langages est que l'HTML utilise des balises prédéfinies (qui conditionneront la manière dont le texte sera affiché dans le navigateur), alors que dans XML les balises sont utilisées seulement pour délimiter les éléments et l'interprétation des données est laissée à l'application qui les lit [Ant,01]. Il nous permet de décrire le processus d'apprentissage de façon formelle.

#### Les avantages de XML :

- Il est caractérisé par la définition de ses propres balises et ses propres attributs. Il possède un nombre élevé de balises contrairement à l'HTML, d'où se réside sa flexibilité.
- Un document XML peut être validé en utilisant le DTD ou des schémas qui permettent de définir sa structure.
- La séparation entre les informations et le traitement de la mise en forme.

### 2. Structure d'un document XML [LOR&PER, 04] :

Un document XML peut être structuré en trois éléments : Le prologue, Un arbre d'éléments et les commentaires.

#### 2.1. Prologue :

##### 2.1.1 Déclaration XML :

C'est la ligne d'introduction d'un document XML :

```
< ?xml version="1.0" encoding='ISO-8859-1' standalone='yes' ?>
```

Cette déclaration permet de fournir trois informations : la version du XML utilisée dans le document, le jeu de codage du caractère utilisé (encoding) et dépendance du document par rapport à une déclaration de type de document (standalone). Si standalone a la valeur 'yes', le processeur de l'application n'attend aucune déclaration de type de document extérieure au document. Sinon, le processeur attend une référence de déclaration de type de document. La valeur par défaut est 'no'.

### 2.1.2 Instructions de traitement :

Les instructions de traitement ne font partie du document (facultatives), elles sont interprétées par l'application afin de traiter le document XML. Un exemple de ces instructions est l'instruction qui sert à déclarer une feuille de style.

### 2.1.3 Déclaration de type de document (DTD) :

La DTD (Document Type Definition) sert à définir la structure d'un document XML. Elle peut être vue comme une grammaire algébrique dont les non terminaux sont les éléments balises et les terminaux sont les différents types de données présents dans l'arborescence du document. Un document XML est dit valide s'il est conforme à une DTD donnée.

### 2.2 Les commentaires :

En XML, les commentaires se déclarent de la même façon qu'en HTML. Ils sont encadrés par les marques de début et de fin de commentaire :

```
<!-- Ceci est un commentaire -->
```

### 2.3 L'arbre d'éléments :

Un document XML peut être structuré en arborescence d'éléments. Cette arborescence est constituée d'un élément racine (unique), des branches et des feuilles. Chaque élément peut contenir un ou plusieurs attributs. Un attribut est composé d'un nom et d'une valeur (« nom = 'valeur' »).

Un document XML peut contenir des entités. Elles peuvent être utilisées aussi bien dans la DTD que dans le document XML. Certains caractères ayant un sens précis en XML, il est nécessaire de leur trouver un remplaçant lorsque l'on a besoin de les insérer dans un document. On a recours dans ce cas à des entités prédéfinies. Ces entités sont :

Caractère	Entité
&	&amp;
<	&lt;
>	&gt;
"	&quot;
'	&apos;

Table 1. Liste des entités prédéfinies

## 3. Schéma XML :

Il est conçu pour pallier les inconvénients du DTD, c'est un langage qui permet de définir le format d'un document XML. Un schéma XML est lui-même un fichier XML (XML Schema

Description en anglais, ou fichier XSD). La connaissance de la structure d'un document XML permet notamment de vérifier la validité de ce document.

#### **4. Les feuilles de styles :**

Pour qu'un fichier XML soit affichable il faut lui ajouter une « feuille de style » servant à définir une présentation physique (marges, polices et tailles de caractères, couleurs, etc.) pour celui-ci. Les feuilles de styles CSS (Cascading style Sheets) permettent de contrôler la mise en page des différents éléments qui composent un document HTML ou XML. XSL (eXtended Style Language), comme CSS, est un complément « feuille de style » de XML.

#### **5. L'API JDOM :**

Java Document Object Model(JDOM) est une API écrite en Java afin de manipuler des fichiers XML. Cette API passe outre les limites de DOM dont elle s'inspire. Il existe également une API pour le traitement des données XML nommée :SAX, qui est très limitée quand il s'agit d'effectuer des modifications au sein du document. L'intérêt principale de JDOM est sa simplicité.

# Annexe C

## UML (Unified Modeling Language)

### 1. Introduction :

L'UML est utilisé pour visualiser, spécifier, construire et documenter les artefacts d'un système à fortes composantes logicielles. Ce langage est issu de la fusion de plusieurs d'autres langages de modélisation orientés objets. Ce formalisme modélise le système en utilisant neuf diagrammes selon deux modes de représentation : le premier est dédié pour la structure du système (diagramme de cas d'utilisation, de classe, etc.) le deuxième concerne sa dynamique de fonctionnement (diagramme d'états transitions, de séquence, d'activité, etc.).

### 2. Les diagrammes de l'UML [laf, 04] :

Nous présentons par la suite seulement les deux diagrammes qu'on a utilisé : diagramme de classe et diagramme d'activité.

#### 2.1. Diagramme de classe :

Ce diagramme sert à modéliser la structure statique du système en utilisant des éléments tels que : classe, paquetage, ... Il montre cette structure comme un ensemble de classes et de relations entre ces classes. Ce diagramme est utilisé pour représenter le modèle d'information de l'IMSLD.

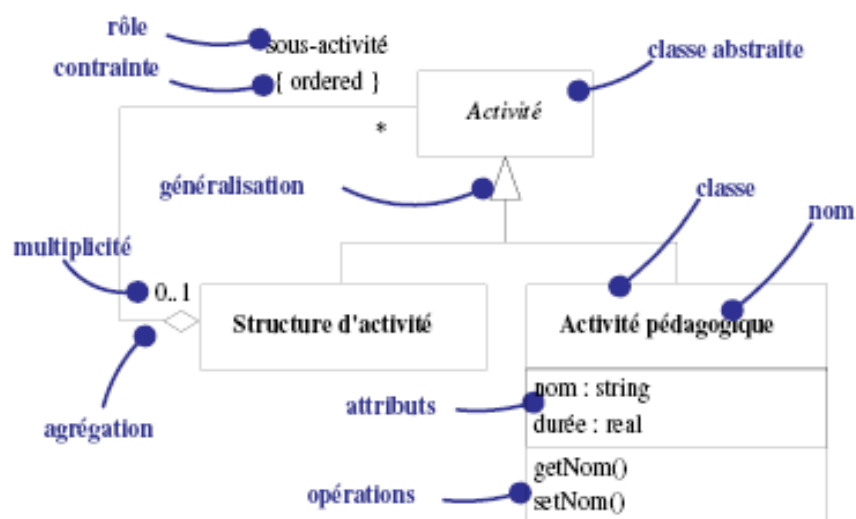


figure1: Un exemple simplifié et annoté de diagramme de classe.

## **2.2. Diagramme d'activité :**

Ce diagramme est utilisé pour représenter le comportement du système. Il permet également de décrire les flux d'informations (workflows) comme les flux de données (dataflow). Le diagramme d'activité sert à spécifier les scénarios de nos situations d'apprentissage collaboratif. La structure de ces scénarios ou organisation des activités est décrite en termes de *act* et *play* dans IMS-LD.

# Annexe D

## Le Questionnaire

### 1. Introduction :

le questionnaire que va être soumis à l'enseignant est composé de deux questions. Le premier permet à l'enseignant de déterminer l'objectif d'apprentissage. Le deuxième sert à préciser le type du contenu pédagogique à utiliser pendant l'apprentissage, ces types sont tirés réellement des modèles de scénarios (modèles d'unités pédagogiques) qui sont définis par Giacomini [Gia,05]. Cette définition comporte la spécification du type de contenu pédagogique utilisé lors de la formation, c-à-d si le cours contient des parties théoriques, des exercices, des projets, des problèmes à résoudre, etc. comme elle comporte aussi la description scénario ou le déroulement de l'apprentissage. Pour notre travail ce qui nous intéresse est le type du contenu pédagogique décrit dans chaque modèle. Le tableau suivant permet de citer ces modèles.

No de modèle	Type d'unité pédagogique
1	Résolution de problème/exercices
2,3	concepts théoriques avec exercices
4	concepts théoriques, exercices de cours, résolution de problèmes
5, 6, 7,8	Résolution de problèmes
9	concepts théoriques, résolution de problèmes
10,11	concepts théoriques
12, 13,14, 15	projet
16	Modélisation et résolution de problèmes

Tableau 1: les modèles d'unités pédagogiques.

### 2. Questionnaire :

La structure du questionnaire est la suivante :

- quels sont les objectifs d'apprentissage?
- Assurer une interdépendance positive
- Encourager la discussion entre les individus
- Résoudre des problèmes complexes
- La mise en compte de l'individu



- quel est le type du contenu pédagogique ?
  - Résolution de problème/exercices
  - Concepts théoriques avec exercices
  - Concepts théorique, exercices de cours,résolution de pbs
  - Résolution de problèmes
  - Concepts théoriques, résolution de problèmes
  - Concepts théoriques
  - Projet
  - Modélisation et résolution de problèmes