

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université IBN KHALDOUN de Tiaret**

Faculté de l'ingénieur et des sciences de l'ingénieur  
Département d'Informatique

**Mémoire**  
En vue de l'obtention du diplôme de  
**Magister en Informatique**

Ecole doctorale: **STIC**  
(Science et Technologie de l'Information et de la Communication)  
Option : **SIC**  
(Systèmes d'Information et de Connaissance)

---

# **Vers un design pédagogique adaptatif**

## Extension de la spécification IMS-LD

---

Présenté par :

**Omar TALBI**

Devant le jury composé de:

*Président :*

Mr. DAHMANI Youcef

Maître de conférences à l'université de Tiaret

*Examineurs :*

Mr. CHIKH Amine

Maître de conférences à l'université de Tlemcen

Mr. NOUALI Omar

Maître de conférences CERIST Alger

Mr. BELARBI Mostefa

Maître de conférences à l'université de Tiaret

*Directeur de mémoire :*

Mr. CHIKH Azeddine

Maître de conférences à l'université de Tlemcen

2008-2009



# Remerciements

*Je remercie ALLAH le tout puissant de m'avoir aidé à l'aboutissement de ce mémoire.*

Je tiens à remercier tous ceux qui m'ont ouvert la voie à ces deux années d'études et de travail et à tous ceux qui ont contribué de près ou de loin à ce cheminement.

Mes vifs remerciements, accompagnés de toute ma gratitude, vont tout d'abord à Monsieur Azeddine CHIKH, maître de conférences à l'université Abou-Bakr BELKAID Tlemcen, qui a accepté de diriger ma thèse et qui a bien voulu me faire profiter de sa rigueur scientifique, et de la pertinence de ses remarques. Je le remercie également pour son aide efficace et ses judicieux conseils.

Mes respects et mes remerciements vont à Monsieur DAHMANI Youcef, Maître de conférences à l'université de Tiaret pour m'avoir fait l'honneur d'accepter de présider le jury de ce mémoire.

Je tiens également à remercier Monsieur CHIKH Amine, Maître de conférences à l'université de Tlemcen, Monsieur BELARBI Mostefa, Maître de conférences à l'université de Tiaret et Monsieur NOUALI Omar, Maître de conférences au CERIST Alger pour m'avoir fait l'honneur d'examiner et de juger ce travail.

Mes remerciements vont également à mes collègues de l'école doctorale STIC, l'équipe de Tlemcen et un grand merci à celle de Tiaret avec qui j'ai passé une année pleine de rigueur et si riche en connaissances.

Pour finir, je tiens à remercier mes chers parents pour m'avoir soutenu moralement, Merci à ma femme, qui m'a toujours encouragé, et qui a su me redonner confiance en moi-même dans les moments difficiles. Merci à mes très chères sœurs et à mon frère, à mes enfants qui malgré leurs jeunes âges m'ont encouragé à travailler sans relâche, Merci à ma belle famille, Mme Henni Halima, Khiati Jamel eddine et un Grand Merci à Khiati Saliha Ilhem pour ses encouragements et son aide si précieuse.

# Résumé

Ce travail de recherche traite du domaine du e-learning et plus précisément du design adaptatif.

En effet la conception des cours ordinaires est désormais munie d'un savoir et d'un savoir-faire relativement très maîtrisés.

On s'intéresse dans ce sujet plus particulièrement aux cours adaptatifs dont le public apprenant cible n'est pas connu à priori avec précision.

Dans notre travail, nous avons présenté en détail la spécification IMS Learning Design (IMS-LD), qui fournit un cadre conceptuel de modélisation d'une unité d'apprentissage instrumentée par les technologies d'information et de communication. Nous avons mis l'accent sur les possibilités offertes en matière d'adaptation des unités d'apprentissage, nous avons montré comment l'adaptation peut être représentée dans la spécification IMS-LD, quels sont les éléments IMS-LD qui y interviennent et comment procéder à leur utilisation.

Finalement, et compte tenu de l'étroite relation qui existe entre ces éléments et le profil de l'apprenant, élément pivot sur lequel s'articule l'adaptation, nous proposons une extension du modèle de la spécification IMS-LD.

**Mots-clés:** e-learning, Learning Design, adaptation, IMS-LD, profil de l'apprenant.

# Abstract

The research task discusses of e-learning domain and more specifically of adaptive design.

Indeed the ordinary course design is now equipped with knowledge and know-how relatively very controlled.

We are interested in this subject more particularly at the adaptive course of which the public learning target is not known in advance with precision.

In our work report, we presented in detail the specification IMS Learning Design (IMS-LD), which provides a conceptual framework for modeling instrumented unit of learning by the technologies of information and communication. We have focused on the possibilities for adaptation of units of learning, we have shown how adaptation can be represented in the IMS-LD, which is the IMS-LD elements that are involved and how to use.

Finally, given the close relationship between these elements and the profile of the learner, the pivot element on which is based adaptation; we propose an extension of the model of IMS-LD specification.

**Key word:** e-learning, Learning Design, adaptation, IMS-LD, learner profile.

## TABLE DES MATIERES

Introduction générale .....	11
1. Contexte et problématique .....	11
2. Plan du mémoire.....	12
<b>Chapitre I : E-learning</b> .....	13
1. Introduction .....	14
2. Définitions .....	14
3. Apports du e-learning .....	14
4. Dispositifs de formation en ligne .....	15
4.1. Définition.....	15
4.2. Cycle de vie.....	15
4.3. Acteurs du e-learning.....	17
5. Des objets d'apprentissage aux langages de modélisation pédagogique.....	17
5.1. Le LOM .....	17
5.2. Limites du LOM.....	19
5.3. SCORM .....	20
5.3.1.Modèle d'agrégation de contenu.....	20
5.3.2.Environment d'exécution.....	21
5.4. Apports et limites du SCORM.....	22
5.5. IMS –LD .....	22
5.5.1.Educational Modelling Language (EML).....	23
5.5.2.Avantages d'EML.....	23
5.5.3.Pourquoi une spécification ? .....	24
5.5.4.Relations entre IMS-LD et les autres spécifications :.....	24
5.5.5.La spécification IMS-LD.....	25
5.5.6.Modèle conceptuel.....	26
<b>Chapitre II : Construction d'une unité d'apprentissage en utilisant la spécification IMS-LD</b> .....	28
1. Introduction .....	29
2. Modèle structurel.....	29
3. La spécification IMS-LD .....	29
3.1. Structure de base d'un Learning Design – niveau A .....	29
3.2. Learning design Niveau B - propriétés et conditions - .....	35
3.3. Learning Design de niveau C .....	37
4. Création du learning design .....	39
4.1. Processus de modélisation d'une unité d'apprentissage .....	39
4.1.1.Phase d'analyse (Analyse descriptive) .....	39
4.1.2.Phase de conception.....	39

4.1.2.1. Production d'un diagramme d'activité UML.....	40
4.1.2.2. Document XML conforme IMS-LD.....	41
<b>4.2. Rédaction du scénario d'apprentissage IMS LD.....</b>	<b>41</b>
4.2.1. Fichiers composants le Learning Design.....	42
4.2.2. Le fichier manifeste : « imsmanifest.xml » .....	42
4.2.2.1. Ressources de type « webcontent » .....	43
4.2.2.2. Ressources de type « imslcontent » : XHTML.....	43
4.2.3. Utilisation d'un éditeur: Reload Learning Design Editor.....	44
<b>4.3. Implémentation : .....</b>	<b>50</b>

## **Chapitre III : Adaptation ..... 52**

<b>1. Introduction .....</b>	<b>53</b>
<b>2. C'est quoi l'adaptation.....</b>	<b>53</b>
<b>3. Rôles et entrées (inputs) .....</b>	<b>54</b>
<b>4. Niveaux d'adaptation .....</b>	<b>56</b>
<b>5. Principales approches de conception d'UA adaptatives (Etat de l'art) .....</b>	<b>56</b>
<b>6. Types d'adaptation .....</b>	<b>59</b>
<b>7. Adaptation sous l'angle IMS-LD.....</b>	<b>61</b>
7.1. Adaptation de l'interface.....	61
7.2. Adaptation basée sur le flux d'apprentissage : .....	61
7.3. Adaptation basée sur le contenu.....	62
7.4. Encadrement interactif de la résolution de problèmes.....	62
7.5. Filtrage d'informations adaptatif.....	63
7.6. Groupement d'utilisateurs adaptatif.....	63
7.7. Evaluation adaptative .....	63
7.8. Faire des changements à la volée .....	64

## **Chapitre IV : Vers un design pédagogique adaptatif, extension de la spécification IMS-LD ..... 65**

<b>1. Introduction .....</b>	<b>66</b>
<b>2. Analyse .....</b>	<b>66</b>
2.1. Première analyse.....	66
2.2. Analyse détaillée .....	67
<b>3. Démarche à suivre .....</b>	<b>68</b>
<b>4. Modèle de données du Profil de l'apprenant: .....</b>	<b>69</b>
<b>5. Principaux modèles existants .....</b>	<b>69</b>
5.1. Présentation du modèle apprenant PAPI .....	69
5.2. Présentation du modèle IMS LIP ( <i>IMS Learner Information Package</i> ) .....	70
5.2.1. C'est quoi IMS LIP.....	71
5.2.2. Aspect de IMS LIP.....	73
5.2.3. IMS- IP comme support de personnalisation d'un apprenant.....	76

<b>6. Identification des éléments de la spécification IMS LD touchés par les changements :</b>	<b>77</b>
6.1. Premier type de changement apporté par le Learning flow :	77
6.2. Deuxième type de changement apporté par le Contenu	81
6.3. Troisième type de changement apporté sur l'interface	83
<b>7. Extension du LD</b>	<b>84</b>
7.1. Introduction	84
7.2. Extension du LD	84
8. Conclusion	87
 <b>Conclusion générale</b>	 <b>88</b>
 <b>Références bibliographiques</b>	 <b>89</b>



## LISTE DES FIGURES

### Chapitre I

Figure I.1 : Cycle de vie d'un dispositif de formation .....	16
Figure I.2 : Catégories LOM .....	18
Figure I.3 : Modèle d'agrégation de contenu de SCORM .....	21
Figure I.4: Environnement d'exécution de SCORM.....	21
Figure I.5 : Vue simplifiée du modèle EML.....	23
Figure I.6 : Inclusion d'un LD dans un IMS CP.....	25
Figure I.7 : Le schéma du modèle conceptuel (relationnel) d'IMS-LD .....	27

### Chapitre II

Figure II.1 : Schéma de structuration générale d'IMS-LD .....	29
Figure II.2: structure du <i>learning-design</i> , au niveau A .....	30
Figure II.3 : structure du modèle <i>components</i> , au niveau A .....	30
Figure II. 4 : structure du modèle <i>roles</i> , au niveau A.....	31
Figure II.5 : structure du modèle <i>method</i> , au niveau A .....	31
Figure II.6 : structure du modèle <i>play</i> , au niveau A .....	32
Figure II.7 : structure du modèle <i>act</i> , au niveau A .....	33
Figure II.8 : structure du modèle <i>activities</i> , au niveau A .....	33
Figure II.9 : structure du modèle <i>environments</i> , au niveau A [.....	34
Figure II.10 : structure des modèles <i>loc-property</i> et <i>glob-propert</i> , au niveau B .....	35
Figure II .11: structure du modèle <i>conditions</i> , au niveau B .....	37
Figure II.12: structure du modèle <i>notification</i> , au niveau C .....	38
Figure II.13: structure du modèle <i>global-elements</i> , au niveau C .....	38
Figure II.14 : Processus de modélisation d'une unité d'apprentissage .....	39
Figure II.15 : Diagramme d'activités UML.....	40
Figure II.16 : structure d'un document XML décrivant une situation d'apprentissage .....	41
Figure II.17 : structure d'un fichier <i>manifest.xml</i> .....	42
Figure II.18: Reload Learning Design Editor, version 2.1.0 .....	44
Figure II.19: onglet <i>Roles</i> .....	45
Figure II.20: onglet <i>Porperties</i> .....	46
Figure II.21 onglet <i>Activities</i> .....	47
Figure II.22: onglet <i>Environments</i> .....	48
Figure II.23: onglet <i>Method</i> .....	49
Figure II.24: onglet <i>Export</i> .....	50

### Chapitre III

Figure III.1 : Inputs et rôles dans un système d'apprentissage adaptatif.....	55
Figure III.2 : Niveaux d'adaptation .....	56
Figure III.3 : Les quatre phases du cycle de vie de l'adaptation dans ALFANET.....	58

### Chapitre IV

Figure IV.2 : Relation entre IMS LIP et PAPI.....	71
Figure IV.3 : Composants d'un SAI .....	72
Figure IV.4 : Structure de données de la spécification LIP.....	74
Figure IV.5 : ensemble de règles adapte le learning flow selon les performances de l'apprenant.....	79
Figure IV.6 : L'utilisateur lui même modifie le learning flow selon ses décisions personnelles.....	80
Figure IV.7 : Extension du Learning design.....	86

## LISTE DES TABLEAUX

Tableau II.1 : les changements apportés aux modèles de niveau A par l'ajout des <i>properties</i> .....	36
Tableau IV.1 : Tableau résumant les stratégies de conception d'une UA.....	59

## Introduction générale

### 1. Contexte et problématique

Tout au long de l'histoire de l'humanité, les technologies ont dessiné et ont fait évoluer l'organisation de nos sociétés. Internet et la rapide évolution des Technologies de l'Information et de la Communication (TIC) ont engendré au cours de ces dernières années une progression de leurs usages dans tous les domaines et en particulier dans ceux de l'éducation et de la formation.

Ces technologies sont connues sous le nom de Technologies de l'Information et de la Communication pour l'Enseignement (TICE). Leurs applications ont donné naissance à une nouvelle forme d'apprentissage appelée e-learning.

Le e-learning est basé sur l'accès à des informations en ligne en utilisant des plates formes de formation ou des « sites e-formation » rendant ainsi le processus d'apprentissage indépendant du temps et du lieu.

Ce travail de recherche s'inscrit dans le contexte du développement des TICE et dans le domaine du e-learning, ayant conduit à la naissance d'un nouvel horizon de recherche sur la conception des cours.

Cette conception de cours, allant de la production des documents jusqu'à l'établissement du processus d'apprentissage a nécessité le développement de plusieurs normes et standards tels que : LOM, SCORM, IMS LD, etc.

Une analyse faite par PERNIN Jean-Philippe [**Pernin, 2006**], fait ressortir quatre approches :

L'approche artisanale : l'enseignant était auteur ; il était formé à des outils lui permettant de créer lui-même ses cours.

L'approche éditoriale : l'enseignant concepteur et prescripteur de contenu ; il donnait son cours à réaliser à d'autres - les services multimédias, en l'occurrence.

L'approche documentaire basée sur les objets d'apprentissage : l'enseignant prospecteur, "référenceur" et organisateur de ressources.

L'approche centrée sur l'activité : l'enseignant est ensuite devenu plutôt un scénariste au travers de ce que l'on nomme les unités d'apprentissage ou les scénarios : l'approche est alors centrée sur l'activité.

L'approche centrée sur l'activité fait apparaître le rôle central du modèle pédagogique nécessaire à tout système d'apprentissage.

La spécification IMS LD met au centre le concept d'unité d'apprentissage (UA) représentant le modèle pédagogique. Cette UA est appelée un Learning Design (LD).

Des apprenants faisant partie d'un même groupe n'ont pas nécessairement les mêmes connaissances préalables, le même profil cognitif ni les mêmes objectifs et motivations vis-à-vis de la formation suivie. Traditionnellement, un cours qui leur est destiné ne prend pas en compte cette divergence, et sera dispensé à tous de la même façon.

Dans ce mémoire, on s'intéresse plus particulièrement aux cours adaptatifs dont le public apprenant cible n'est pas connu à priori avec précision.

Par conséquent les éléments pédagogiques appartenant au LD et susceptibles de changer doivent être suffisamment adaptés en vue de prendre en compte une variété de choix et de préférences.

Comment alors ces caractéristiques adaptatives et ces éléments peuvent être modélisés dans IMS-LD ?

En faisant appel au niveau B de la spécification IMS-LD, nous proposons une extension de la spécification IMS-LD.

## 2. Plan du mémoire

Ce travail est réparti en quatre chapitres :

Le chapitre I intitulé **e-learning** présente le domaine du e-learning, le contexte dans lequel se situe notre travail, tout en précisant les composants sur lesquels repose cette nouvelle forme d'apprentissage en ligne. Nous présentons également les différentes approches dans la modélisation et la représentation des situations d'apprentissage, celles centrées apprenant et celles centrées activités et vers quelles normes ou propositions elles ont abouti, à savoir le LOM, le SCORM et IMS-LD.

Le second chapitre intitulé **construction d'une UA en utilisant la spécification IMS-LD**, aborde la spécification IMS LD, cadre théorique de notre recherche, beaucoup plus en détail, afin de mieux comprendre la structure des éléments composant le modèle IMS LD tout en précisant leurs rôles respectifs tirés de l'IMS Learning Design Information Model. Nous clorons ce chapitre en présentant les étapes par lesquelles passent la création et l'implémentation d'un Learning Design utilisant la spécification IMS-LD.

Le troisième chapitre intitulé **Adaptation**, concept fondamental de la problématique qu'on s'est proposée d'étudier, et qui en constitue un état de l'art met en évidence cette notion en décrivant ses types, ces critères ainsi que ses méthodes. Nous verrons également l'aspect adaptation sous l'angle IMS-LD.

Dans le quatrième et dernier chapitre nous présentons notre contribution qui consiste à aller vers un design adaptatif en mettant l'accent sur les changements qui peuvent être apportés au Learning Design afin que ce dernier puisse s'adapter au profil de l'apprenant ainsi que les éléments de la spécification IMS LD touchés par ces changements. Ce qui nous emmène finalement à présenter une proposition d'extension de la spécification IMS LD.

Nous clorons ce mémoire par une conclusion et mettons le doigt sur les perspectives de recherche dans le cadre du travail présenté.

## Chapitre I : E-learning

Dans ce premier chapitre nous présentons le domaine du e-learning, le contexte dans lequel se situe notre travail. Nous présentons également les différentes approches, celles centrées apprenant et celles centrées activités et vers quelles normes ou propositions elles ont abouti, à savoir le LOM, le SCORM et IMS-LD.

## 1. Introduction

Le réseau Internet, cet énorme réservoir de ressources informationnelles sans cesse alimenté et l'émergence des Technologies de l'Information et de la Communication pour l'Enseignement (TICE) ont donné naissance à un nouveau mode d'apprentissage appelé « e-learning ».

L'introduction de ces TICEs vise à améliorer la qualité de la formation en facilitant l'accès aux ressources et services du web d'une part, et la collaboration à distance d'autre part.

## 2. Définitions

Le e-learning (terme anglophone pour e-formation) que l'on pourrait traduire par apprentissage (Learning) en ligne (e : Abrév. de "électronique" et maintenant de "en ligne".), est selon la commission européenne qui propose une définition très large:

*«L'utilisation des nouvelles technologies multimédias et de l'Internet pour améliorer la qualité de l'apprentissage en facilitant l'accès à des ressources et des services, ainsi que les échanges et la collaboration à distance». [Cons, 2001]*

Une autre définition proposée par le ministère de l'éducation française est la suivante :

*«Le e-learning » définit tout dispositif de formation qui utilise un réseau local, étendu ou l'Internet pour diffuser, interagir ou communiquer, ce qui inclut l'enseignement à distance, en environnement distribué, l'accès à des sources par téléchargement ou en consultation sur le net. Il peut faire intervenir du synchrone ou de l'asynchrone, des systèmes tutorés, des systèmes à base d'autoformation, ou une combinaison des éléments évoqués. Le e-learning résulte donc de l'association de contenus interactifs et multimédia, de supports de distribution (PC, Internet, intranet, extranet), d'un ensemble d'outils logiciels qui permettent la gestion d'une formation en ligne et d'outils de création de formations interactives. L'accès aux ressources est ainsi considérablement élargi de même que les possibilités de collaboration et d'interactivité». [Educnet]*

## 3. Apports du e-learning

Le e-learning est donc un mode d'apprentissage à distance s'appuyant sur les technologies de l'information et de la communication (TIC), il permet :

- de réduire les contraintes d'espace et de temps,
- d'individualiser le parcours de formation (l'ensemble des modules de formation à suivre par l'apprenant),
- de favoriser l'autonomie de l'apprenant et
- de développer des relations en ligne avec d'autres apprenants ou le tuteur qui a pour mission de le suivre et de l'accompagner au cours de la formation

C'est donc un dispositif de formation supporté par les technologies, où les lieux et temps de formation sont flexibles et sont donc choisis en fonction des disponibilités des apprenants. Outre le tuteur, nécessaire au suivi de l'apprenant, une communauté d'apprenants peut aussi être fédérée afin de créer un système d'entraide et de travail collaboratif en ce sens où les apprenants collaborent aux apprentissages du groupe et, en retour, le groupe collabore à ceux des apprenants.

Les communications entre acteurs du e-learning peuvent se faire de façon : synchrone, en temps réel (chat par exemple) et /ou en asynchrone, en différé (forums, mails).

Le domaine du e-learning repose sur deux composantes à savoir :

(1) l'ingénierie pédagogique, qui désigne la démarche pédagogique adoptée dans la conception et la diffusion des cours (IMS-LD que nous verrons par la suite s'inscrit dans cette démarche) ; et

(2) la solution technologique, qui s'appuie sur des plates-formes aux fonctionnalités variées [Berk, 2008].

Des outils tels que des plateformes de e-learning ou Plateforme d'apprentissage en ligne (LMS : Learning Management System) assurent et optimisent, la gestion de l'ensemble des activités depuis l'inscription des participants, la distribution des ressources, l'organisation de parcours pédagogiques individualisés, le suivi des formations par le tuteur (gestion intégrée des interactions apprenants - formateur) jusqu'à l'animation de communautés d'apprentissage.

## 4. Dispositifs de formation en ligne

### 4.1. Définition

Un dispositif de formation en ligne est un ensemble d'éléments (méthodes, outils, services, procédures, routines et principes d'action) articulés, ayant pour finalité de produire des compétences individuelles et collectives et de faciliter l'apprentissage [Web3.2].

Néanmoins, le dispositif ne se limite pas aux aspects techniques ou opérationnels, mais prend en compte également les relations entre les acteurs (formateurs, apprenants, ...). Il y'a une prise en charge des différents types d'interactions pouvant exister entre apprenant - apprenant, apprenant - formateur, etc.

Nous pouvons distinguer plusieurs types de dispositifs de formation. Nous présentons la classification suivante, selon les objectifs [Berk, 2008] :

- gestion de la formation, englobe les fonctionnalités administratives permettant d'automatiser l'organisation de la formation, l'accessibilité des cours et le suivi ;
- gestion des compétences, ce sont les fonctions permettant d'évaluer les compétences ;
- bibliothèque de formation, désigne la constitution de la base de données des cours et l'organisation du catalogue des formations ;
- création de cours, ce sont les outils permettant de créer des formations et de formaliser une certaine démarche pédagogique sans avoir à utiliser la programmation ;
- accès au cours, c'est tout ce qui a trait à la diffusion des formations ;
- accompagnement de l'apprenant, regroupe les moyens fournis pour suivre l'apprenant.

Dans ces dispositifs de formation, trois modes de formation sont possibles :

- Asynchrone : c'est une méthode de formation où les interactions entre l'apprenant et le tuteur sont en temps différé (utilisation de la messagerie ou des forums).
- Synchrone : c'est une méthode de formation où les interactions entre l'apprenant et le tuteur sont en temps réel (utilisation du chat ou de la visioconférence).
- Mixte, qui combine les éléments asynchrones et synchrones.

Parmi les plateformes d'enseignement utilisées, on peut citer Moodle, Blackboard, Acolad...

### 4.2. Cycle de vie

Le cycle de vie d'un dispositif de formation en ligne (cf. **Figure I.1**) est composé de deux phases principales: (1) la phase d'acquisition et (2) la phase d'utilisation.

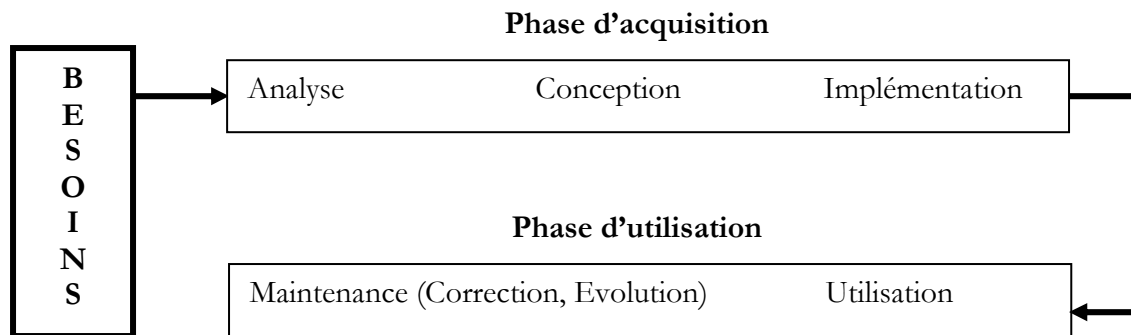


Figure I.1 : Cycle de vie d'un dispositif de formation

**La phase d'acquisition**, comprend généralement trois étapes : (a) l'étape d'analyse, (b) l'étape de conception et (c) l'étape d'implémentation.

- L'étape d'analyse consiste à l'établissement d'un cahier de charge dans lequel sont décrits les besoins et les exigences en matière de formation en ligne.
- L'étape de conception consiste à spécifier selon un modèle tel que IMS-LD, les scénarios d'apprentissage conformément au cahier de charge. Ces derniers sont exprimés par l'enseignant en décrivant toutes les scènes (rôles, actions à réaliser, ...) et leurs enchaînements. Par la suite, un ingénieur pédagogue connaissant les particularités de la plate-forme de destination, en collaboration avec l'enseignant formalisent le scénario pédagogique. Il s'agit de spécifier les différents composants de ce scénario: (1) les rôles des acteurs (apprenants, enseignants, tuteurs,...), (2) les activités à mettre en œuvre et leurs différentes phases de réalisation et (3) l'environnement adéquat composé de services, d'outils et de ressources.

Cette étape est particulièrement importante, puisque la qualité de la conception pédagogique est un élément clé pour la réussite auprès des apprenants. Une description plus détaillée sera présentée dans le chapitre suivant, qui sera consacré à une étude de la spécification IMS-LD.

- L'étape d'implémentation consiste à réaliser, dans un environnement d'apprentissage tel qu'un LMS<sup>1</sup>, les spécifications des scénarios d'apprentissage obtenues dans l'étape précédente.

**La phase d'utilisation**, concerne le déroulement du dispositif développé dans la phase précédente. On distingue deux étapes : (a) l'étape d'utilisation et (b) l'étape de maintenance.

- L'étape d'utilisation, consiste à suivre et à évaluer les différentes activités d'apprentissage. Plus précisément, il s'agit (1) d'évaluer le comportement des apprenants et des tuteurs afin de vérifier que le modèle prévu initialement correspond bien aux besoins des différents acteurs et (2) détecter les éventuels ajustements qui pourront par la suite être intégrés dans le module. Le comportement de la plate-forme au cours de son utilisation en condition réelle, peut également être évalué, demandant peut-être alors d'apporter des solutions en reconfigurant et en développant encore une fois certains de ses composants.
- L'étape de maintenance, consiste à apporter les éventuelles corrections et améliorations, qui sont d'ordre technique ou pédagogique.

<sup>1</sup> Learning Management System



### 4.3. Acteurs du e-learning

Toute une équipe d'individus qualifiés appelés acteurs du e-learning vont intervenir dans les phases d'acquisition et d'utilisation d'un dispositif de formation en ligne [Berk, 2008]

- Acteurs intervenant dans la phase d'acquisition

Les acteurs pouvant intervenir dans la phase d'acquisition d'un dispositif de formation en ligne sont :

L'enseignant : rôle général représentant l'ensemble des acteurs : didacticien, sociologue, psychologue, ou pédagogue, pouvant intervenir dans la conception de situations d'apprentissage.

L'ingénieur pédagogue : assiste l'enseignant dans la conception de ses scénarios pédagogiques, l'aide à structurer son cours et à choisir les outils et les médias les plus adéquats. Il assure la formalisation des modèles en accord avec l'enseignant et en tenant compte des caractéristiques de la plate-forme. Il assure également la coordination entre l'enseignant et l'équipe technique depuis la conception du dispositif jusqu'à sa réalisation.

L'équipe technique : constituée de profils différents : développeur informatique, ingénieur son, ingénieur vidéo, etc. Chacun intervient selon son rôle dans la réalisation des dispositifs (cours en ligne, application, module de gestion des étudiants, enregistrement de conférence, etc.). Par ailleurs, ces individus assistent l'enseignant pour la maîtrise et la résolution des problèmes techniques qu'il peut rencontrer.

Le fournisseur de ressources pédagogiques : conçoit, développe et met à disposition de l'enseignant et de l'ingénieur pédagogue des ressources pédagogiques (numérique ou non) pouvant être utilisées lors de l'exécution d'un module. Il procure les composants « métier », i.e., les différentes briques directement liées à la fonction de base de la plateforme, l'enseignement.

- Acteurs intervenant dans la phase d'utilisation

Ce sont les acteurs qui vont utiliser le dispositif de formation en ligne, on citera :

Le tuteur : aide à faire progresser les apprenants dans leur processus de formation en ligne. Selon [Paquette, 98], le tuteur est (1) un formateur dans le sens où sa fonction est de faciliter l'apprentissage et (2) un gestionnaire permettant de gérer les apprenants et les environnements.

L'administrateur : chargé d'entretenir l'environnement technique : assure l'installation et la maintenance du système, gère les droits d'accès, crée des liens vers d'autres systèmes et ressources externes (dossiers administratifs, catalogues, ressources pédagogiques, etc.).

Nous définissons ci-dessous l'apprenant qui est le premier bénéficiaire de l'élaboration du dispositif de formation en ligne :

L'apprenant : le premier bénéficiaire de l'élaboration du dispositif d'apprentissage, il ne peut y avoir de formation sans apprenant. Ses activités peuvent être la consultation en ligne ou le téléchargement des contenus pédagogiques, la participation à des activités d'apprentissage en ligne (activités individuelles ou collaboratives), l'échange des données, la résolution des exercices, l'auto-évaluation et la transmission des travaux au tuteur pour une éventuelle correction.

## 5. Des objets d'apprentissage aux langages de modélisation pédagogique

### 5.1. Le LOM

Une première approche consistait à la mise à disposition « d'un stock de connaissances » permettant de partager et de réutiliser des documents numériques pédagogiques, cependant, ces objets pédagogiques se

trouvaient noyés dans une immense masse d'informations contenue dans le web, les retrouver est une tâche très compliquée et coûteuse en temps et en argent.

Pour ce faire, il fallait les indexer. C'est alors que s'est sentie la nécessité de définir judicieusement ces objets pédagogiques.

IEEE, Institute of Electrical and Electronics Engineers, définit un objet pédagogique (LO, Learning object) comme :

« Toute entité numérique ou non, qui peut-être utilisée, réutilisée ou référencée lors d'une formation dispensée à partir d'un support technologique. »

Il proposa un standard, le LOM (Learning Object Metadata), permettant la description des objets pédagogiques en proposant 45 éléments descriptifs de premier niveau regroupés en neuf catégories

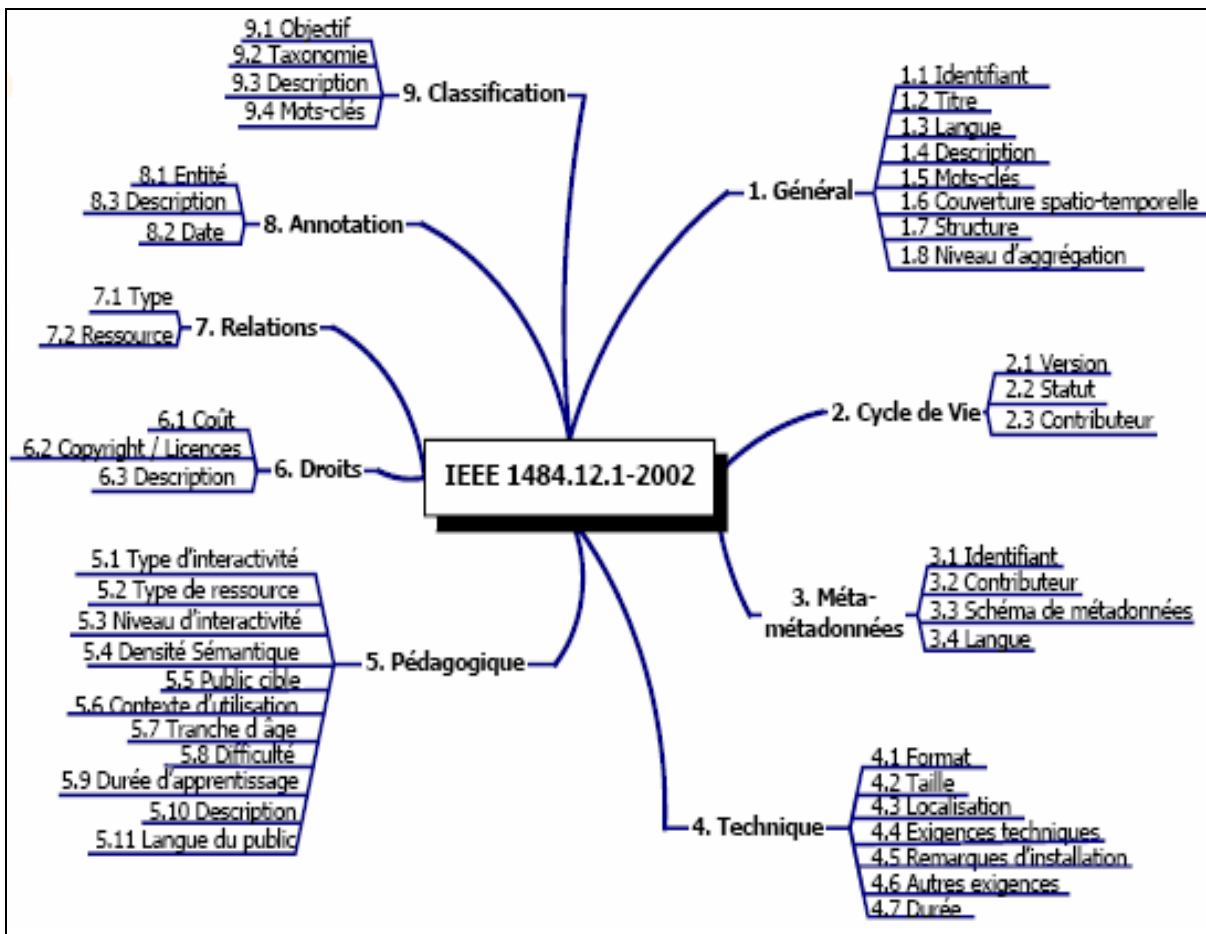


Figure I.2 : Catégories LOM (version 1.0)

- **Générale** (8 éléments) : englobe les éléments relatifs à la description du document pédagogique dans son ensemble. On trouve les caractéristiques comme le titre, la langue du document, une description et des mots clefs, le niveau de granularité (média, leçon, cours, curriculum) ;
- **Cycle de vie** (3 éléments) : contient des informations sur l'historique, l'état actuel du document et les différentes contributions effectuées ;

- **Métamétadonnées** (4 éléments) : représente un ensemble de métadonnées sur les métadonnées décrivant l'objet pédagogique ;
- **Informations techniques** (7 éléments) : définit les exigences techniques d'une ressource, comme le format, la taille, l'emplacement de la ressource en vue de son téléchargement, les contraintes techniques d'utilisation telles que la version du navigateur ou du système d'exploitation et des informations complémentaires sur l'installation ;
- **Informations pédagogiques** (11 éléments) : regroupe des informations de nature pédagogique telles que :
  - Le type d'interactivité ;
  - le type de document (exercice, simulation, questionnaire à choix multiples, etc.) ;
  - le niveau d'interactivité ;
  - la densité sémantique ;
  - le rôle de l'utilisateur de la ressource (professeur, apprenant ou administrateur) ;
  - le contexte pédagogique (depuis l'éducation primaire jusqu'au niveau post graduation) ;
  - le niveau de difficulté ;
  - la durée pédagogique (le temps dont a besoin un apprenant typique).
- **Droits** (3 éléments) : exprime les conditions d'usage des ressources pédagogiques ;
- **Relation** (2 éléments) : couvre les caractéristiques exprimant les liens avec d'autres objets pédagogiques et la nature de ces liens ;
- **Annotation** (3 éléments) : regroupe des informations et des commentaires sur l'utilisation pédagogique de la ressources, en précisant l'origine et la date de ces commentaires ;
- **Classification** (4 éléments) : propose un schéma ouvert, permet d'indiquer où se place l'objet pédagogique dans une classification donnée (niveau de compétences, discipline, idée, restriction, objectif pédagogique, pré requis, type d'accès, niveau d'études, etc.) en associant des mots clés et une description.

## 5.2. Limites du LOM

De nouvelles exigences sont apparues telles que :

- l'intégration au sein d'un même modèle des entités de niveaux conceptuellement différents : les ressources nécessaires à la mise en place d'activités pédagogiques et les activités elles-mêmes.

Selon Jean-Philippe Pernin dans son article intitulé « *À propos d'objets pédagogiques* », la définition d'un LO proposée par IEEE permet de considérer comme objet pédagogique un document imprimé, un cours, un exercice, une étude de cas, une présentation, mais également une salle de cours, un rétroprojecteur, etc.

- une certaine incohérence entre la définition générique des objets pédagogiques proposée par IEEE et les éléments permettant de les décrire (en particulier, il est complexe de décrire des entités non numériques).

### 5.3. SCORM

Une des principales actions du consortium ADL (Advanced Distributed Learning) du Département de Défense américain était de fournir un modèle de référence permettant de garantir la qualité des contenus en termes de :

- Réutilisabilité : facilement modifiable et utilisable par différents outils de développement
- Accessibilité : peut être recherché et rendu disponible aussi bien par des apprenants que des développeurs
- Interopérabilité: peut fonctionner sur une grande palette de matériel, plates-formes, systèmes d'exploitation, navigateurs Web, etc.
- Durabilité ou Pérennité : ne requiert pas d'importantes modifications avec les nouvelles versions des logiciels.

SCORM acronyme de *Sharable Content Object Reference Model*, était la « norme » proposée, elle soulève certaines lacunes du LOM en proposant deux types de recommandation :

- Le modèle d'agrégation de contenu (*Content Aggregation Model*).
- L'environnement d'exécution (*Run-Time Environment*).

#### 5.3.1. Modèle d'agrégation de contenu

Identification des ressources de base et leur agrégation dans un contenu structuré de formation.

Pour ce faire SCORM définit trois niveaux de composants :

- la ressource numérique élémentaire (*asset*) : brique élémentaire qui pourrait être un document simple (image JPEG ou GIF, son WAV ou MP3, page web) ou tout ensemble d'informations pouvant être délivré vers un client Web (document Flash, code Javascript, applet Java, etc.)
- l'objet de contenu partageable (*SCO : Sharable Content Object*), ensemble cohérent d'*assets* qui peut être composé par exemple de pages HTML, d'animations, de dessins, de vidéos.... en d'autres termes un *SCO* est une unité (ou grain) de contenu qui possède un sens pédagogique, et qui peut être réutilisée en l'agrégeant dans d'autres unités d'apprentissage (un autre cours).

Plusieurs SCOs pourront former un Learning Object (LO) et un ou plusieurs LO pourront former un cours, qui sera reconnaissable par une plate-forme SCORM.

- les agrégats de contenu: (*Content Aggregation*) est un ensemble de ressources pédagogiques structuré de façon cohérente au sein d'une entité de plus haut niveau, telle qu'un cours, un chapitre, un module, etc.

Les ressources pédagogiques peuvent être aussi bien des assets que des SCOs. C'est au niveau des agrégats de contenu qu'il est spécifié l'activité attendue grâce à un graphe d'enchaînement des SCOs.

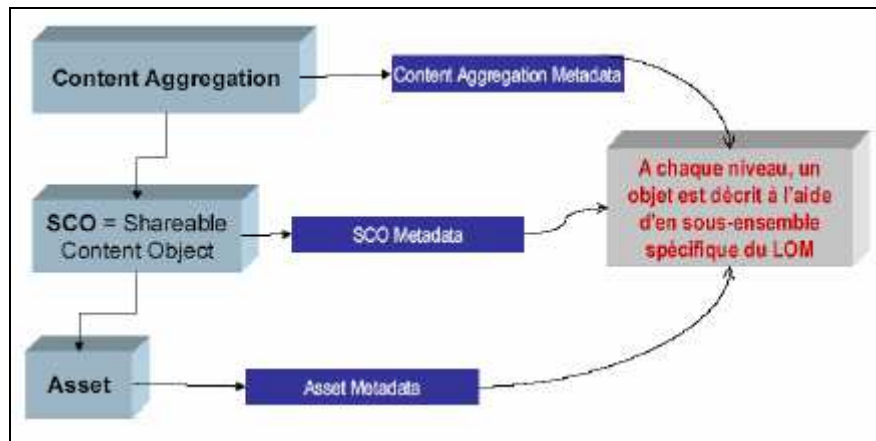


Figure I.3 : Modèle d'agrégation de contenu de SCORM [Pernin, 2004]

### 5.3.2. Environnement d'exécution

Un environnement d'exécution permet d'avoir une perception de l'activité de l'apprenant depuis un LMS (cf. **Figure I. 4**). Il fournit les directives et les recommandations aux développeurs afin qu'ils puissent intégrer, exploiter et garder les traces sur les divers objets pédagogiques manipulés par un LMS.

Au moment de l'exécution, le principe de fonctionnement et de communication est le suivant :

- L'apprenant reçoit un objet SCO via un navigateur ;
- Le SCO contient le contenu pédagogique plus un composant de contrôle pour déterminer l'état de l'objet manipulé ;
- Etablissement de l'état actuel du SCO après chaque manipulation de ce dernier. Cet état peut prendre une des valeurs suivantes : *passed*, *completed*, *failed*, *incomplete*, *browsed* et *not attempted* ;
- Détermination de l'état d'avancement de l'apprenant à travers la valeur retournée.

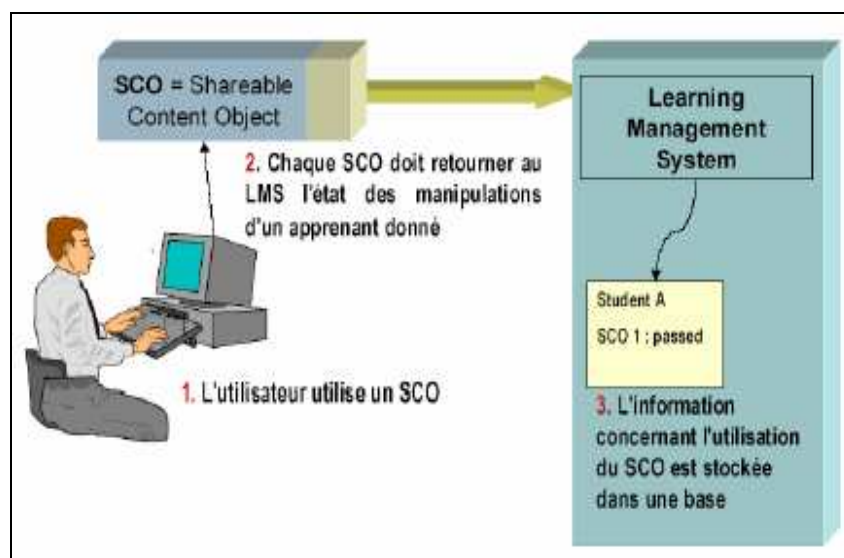


Figure I.4: Environnement d'exécution de SCORM [Pernin, 2004]

## 5.4. Apports et limites du SCORM

Introduction de la notion d'activité qui est représentée de deux façons :

- La spécification au sein des agrégats de contenu l'activité attendue grâce à un graphe d'enchaînement des SCOs.
- Exiger des SCOs qu'ils contiennent des composants de contrôle qui renseignent sur l'activité de l'apprenant vis-à-vis du contenu.

Cependant, pour une large exploitation et réutilisation des SCOs dans des conditions variées, la séparation entre le contenu pédagogique et le contrôle de l'activité de l'apprenant est devenue une exigence.

D'autres faiblesses du modèle proposé par SCORM [Pernin, 2006] :

- Les concepts introduits par la norme SCORM (les agrégats de contenu) sont beaucoup plus considérés comme des concepts d'agrégation, le concept d'activité de l'apprenant n'est pas défini de façon explicite. SCORM privilégie donc une approche centrée davantage sur les contenus que sur les activités.
- Par la mise en place par SCORM d'un protocole élémentaire de suivi de l'apprenant en termes de navigation et d'évaluation, il favorise un certain type d'apprentissage fondé sur la consultation de ressources et la vérification de connaissances ou compétences à l'aide de tests, il n'y a pas de place à des démarches d'apprentissage plus constructivistes, à des modalités de suivi et d'évaluation qui prennent en compte les échanges entre apprenants, leurs productions ou la nature de leurs manipulations.

Nous pouvons dire en résumé que :

LOM et SCORM sont centrés sur les contenus et favorisent généralement un mode d'apprentissage basé sur la structuration de ces contenus en leçons, cours, modules, etc.

Le suivi introduit par SCORM est axé sur la consultation de ressources pédagogiques.

Les concepts employés favorisent cette structuration et ne permettent pas de décrire de façon explicite les activités de l'apprenant.

## 5.5. IMS –LD

Rob Koper de l'Open University of the Netherlands (OUN) propose un point de vue qui se démarque des précédents en affirmant que ce sont les activités et non les objets de connaissance qui constituent le concept-clé de la réussite d'un environnement d'apprentissage [Koper, 2001].

Il énonce un ensemble de propriétés qui doivent être vérifiées pour un tel environnement :

- un environnement d'apprentissage n'est ni un cours ni un programme de cours,
- au sein d'un environnement d'apprentissage, ce sont davantage les activités qui sont centrales que les objets,
- un environnement d'apprentissage centré sur la connaissance est plus qu'une simple encyclopédie,

- un environnement d'apprentissage centré sur les compétences est plus qu'un simple environnement d'exercices,
- au sein d'un environnement d'apprentissage, personne n'apprend de la même façon,
- au sein d'un environnement d'apprentissage, le suivi individuel de chaque apprenant est nécessaire,
- il est nécessaire de passer d'une conception de l'enseignement vers une conception de l'apprentissage.

#### 5.5.1. Educational Modelling Language (EML)

Koper propose de décrire les situations d'apprentissage à l'aide d'un Langage de Modélisation Pédagogique (EML : Educational Modelling Language).

Le modèle EML repose au plus haut niveau sur la structuration des situations d'apprentissage en "Unités d'étude" (*Units of Study*). Typiquement, une Unité d'Étude (UE) peut être un cours, une leçon, une étude de cas, un travail pratique, etc. EML place l'activité au centre du dispositif. L'unité principale de structuration, l'unité d'étude, doit répondre aux contraintes suivantes :

- Une UE correspond à un objectif pédagogique précis et nécessite un certain nombre de prérequis.
- Une UE est composée d'un ensemble d'activités.
- Une activité est réalisée par un ou plusieurs acteurs tenant chacun un rôle.
- Un acteur peut être un apprenant ou un membre de l'équipe pédagogique.
- Une activité est réalisée à l'aide d'un ensemble de ressources (ou objets pédagogiques) accessibles au sein d'un environnement.

La figure suivante montre une vue simplifiée du modèle EML :

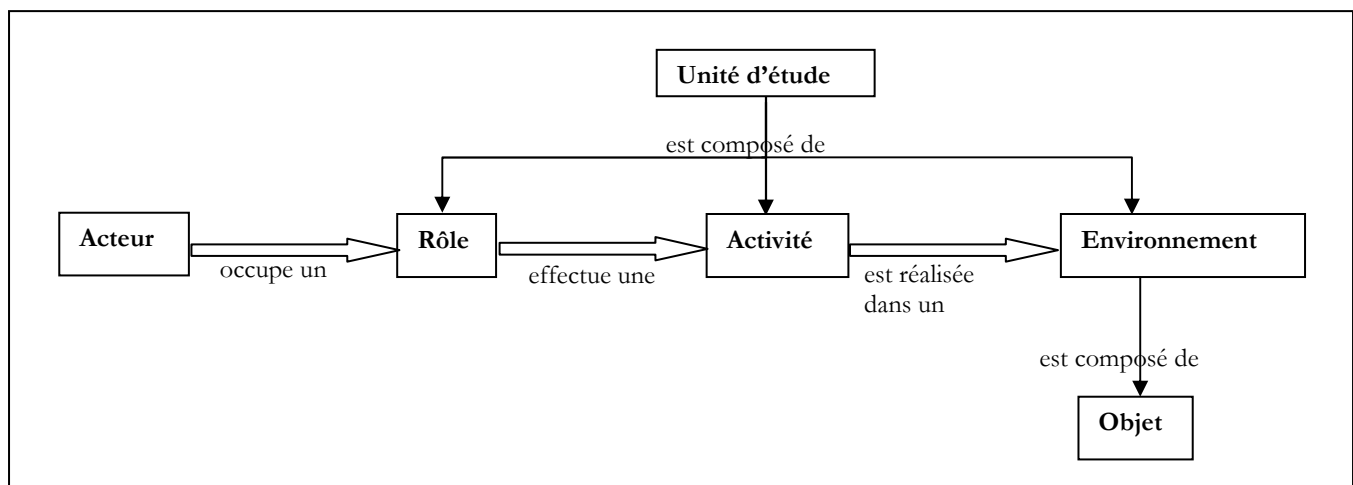


Figure I.5 : Vue simplifiée du modèle EML

#### 5.5.2. Avantages d'EML

Parmi les avantages d'EML on peut citer [Koper, 2001] :

- Le concept d'activité est explicite et est au centre du dispositif.
- Le modèle est ouvert : il peut s'appliquer à des processus d'apprentissage traditionnels ou ceux basés totalement ou partiellement sur les technologies numériques ;

- Les activités et les ressources sont séparées de façon explicite, il devient envisageable de disposer de « patrons d'activité » pouvant être réutilisés dans des contextes différents et avec des ressources différentes.

### 5.5.3. Pourquoi une spécification ?

Une spécification e-formation a pour but de faciliter l'écriture et la modélisation d'un cours, de faire travailler des étudiants dessus et de le conserver d'une plate-forme à l'autre. Libre de droits et gratuite, elle peut s'utiliser de manière simple. La seule contrainte est de s'habituer à la manière de concevoir le contenu dans une application selon l'approche proposée par la spécification. Grâce au respect de la spécification, une fois que les contenus sont créés ils perdurent : ce qui a été conçu peut être conservé quelle que soit la plate-forme [Burgos et al., 2006a].

Depuis sa création en 1997, les travaux d'IMS<sup>2</sup> se sont successivement concentrés sur les différents domaines liés à l'ingénierie pédagogique [Lejeune A., 2004]:

IMS Meta-Data : indexation des ressources d'apprentissage

IMS Learner Information Package (IMS-LIP): profil de l'apprenant

IMS Reusable Definition of Competency or Educational Objectives (IMS-RDCEO) : description des compétences

IMS Enterprise : interrelations au sein d'une organisation

IMS Content Packaging (IMS-CP) : interopérabilité des systèmes et des environnements

IMS Question & Test Interoperability (IMS-QTI) : évaluation des connaissances et des compétences

IMS Learning Design (IMS-LD): métamodélisation d'une UA

IMS Shareable State Persistence: interopérabilité des banques de ressources pédagogiques

### 5.5.4. Relations entre IMS-LD et les autres spécifications :

Il existe des relations entre IMS LD et les autres spécifications IMS : Il existe des relations de communication entre les différentes spécifications, pour sa part la spécification IMS LD est considérée comme une couche d'intégration de plusieurs spécifications existantes ; elle peut être étendue par l'utilisation ou l'inclusion d'autres spécifications. Cette extension intervient à plusieurs niveaux dans la modélisation de l'UA [Lejeune A., 2004] :

- Les principes d'IMS Simple Sequencing peuvent être appliqués (a) pour séquencer les ressources d'un objet d'apprentissage et (b) pour ordonnancer les différents objets d'apprentissage et les services dans un environnement ;
- Les métadonnées d'IMS/LOM Meta-Data peuvent être incluses à l'emplacement correspondant dans de nombreuses structures d'une UA IMS LD.
- Les tests d'évaluation que peut contenir une UA peuvent être modélisés selon la spécification Question and Test Interoperability, puis intégrés en tant que schéma indépendant ou comme ressource spécifique.
- Les objectifs d'apprentissage et les prérequis d'une unité peuvent faire référence à des ressources conformes à IMS Reusable Definition of Competency or Educational.

<sup>2</sup> IMS : Instructional Management Systems



- La structure des propriétés permettant la personnalisation d'une UA peut être entièrement assurée par IMS Learner Information Package.
- IMS Entreprise offre des moyens pour l'affectation des rôles modélisés avec IMS LD.
- Une UA peut être définie par l'inclusion d'un « Learning Design » dans un IMS Content Package, en lieu et place de l'élément *Organisation* (spéc. IMS Content Packaging) (cf. **Figure I.6**), comme nous le verrons dans le deuxième chapitre.

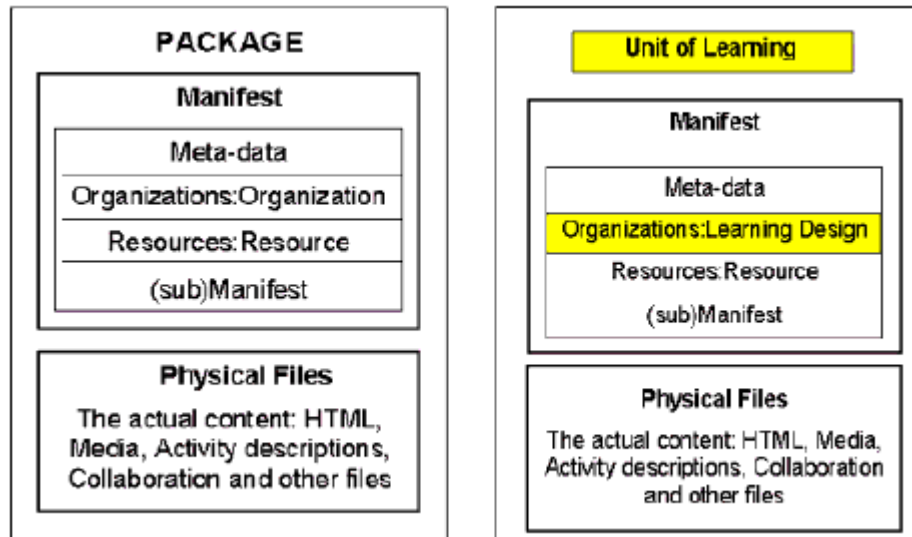


Figure I.6 : Inclusion d'un LD dans un IMS CP [Lejeune A., 2004]

Il est à noter que l'intégration de ces différentes spécifications obéit aux mécanismes d'espaces de noms XML, toutes les spécifications IMS ayant leur propre espace de nom.

### 5.5.5. La spécification IMS-LD

Parmi celles-ci, IMS-Learning Design (ingénierie pédagogique), incorpore la flexibilité pédagogique et complète certains aspects traités par les autres [Burgos et al, 2006a].

Le consortium nord américain IMS a entrepris d'étudier et de fournir une spécification d'un langage EML répondant à la définition donnée par le Comité Européen de Normalisation (CEN) comme :

*"Un modèle d'information et d'agrégation sémantique, décrivant les contenus et les processus engagés dans une UA selon une perspective pédagogique et dans le but d'assurer la réutilisabilité et l'interopérabilité".*

Très largement inspirée du langage EML développé par R. Koper c'est en février 2003, que la spécification Learning Design V1.0 (IMS-LD) voit le jour.

IMS-LD a été publié en 2003 par le IMS/GLC. (Instructional Management Systems Global Learning Consortium : Consortium pour l'apprentissage global avec les systèmes de gestion de formations, le nom original, quand IMS a débuté en 1997 était Instructional Management Systems project).

IMS-LD n'impose pas de modèle pédagogique particulier (neutralité pédagogique) et peut être utilisée avec un grand nombre de scénarios et de modèles pédagogiques, ce qui prouve sa flexibilité ; IMS-LD est souvent appelé un méta-modèle pédagogique.

La spécification IMS Learning Design permet la formalisation dans un langage informatique (XML) de scénarios d'apprentissage très variés, elle est divisée en trois niveaux de complexité, chacun reposant sur le niveau inférieur en y ajoutant certaines fonctionnalités.

Le **niveau A** : comprend la base du système, permet de définir le rôle et les activités de chacun des acteurs du scénario pédagogique : les acteurs jouent différents rôles (*roles*) afin d'accomplir certains objectifs (*objectives*) en ayant des activités de support (*support-activities*) ou d'apprentissage (*learning-activities*) (suivant leur rôle – apprenant ou tuteur). Les activités peuvent en outre comprendre un ou plusieurs environnements (*environements*), qui consistent en des objets (*objects*) et/ou en services (*services*) (tels qu'un forum de discussion par exemple).

Le **niveau B** ajoute à ces concepts ceux des propriétés (*properties*) (variables) et de conditions (*conditions*). Une propriété doit être remplie pour qu'une activité soit considérée comme terminée – et inversement, l'achèvement d'une activité peut influencer une propriété. L'utilisation de conditions permet de déclencher certains événements (tels l'apparition d'un contenu ou l'assignation d'une valeur à une propriété) seulement sous certaines conditions.

Le **niveau C** ajoute la possibilité d'avoir recours à des *notifications* qui permettent la notification automatique (par courrier électronique) d'un événement à un rôle, et aussi de déclencher ou de remplir une propriété ou une condition.

#### 5.5.6. Modèle conceptuel

Le modèle conceptuel (relationnel) s'apparente au « script » de la situation d'apprentissage. Il précise quelle activité utilise quel environnement et quel rôle fait quelle activité [Lejeune A., 2004].

Il explique les relations de composition des éléments les uns par rapport aux autres, La représentation graphique choisie pour chacun de ces modèles est celle des diagrammes de classe UML<sup>3</sup> (*Unified Modelling Language*).

---

<sup>3</sup> Indications de lecture des diagrammes de classe UML :

- un losange vide exprime une relation d'agrégation : la classe *composant* est une collection de classes *ressource* ;
- un losange plein exprime une relation d'agrégation contrainte (composition) : la classe *pièce* est une collection de classes *acte*, lesquelles ne peuvent être partagées par plusieurs instances de la classe *pièce* ;
- un triangle désigne la spécialisation (sous typage) de la classe mère en plusieurs classes aux attributs spécifiques : une *ressource* peut être un *service*, un *dossier*, une *personne*, etc. ;

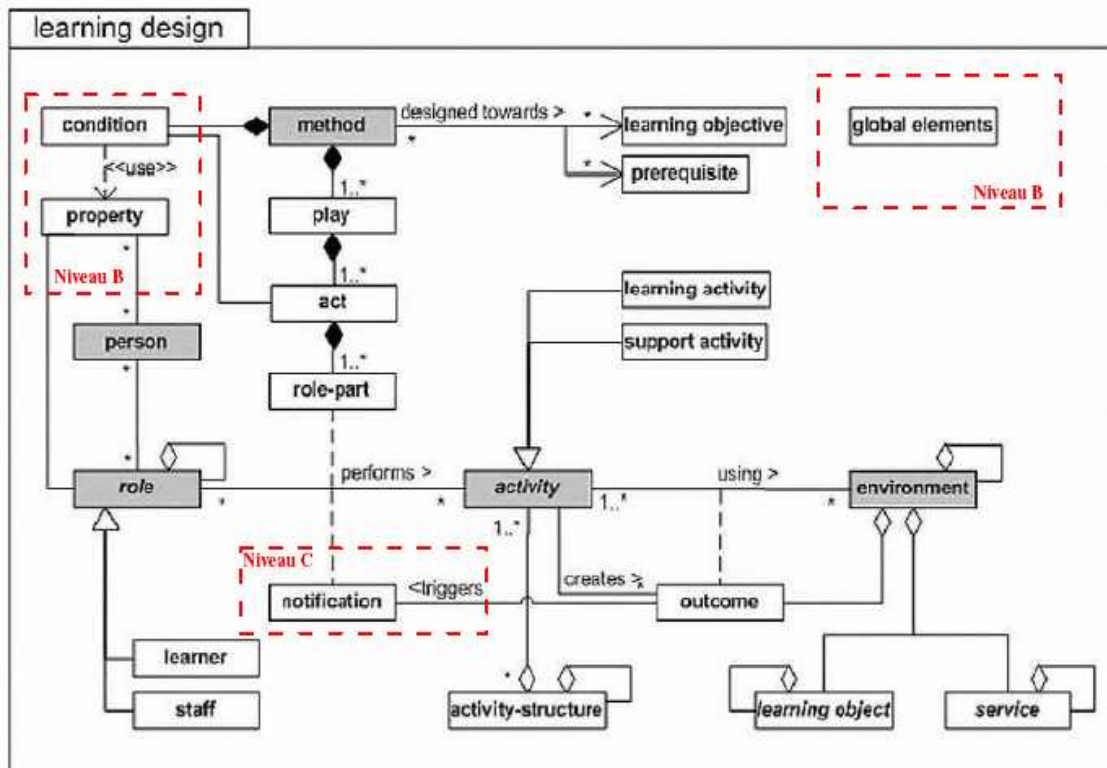


Figure I.7 : Le schéma du modèle conceptuel (relationnel) d'IMS-LD [IMSLD03]

Le concept clé de la spécification IMS-LD est le fait qu'indépendamment d'une approche pédagogique, une personne occupe un rôle (*role*) dans le processus d'apprentissage, typiquement un rôle d'apprenant (*learner*) ou de formateur (*staff*). Cette personne vise à obtenir certains résultats (*outcome*) par la réalisation d'activités d'apprentissage (*learning activity*) et/ou de support (*support activity*) plus ou moins structurées (*activity-structure*) dans un environnement (*environment*). L'environnement consiste en des objets d'apprentissage (*learning object*) et des services (*service*) appropriés qui seront utilisés durant la réalisation des activités. Quand et quelles activités les rôles vont accomplir est déterminé par la méthode (*method*) ou par la notification (*notification*). La méthode est conçue en vue de répondre aux objectifs d'apprentissage (*learning objective*) et présuppose certains prérequis (*prerequisite*). Elle consiste en un ou plusieurs pièces (*play*) concurrents. Une pièce consiste en un ou plusieurs actes (*act*) séquentiels. Chaque acte est décomposé ensuite en un ensemble de partitions (*role-part*) qui se déroulent parallèlement. Les activités se déroulent dans un environnement comprenant des services (chat, forum, ...) et des ressources de contenu.

En résumé, nous pouvons dire qu'il existe deux grandes approches :

- 1) une approche "documentaliste" fondé sur l'indexation et l'agrégation d'objets d'apprentissage (LOM et SCORM).
- 2) une approche d'ingénierie pédagogique centrée sur les processus IMS LD.

## **Chapitre II : Construction d'une unité d'apprentissage en utilisant la spécification IMS-LD**

Ce second chapitre aborde la spécification IMS LD, beaucoup plus en détail afin de mieux comprendre la structure des éléments composant le modèle IMS LD tout en précisant leurs rôles respectifs tirés de l'IMS Learning Design Information Model. Pour finir, ce chapitre présente les étapes par lesquelles passent la création et l'implémentation d'un Learning design utilisant la spécification IMS-LD.

### 1. Introduction

Le Learning Design tel que modélisé par la spécification IMS-LD est une structure XML, un peu complexe, que nous présentons en détail, tout en précisant le rôle de chacun des éléments du Learning Design.

Nous examinons la structure, l'organisation et le fonctionnement du Learning Design, ce qui va nous permettre d'analyser les éléments du LD et de dégager ainsi une première analyse à la problématique qu'on s'est proposée d'étudier.

### 2. Modèle structurel

En reprenant la métaphore théâtrale, le modèle structurel décrit la situation d'apprentissage d'un point de vue logistique et organisationnel. Il spécifie que le décor (environnement) doit comporter tels ou tels éléments (services, objets d'apprentissage,...), que la pièce est découpée en un certain nombre d'actes, chaque acte distribuant les activités à chacun des rôles.

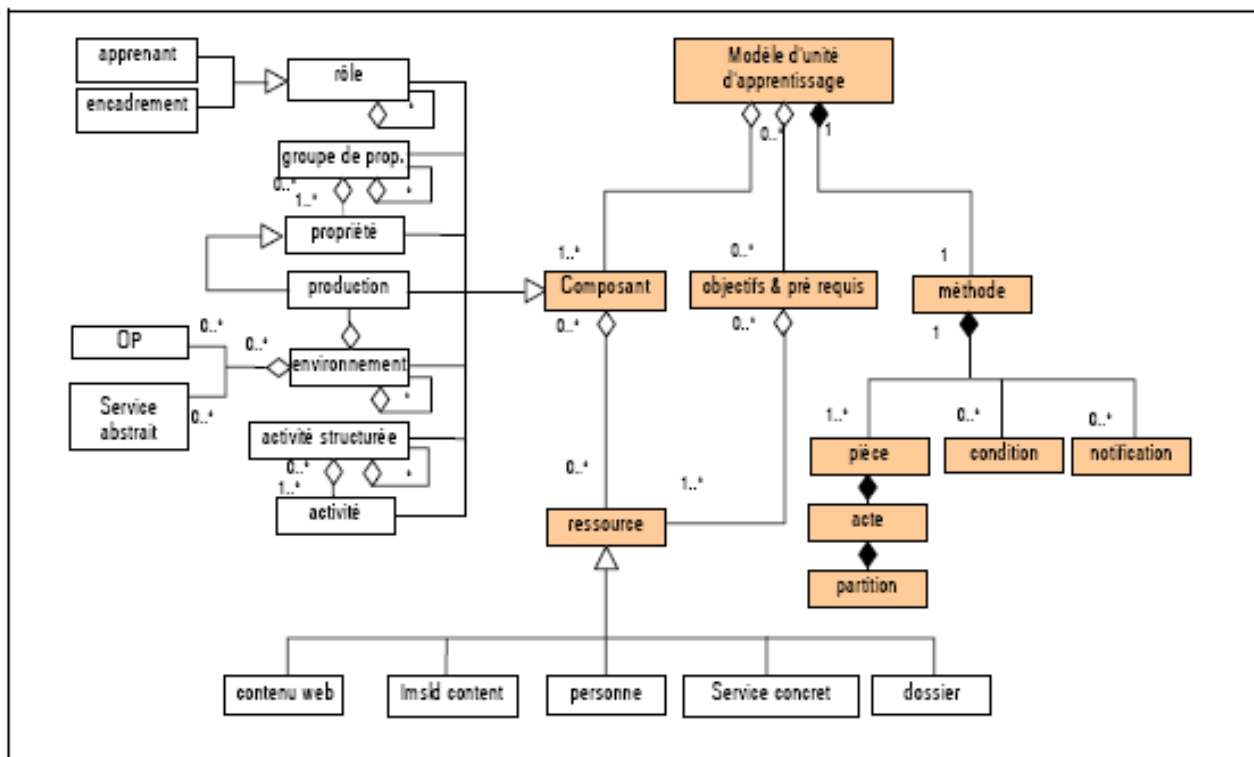


Figure II.1 : Schéma de structuration générale d'IMS-LD [IMSLD03]

Une métaphore du modèle IMS LD ou plus précisément un élément Learning design serait une recette de cuisine où les ingrédients nécessaires à la préparation de la recette seraient les composants (*components*) constituant ainsi la partie **statique** du modèle, quant à la préparation proprement dite de la recette, la façon de combiner les ingrédients, de séquencer les actes de préparation serait la méthode (*method*) ou encore le scénario d'apprentissage constituant ainsi la partie **dynamique** du modèle.

### 3. La spécification IMS-LD

#### 3.1. Structure de base d'un Learning Design – niveau A

L'élément *learning-design* traduit la conception du Design Pédagogique faite par l'auteur et/ou l'ingénieur pédagogique en utilisant la spécification IMS-LD.

C'est l'élément racine à partir duquel viennent s'emboîter tous les éléments mis à disposition par la spécification IMS-LD.

La structure du Learning Design au niveau A est la suivante :

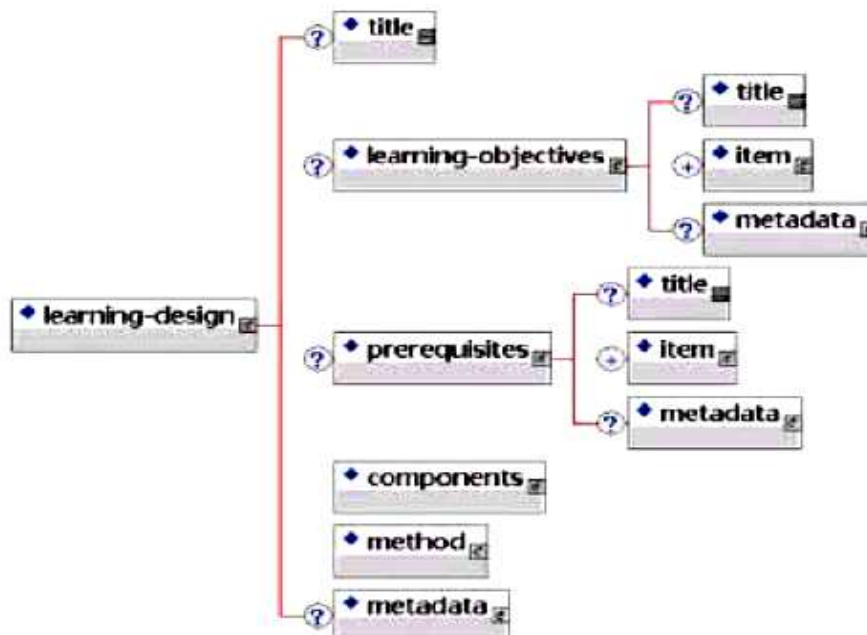


Figure II.2: structure du *learning-design*, au niveau A [IMSLD03]

**Légende :** (valable pour toutes les figures II.2 jusqu'à II.13)  
 + : Obligatoire  
 ? : Optionnel  
 \* : Un ou plusieurs

Les *components* : décrivent tous les composants du Learning Design : les éléments *roles*, *activities* et *environments* utilisés par l'élément *method*. (qu'on verra par la suite)

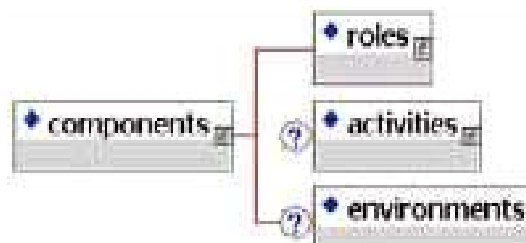


Figure II.3 : structure du modèle *components*, au niveau A [IMSLD03]

Les différents acteurs du scénario sont associés à des *rôles*.

Les rôles sont divisés en rôle d'apprenants (*learner role*) et en rôle de tuteurs (d'encadrement) (*staff role*).

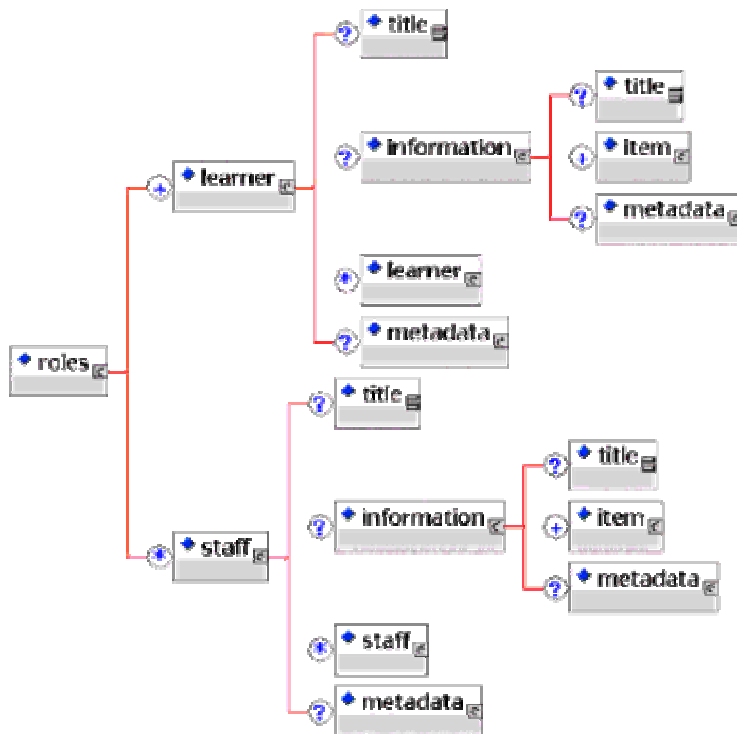


Figure II. 4 : structure du modèle *roles*, au niveau A [IMSLD03]

- La *method* est une description précise du scénario d'apprentissage. Elle consiste en un ou plusieurs *plays*, qui sont eux-mêmes composés d'un ou plusieurs *acts*.

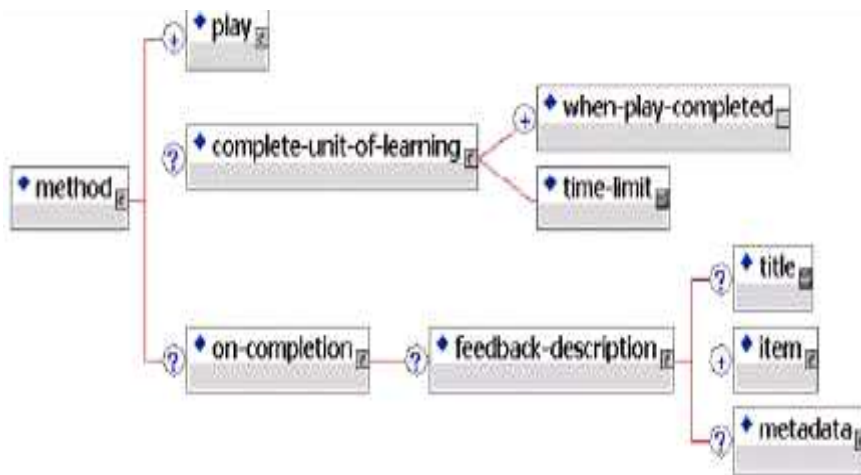


Figure II.5 : structure du modèle *method*, au niveau A [IMSLD03]

- Le scénario d'apprentissage est composé d'un ou de plusieurs pièces de théâtre (*plays*): chaque play est indépendant des autres, il représente des parties du processus d'apprentissage indépendants.

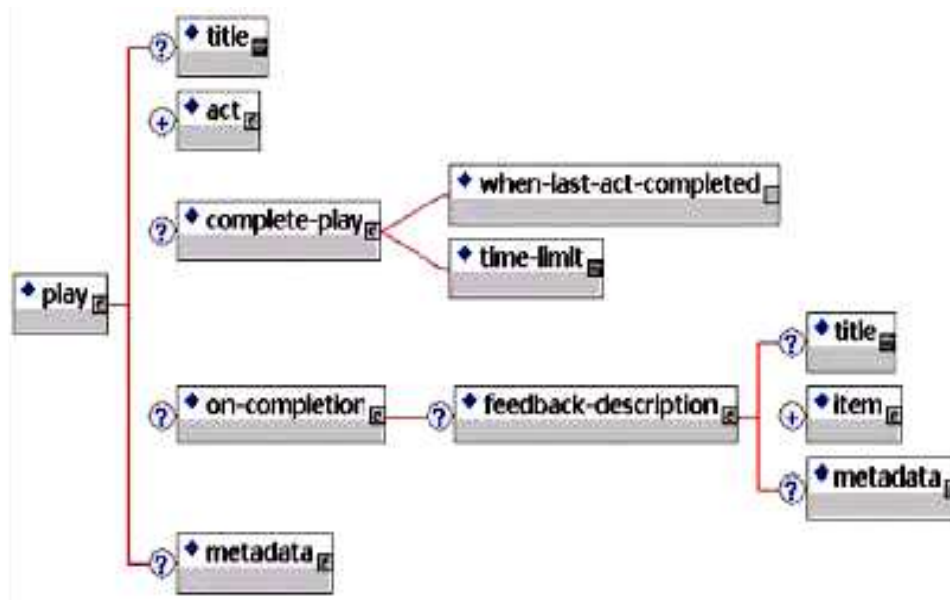


Figure II.6 : structure du modèle *play*, au niveau A [IMSLD03]

Chaque *play* est composé d'un ou de plusieurs *acts*.

Tous les actes commencent et se terminent en même temps pour tous les acteurs ayant un rôle à jouer dans le *play* concerné.

- ✚ Un *act* doit être terminé (un *act* se termine en même temps pour tous les acteurs du scénario) pour rendre disponible le suivant aux acteurs.

Ces actes sont constitués d'un ou de plusieurs *role-parts* (partitions), au cours desquels un *role* est associé à un des éléments suivants :

- une activité (*activity*), une activité d'apprentissage (*learning-activity*) ou une activité de support (*support-activity*)
- une unité d'apprentissage (*unit-of-learning*)
- un ensemble d'activités rassemblées en une structure d'activité (*activity-structure*)
- un environnement (*environment*)



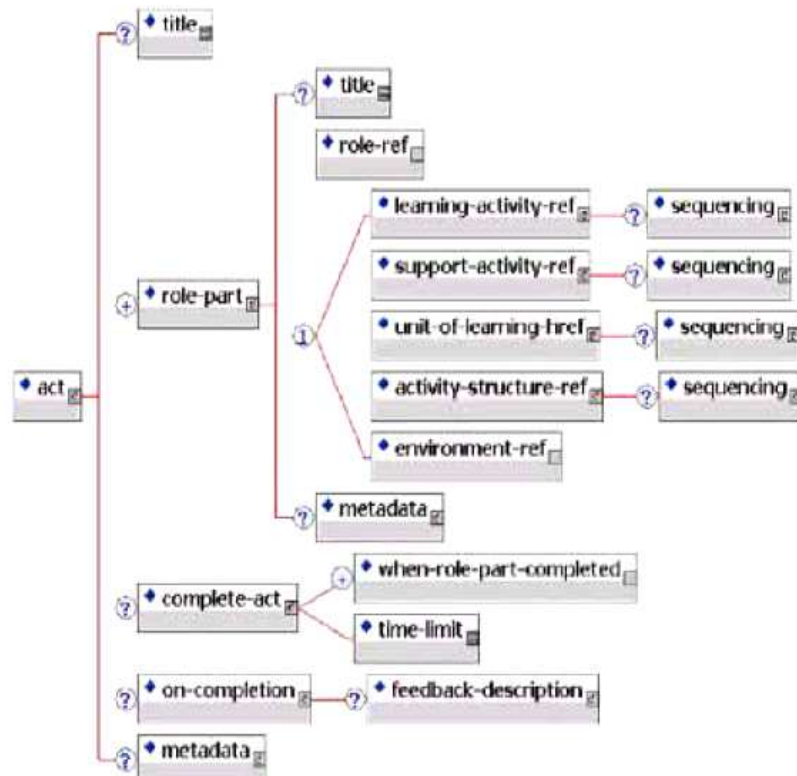


Figure II.7 : structure du modèle *act*, au niveau A [IMSLD03]

L'élément *act* est donc une succession (pas forcément une séquence) de *role-parts* au cours desquels les différents rôles se voient attribuer des *activités*.

Les *activités* peuvent être de deux types :

- *learning-activity* : destinées à un rôle d'apprenant et consistant en du matériel d'apprentissage ou des directives.
- *support-activity* : destinées à être associées à un rôle de « *staff* », vise un ou plusieurs rôles, et peut être répétée pour chaque acteur du ou des rôles visés.

Il est en outre possible de regrouper des activités par thème (ou par n'importe quelle autre critère) en *activity-structures* – elles-mêmes utilisables en tant qu'*activités*. Il existe ainsi formellement trois différents types d'*activités* :

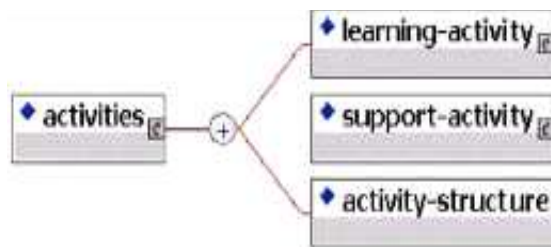


Figure II.8 : structure du modèle *activities*, au niveau A [IMSLD03]

Les *environments* peuvent être associés à une *activity*, ainsi, un *role-part* peut alors associer un *rôle* avec une *activity*, elle-même associée à un (ou plusieurs) *environments*.

On peut donc créer une activité d'apprentissage (ou de support) complète en liant des *environments* à une *activity*, de manière à fournir les ressources (tels que des outils ou objets d'apprentissage) nécessaires à l'achèvement de cette activité.

Les *environments* permettent de référencer différents types de ressources affichables par un navigateur (*learning-object*), ainsi que d'autres *environments*.

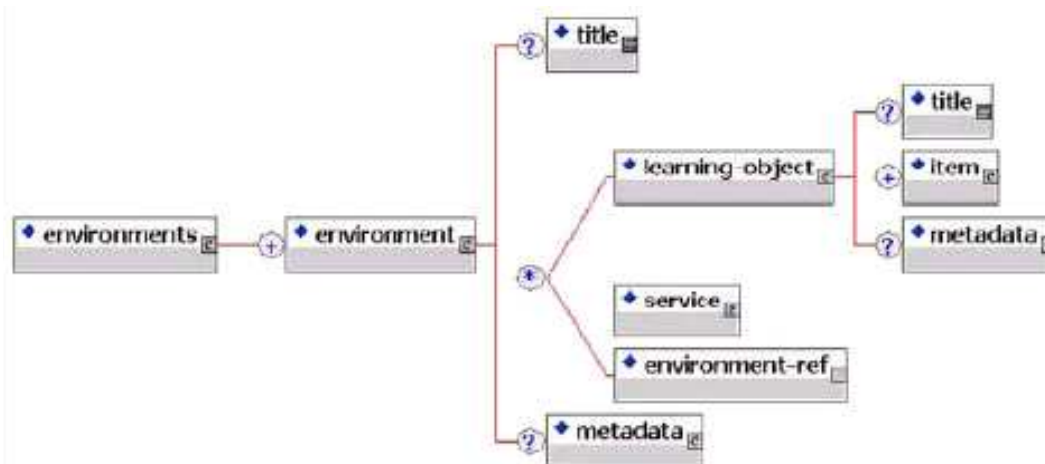


Figure II.9 : structure du modèle *environments*, au niveau A [IMSLD03]

Ils permettent en plus l'implémentation de *services*.

Les services sont des outils prédéfinis par la spécification IMS LD, et peuvent être de trois types différents:

- *send-mail* : utilisé pour l'envoi de courriels aux acteurs de *roles* choisis
- *conference* : utilisé pour gérer les droits des acteurs (selon leur *role*) dans différents types de conférences (*synchronous*, *asynchronous*, *announcement*) ;
- *index-search* : utilisé pour définir des fonctions de recherche au sein du Learning Design.

L'environnement d'exécution doit pouvoir créer ces services d'après les définitions faites dans le Learning Design. Si des outils logiciels externes à celui-ci doivent être utilisés pour la mise en place de forums, par exemple, alors ces définitions doivent permettre à la personne qui met en place l'UA en vue de son exécution de créer le service manuellement de la manière appropriée.

Nous pouvons dire en résumé que :

Le Learning Design de niveau A a la forme d'une pièce de théâtre, où les acteurs jouent des rôles spécifiques. Le scénario consiste en une succession linéaire d'activités prédéfinies, sans possibilité de choix entre différents chemins pédagogiques par l'apprenant.

### 3.2. Learning design Niveau B - propriétés et conditions -

Le niveau B étend le niveau A en rajoutant deux éléments principaux :

- Les propriétés (*properties*)
- Les conditions (*conditions*)

Les *conditions* ont une structure standard *if-then-else*. Les *conditions* sont continuellement réévaluées par l'environnement d'exécution (*run-time environment*).

Les *properties* sont des variables, dont le *datatype* associé peut être *string*, *boolean*, *integer*, *URI*, *datetime*, *file*, *real*, *text*, *duration* ou autre.

Il existe quatre types de *properties* :

1. les propriétés locales (*loc-property*) n'ont d'existence que dans le cadre d'une exécution de l'unité d'apprentissage (*unit of learning*), et ont la même valeur pour tous les utilisateurs.
2. les propriétés globales (*glob-property*) ont la même valeur pour tous les utilisateurs, et sont indépendantes du contexte de l'unité d'apprentissage : elles peuvent être définies dans IMS LD ou à l'extérieur.
3. les propriétés personnelles (globales ou locales : *locpers-property* et *globpers-property*) sont des propriétés globales ou locales qui ont (peuvent avoir) une valeur différente pour chaque acteur du scénario.
4. les propriétés de rôle (*locrole-property*) sont toujours locales et ont la même valeur pour tous les acteurs d'un rôle.

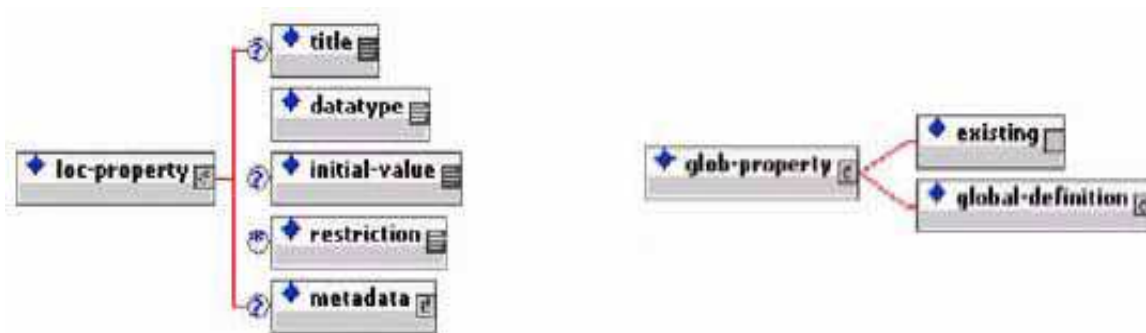


Figure II.10 : structure des modèles *loc-property* et *glob-property*, au niveau B [IMSLD03]

Les *glob-properties* (globales) doivent être *existing* (préexistantes) ou, dans le cas contraire, comprendre une *global-definition* permettant de les définir de manière globalement unique.

Les *loc-properties* (locales) nécessitent d'être définies (cf. **Figure II.10**)

L'ajout de *properties* (variables) et de *conditions* dans le niveau B de la spécification change les modèles définis dans le niveau A comme suit :

1	le modèle <i>components</i> est étendu pour inclure les <i>properties</i>
2	les modèles <i>complete-activity</i> , <i>complete-act</i> , <i>complete-play</i> , <i>complete-unit-of-learning</i> sont étendus pour inclure l'élément <i>when-property-is-set</i> , c'est-à-dire : lorsqu'une <i>property</i> prend une certaine valeur, l'achèvement d'une <i>activity</i> , d'un <i>act</i> , d'un <i>play</i> ou de l' <i>unit of learning</i> peuvent être déclenchées
3	le modèle <i>on-completion</i> est étendu pour inclure l'élément <i>change-property-value</i> , c'est-à-dire : la valeur d'une <i>property</i> peut être modifiée automatiquement à l'achèvement d'une <i>activity</i> / <i>act</i> / <i>play</i> / <i>method</i> ; (c'est l'inverse du point précédent)
4	le modèle <i>email-data</i> est étendu pour accepter les attributs <i>email-property-ref</i> et <i>username-property-ref</i> , qui se réfèrent à des propriétés globales personnelles ( <i>globpers-property</i> ) contenant le nom et l'adresse de courrier électronique d'un acteur.
5	le modèle <i>time-limit</i> est étendu pour accepter l'attribut <i>property-ref</i> , qui se réfère à une <i>property</i> – on peut donc associer la limite de temps (associée à la achèvement d'une <i>activity</i> / <i>act</i> / <i>play</i> / <i>method</i> ) à une <i>property</i> .
6	le modèle <i>method</i> est étendu pour accepter l'élément <i>conditions</i> – c'est l'endroit où toutes les <i>conditions</i> sont exprimées.
7	le modèle <i>complete-act</i> est étendu pour accepter l'élément <i>when-condition-true</i> – c'est-à-dire qu'un <i>act</i> peut se terminer lorsqu'une <i>condition</i> est vérifiée.
8	un groupe séparé, appelé <i>global-elements</i> , est inclus pour lire et régler des <i>properties</i> à partir de différents types de schémas de contenu basés sur le XML (tel XHTML) – ceci sert à laisser des utilisateurs régler ou afficher des <i>properties</i> pendant l'exécution du scénario, ces éléments ne font pas partie du modèle <i>learning-design</i>
9	le modèle <i>services</i> est étendu pour inclure l'élément <i>monitor</i> – il s'agit d'un service (faisant donc partie d'un <i>environment</i> ) permettant l'affichage de <i>properties</i> des acteurs d'un rôle prédéfini par l'intermédiaire des <i>globalelements</i> (cf. point précédent)
10	on peut utiliser l'attribut global <i>class</i> du W3C pour utiliser, dans toutes sortes de schémas de contenu basés sur le XML (tel XHTML), des conditions sur la visibilité d'éléments de contenu.

**Tableau II.1 :** les changements apportés aux modèles de niveau A par l'ajout des *properties*



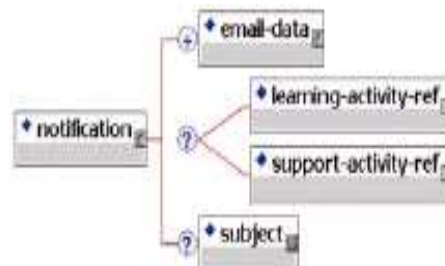


Figure II.12: structure du modèle *notification*, au niveau C [IMSLD03]

Une *notification* consiste donc en l'envoi automatique d'un courrier électronique aux acteurs d'un *role* (par l'entremise du modèle *email-data*, voir Figure 10). Ce courrier électronique a un sujet prédéfini, et contient un lien dirigeant le destinataire directement vers une *activity* (de support ou d'apprentissage).

L'ajout de *notifications* dans le niveau C de la norme permet donc formellement l'altération de la manière suivante des modèles définis dans les niveaux A et B :

- le modèle *on-completion* est étendu pour accepter un élément *notification* – c'est à- dire que la achèvement d'une *activity* / *act* / *play* / *method* peut déclencher une *notification*
- le modèle *then* est étendu pour accepter un élément *notification* – c'est-à-dire que la vérification d'une *condition* peut déclencher une *notification*
- les *global-elements set-property* et *set-property-group* sont étendus pour accepter un élément *notification* (voir Figure 11) – c'est-à-dire qu'on peut utiliser des schémas basés sur le XML (comme XHTML p.ex., voir sous-section 3.1.3) pour laisser des acteurs déclencher eux-mêmes des *notifications*.

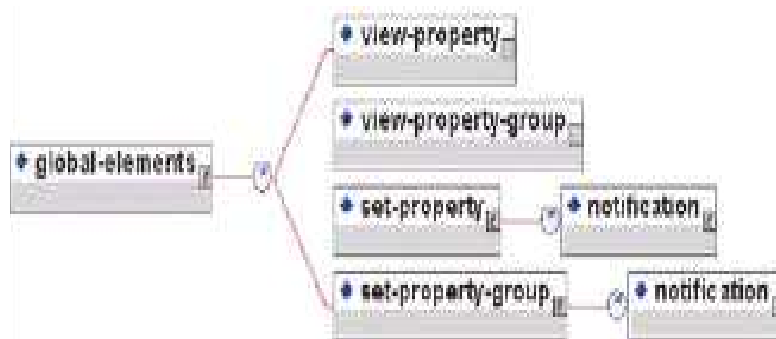


Figure II.13: structure du modèle *global-elements*, au niveau C [IMSLD03]

Nous pouvons dire en résumé que :

Les notifications sont donc un moyen d'avertir les acteurs d'un Learning Design de l'achèvement d'un élément de la *method*, de la vérification d'une *condition* et de la décision d'un acteur.

Elles permettent d'avertir ces acteurs en utilisant le courrier électronique et de les rediriger directement vers une *activity* au sein de l'environnement.

***Il n'est pas obligatoire que ces acteurs se trouvent dans l'environnement d'apprentissage pour qu'ils soient notifiés [IMSLD03].***

## 4. Création du learning design

### 4.1. Processus de modélisation d'une unité d'apprentissage

La modélisation d'une UA suit un processus incrémental en deux phases : une phase d'analyse où le scénario pédagogique est précisé par un texte descriptif, puis une phase de conception en deux étapes, la première s'attachant à représenter le déroulement de l'unité d'apprentissage sous forme d'un diagramme d'activités UML (*Unified Modelling Language*), la seconde formalisant par un document XML structuré conformément à la spécification, l'ensemble de l'unité (composants et schéma d'exécution) [Lejeune A., 2004].

Anne Lejeune et Olga Marino [IMS Learning Design] proposent le processus de modélisation suivant qui passe par trois étapes (cf. Figure II.4):

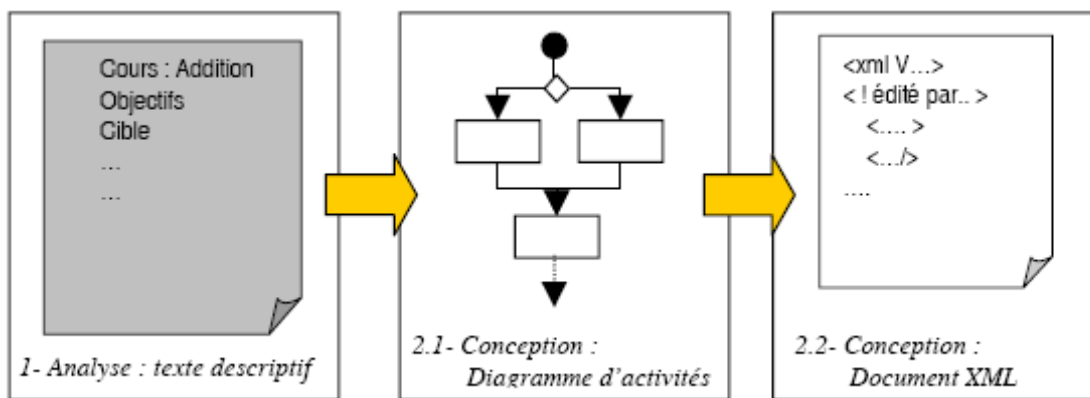


Figure II.14 : Processus de modélisation d'une unité d'apprentissage [IMS Learning Design]

#### 4.1.1. Phase d'analyse (Analyse descriptive)

Analyse du texte descriptif qui consiste à décrire le plus précisément possible le scénario de l'unité d'apprentissage sous une forme textuelle. La narration doit décrire de façon complète une situation d'apprentissage en termes de scénarios et de cas d'usage, du point de vue de l'apprenant et du staff (l'encadrement).

Le document doit être suffisamment détaillé pour permettre d'en déduire un graphe d'enchaînement d'activités par rôle, que le « *Best Pratices and Implementation guide* » propose de modéliser dans la phase de conception en utilisant le formalisme UML de diagrammes d'activités [IMS,LD Best Practice and implementation], on identifiera:

- les rôles
- les activités
- les évènements

#### 4.1.2. Phase de conception

Cette phase passe par deux étapes :

- Production d'un digramme d'activité UML et

- Le document XML conforme IMS-LD.

4.1.2.1. Production d'un diagramme d'activité UML

Le diagramme d'activités représente l'enchaînement chronologique des activités proposées à chacun des rôles. La figure se lit du haut vers le bas, chaque rôle étant matérialisé par une colonne (couloir) du diagramme. Les points de synchronisation sont indiqués par des traits horizontaux épais, l'espace entre deux traits horizontaux matérialisant ainsi les actes de l'unité d'apprentissage [Lejeune A., 2004].

La description du diagramme d'activités à partir du texte descriptif se fait en plusieurs étapes

- Identifier et nommer les activités
- Identifier les différents rôles
- Regrouper si nécessaire les activités dans une structure d'activités ;
- Synchroniser les activités de chaque rôle
- Choix du nombre de scénarios nécessaires

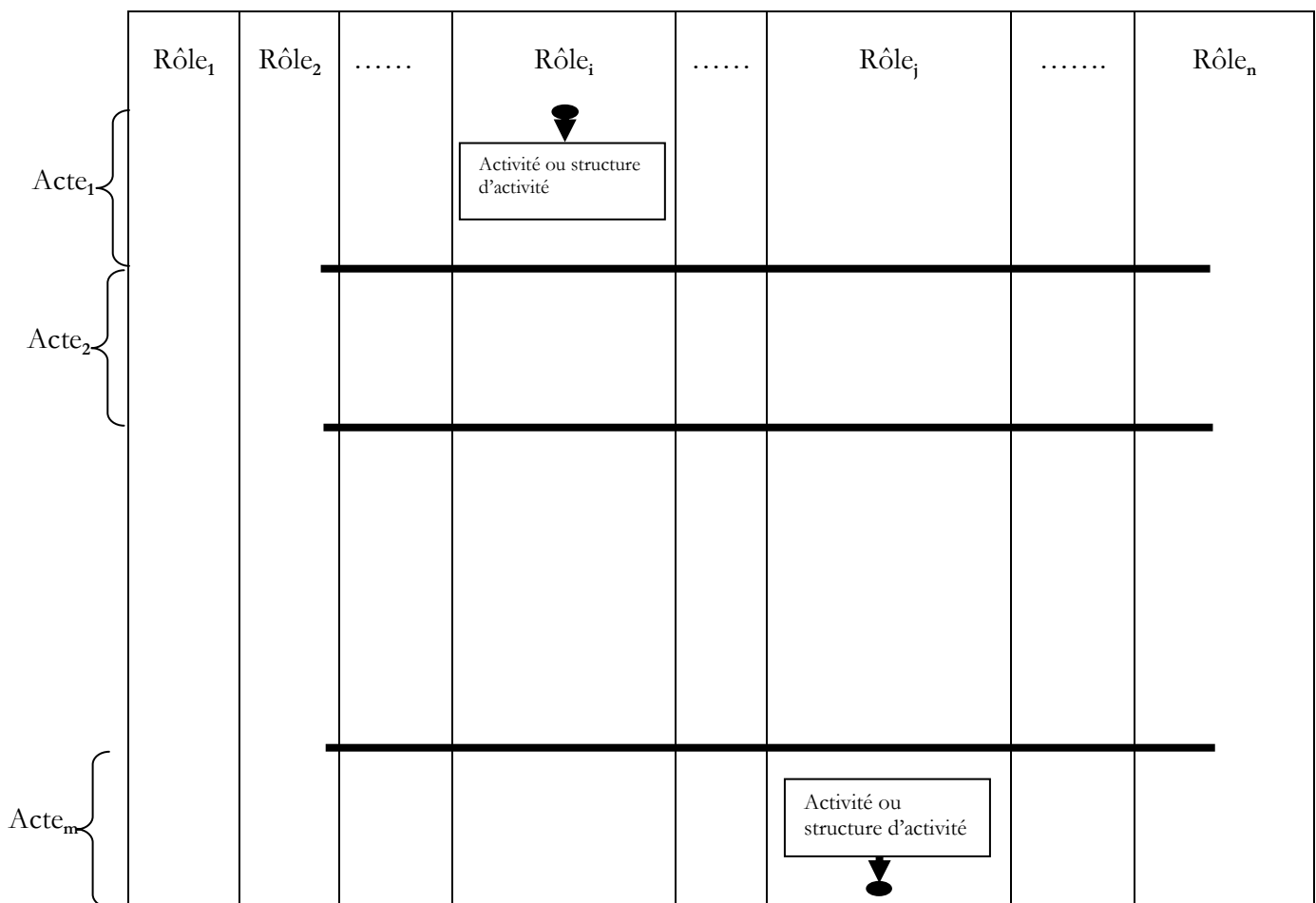


Figure II.15 : Diagramme d'activités UML



## 4.1.2.2. Document XML conforme IMS-LD

La structure d'un document XML décrivant une situation d'apprentissage est le suivant :

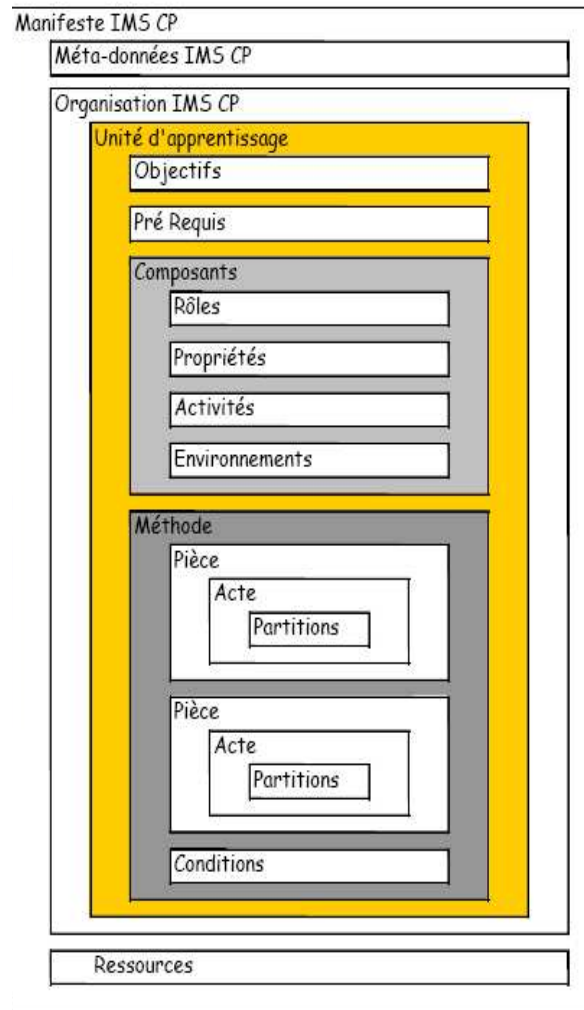


Figure II.16 : structure d'un document XML décrivant une situation d'apprentissage [IMS Learning Design]

#### 4.2. Rédaction du scénario d'apprentissage IMS LD

La rédaction du scénario d'apprentissage consiste à :

- Créer le Learning Design lui-même,
- Ajouter les différents fichiers nécessaires à l'achèvement des différentes activités.

Nous détaillerons dans les paragraphes suivants :

La création de chacun des éléments du Learning Design,

Les différents fichiers composant le Learning Design

L'utilisation d'un éditeur permettant de créer facilement le fichier manifeste.

### 4.2.1. Fichiers composants le Learning Design

Pour des raisons de portabilité, le Learning Design est incorporé dans un Content Package IMS (IMS CP) (cf. **Figure III.16**). Il s'agit d'un fichier ZIP compressé, comprenant un fichier *manifeste* (qui contient le Learning Design à proprement parler) et d'autres fichiers composant le contenu à afficher. La présence de ces derniers dans le fichier ZIP n'est toutefois pas obligatoire, dans la mesure où le contenu ne doit pas obligatoirement être présent localement : le contenu peut en effet être « appelé » par URL, et donc se trouver virtuellement n'importe où sur Internet. Le fichier ZIP ainsi constitué est appelé *Unit of Learning* (Unité d'Apprentissage). Les paragraphes suivants présentent en détail les constituants de l'unité d'apprentissage, à savoir le fichier manifeste ainsi que les différents types de fichiers pouvant le composer.

### 4.2.2. Le fichier manifeste : « imsmanifest.xml »

Tous les aspects du Learning Design sont contenus dans un fichier XML nommé « *imsmanifest.xml* ».

```

<?xml version="1.0" encoding="UTF-8"?>
<manifest>
  <organizations>
    <imsld:learning-design>
      <imsld:components>
        <imsld:roles>
          ... (roles déclarés ici)
        </imsld:roles>
        <imsld:properties>
          ... (properties déclarées ici)
        </imsld:properties>
        <imsld:activities>
          ... (activités décrites ici)
        </imsld:activities>
        <imsld:environments>
          ... (environnements décrits ici)
        </imsld:environments>
      </imsld:components>
      <imsld:method>
        <imsld:play> (1 ou plusieurs)
          <imsld:act> (normalement plusieurs)
            <imsld:role-part> (normalement plusieurs)
              ... (role-part declare ici)
            </imsld:role-part>
          </imsld:act>
        </imsld:play>
        <imsld:conditions> (éventuellement plusieurs, pour groupage logique)
          ... (toutes les conditions sont exprimées ici)
        </imsld:conditions>
      </imsld:method>
      <imsld:metadata>
        ... (données méta du Learning Design situées ici)
      </imsld:metadata>
    </imsld:learning-design>
  </organizations>
  <resources>
    ... (liste des ressources ici)
  </resources>
</manifest>

```

Figure II.17 : structure d'un fichier manifest.xml [IMS Learning Design]

On peut remarquer ici les balises `<manifest>`, `<organizations>` et `<resources>`, qui ne font pas partie du modèle Learning Design ; ces balises font en fait partie de la spécification IMS Content Packaging, sur laquelle repose le fichier ZIP du Learning Design.

L'élément *manifest* est simplement le conteneur global du Learning Design et l'élément *organizations* a pour seul but dans un Learning Design de contenir l'élément *learning-design*.

L'élément *resources* permet de regrouper à un seul endroit toutes les ressources (fichiers externes) utilisées par le Learning Design, et de spécifier leur *type* (*webcontent* ou *imsldcontent*, cf. paragraphes suivants).

Il est encore à noter que, alors qu'une *activity*/composé d'*environment*/etc. ne pointe que vers une seule *resource*, une *resource* peut contenir des références vers plusieurs fichiers.

Un tel ensemble de fichiers est alors considéré comme tous du même type, et l'utilisation de la *resource* implique l'utilisation de l'ensemble des fichiers qui y est associé.

#### 4.2.2.1. Ressources de type « webcontent »

Les ressources – locales ou distantes – de type « webcontent » peuvent pointer vers n'importe quel type de fichier informatique pouvant être visualisé dans un navigateur Internet.

Les types de fichiers qui nécessitent l'utilisation de plug-ins, tels RealMedia, Flash, Acrobat, etc., ainsi que les exécutables pouvant être lancés à partir d'un tel navigateur sont également acceptés.

#### 4.2.2.2. Ressources de type « imslcontent » : XHTML

Les ressources – locales ou distantes – de type « imslcontent » permettent, par le biais des *global-elements*, l'affichage et le réglage de *properties*.

Elles permettent aussi l'utilisation de l'attribut *class* pour influencer sur la visibilité d'éléments de leur contenu.

Ces documents utilisent des schémas basés sur le XML (p.ex. XHTML) et doivent, pour fonctionner correctement, utiliser le *namespace* d'IMS-LD.

Pour rendre visible ou invisible un contenu, l'attribut *class* s'utilise de la manière suivante :

On l'ajoute à un élément du fichier XHTML, par exemple :

```
<div class="nom_de_classe"> (texte et/ou autres balises imbriquées ici) </div>
```

On utilise alors le modèle *then* (dans le Learning Design présent sous forme XML dans le fichier *imsmanifest.xml*) de la manière suivante :

```
<then><show>
<class class="nom_de_classe" />
</show></then>
```

Ou

```
<then><hide>
<class class="nom_de_classe" />
</hide></then>
```

Pour faire apparaître ou disparaître à volonté l'élément `<div>` en question :

Pour régler ou afficher une *property*, on peut utiliser les balises <set-property> (respectivement <set-property-group>) et <view-property> (respectivement <viewproperty-group>) dans le document XHTML, de la manière suivante :

```
<set-property ref="ref_property" property-of="self" view="value" />
<view-property ref="ref_property" property-of="self" view="value" />
```

L'attribut *ref* contient la référence définissant la *property* à utiliser.

L'attribut *property-of* spécifie le propriétaire de la *property* en question (dans le cas de *personal properties*), et peut être égal à "self" (soi-même) ou à "supported-person" (personne supportée, dans le cas d'une *support-activity*, ou d'un *monitor-service*).

L'attribut *view* peut prendre les valeurs "value" ou "title-value", suivant que l'on désire afficher la valeur uniquement ou à la fois la valeur et le titre de la *property*.

#### 4.2.3. Utilisation d'un éditeur: Reload Learning Design Editor

Le logiciel Reload Learning Design editor (« Reload ») permet la génération simplifiée de Learning Design, c'est-à-dire du fichier manifeste.

Basé sur la technologie Java, il est disponible en tant qu'exécutable pour les systèmes d'exploitation Windows, MacOS et Linux, il est gratuit et open source.

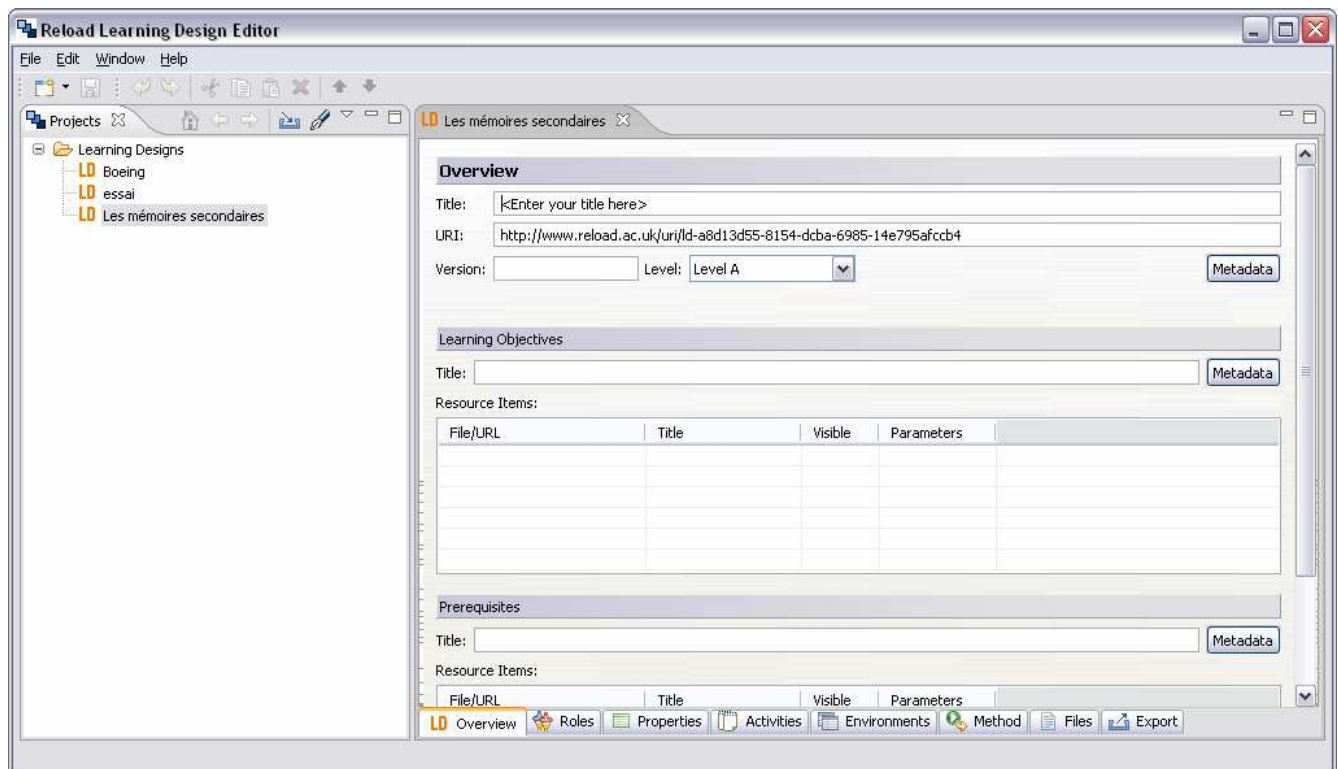


Figure II.18: Reload Learning Design Editor, version 2.1.0 [Reload LD Editor, ver 2.1.0]

Lors de la création d'un nouveau Learning Design (ou de l'importation d'un existant) avec Reload, le programme crée un dossier de travail où il stocke tout le contenu local, Une interface est alors affichée, fournissant des masques qui permettent la création facilitée de tous les composants de la *method*.

Pour ce faire, Reload segmente son interface en différentes parties, représentées par des onglets dans la Figure 17 ci-dessus :

**Overview:** contient les attributs principaux du Learning Design, tels que :

a. URI (Identifiant Universel de Ressource) ; A la forme d'un URL sans en être un, et permet uniquement de fournir un identifiant universellement unique pour le Learning Design, tel que par exemple : `http://www.unil.ch/groupe/nom_ressource`.

b. objectifs d'apprentissage et pré requis (sous la forme de ressources) ;

c. données méta relatives au Learning Design ; Aucun éditeur de données méta n'étant incorporé, il convient donc d'éditer celles-ci manuellement ; l'autre solution consiste à utiliser un éditeur tel que Reload Editor, disponible à [5], puis de les exporter/réimporter à travers un fichier XML).

d. niveau du Learning Design ; A spécifier impérativement pour l'interpréteur.

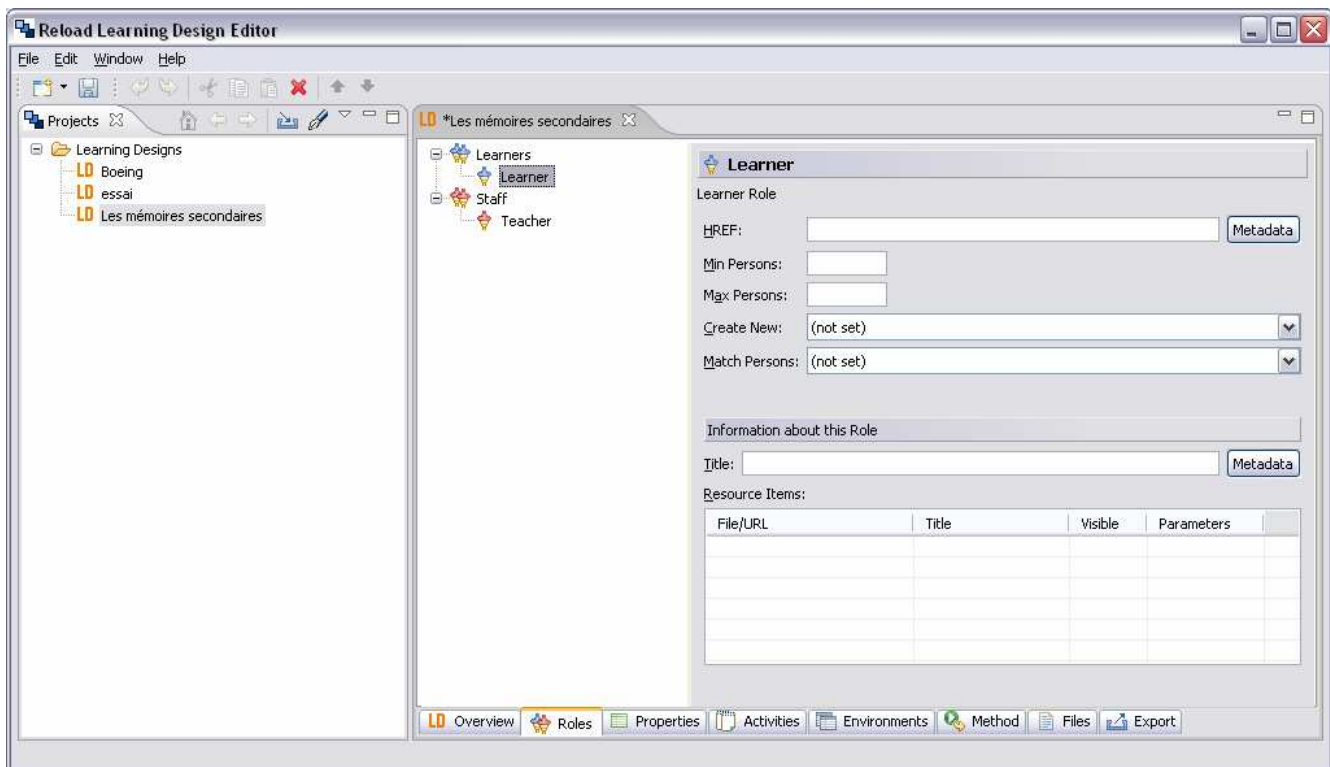


Figure II.19: onglet *Roles* [Reload LD Editor, ver 2.1.0]

**Roles:** permet la gestion des différents *roles*.

Peuvent être de deux types différents : *staff* et *learner* de façon à permettre un groupage logique.

Une *support-activity* peut en effet aussi bien pointer vers un (ou plusieurs) *role* que vers un (ou plusieurs) type de *role*, permettant (par exemple) d'effectuer une activité de support dirigée vers tous les apprenants quel que soit leur *role* exact.

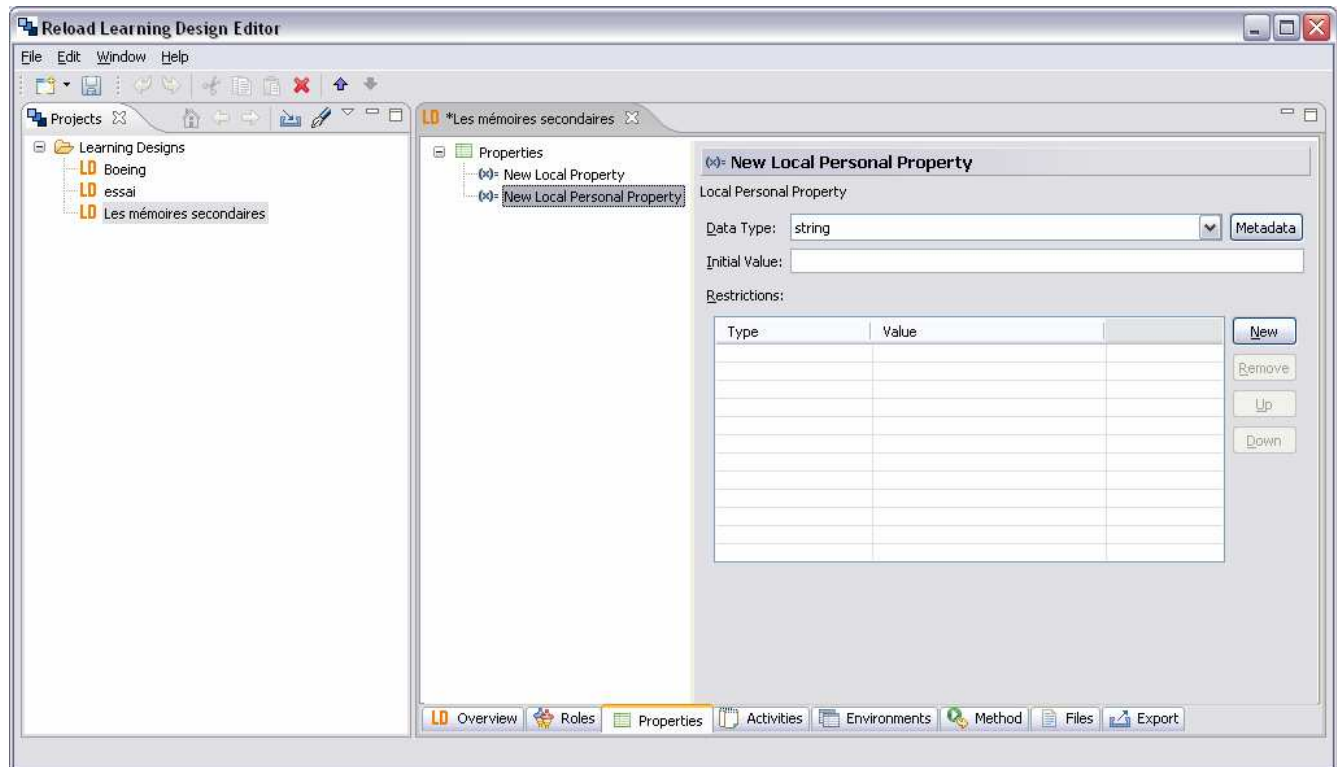


Figure II.20: onglet *Properties* [Reload LD Editor, ver 2.1.0]

**Properties:** permet la déclaration des propriétés et de leurs attributs.

Quatre types distincts de *properties* – ainsi que le type de données qui leur est associé – sont définis par le Learning Design.

Cet onglet permet de définir ces paramètres, ainsi que des attributs tels que la valeur initiale, la définition globale (pour les *global properties*), ainsi que les restrictions sur les valeurs que peut prendre la *property*.

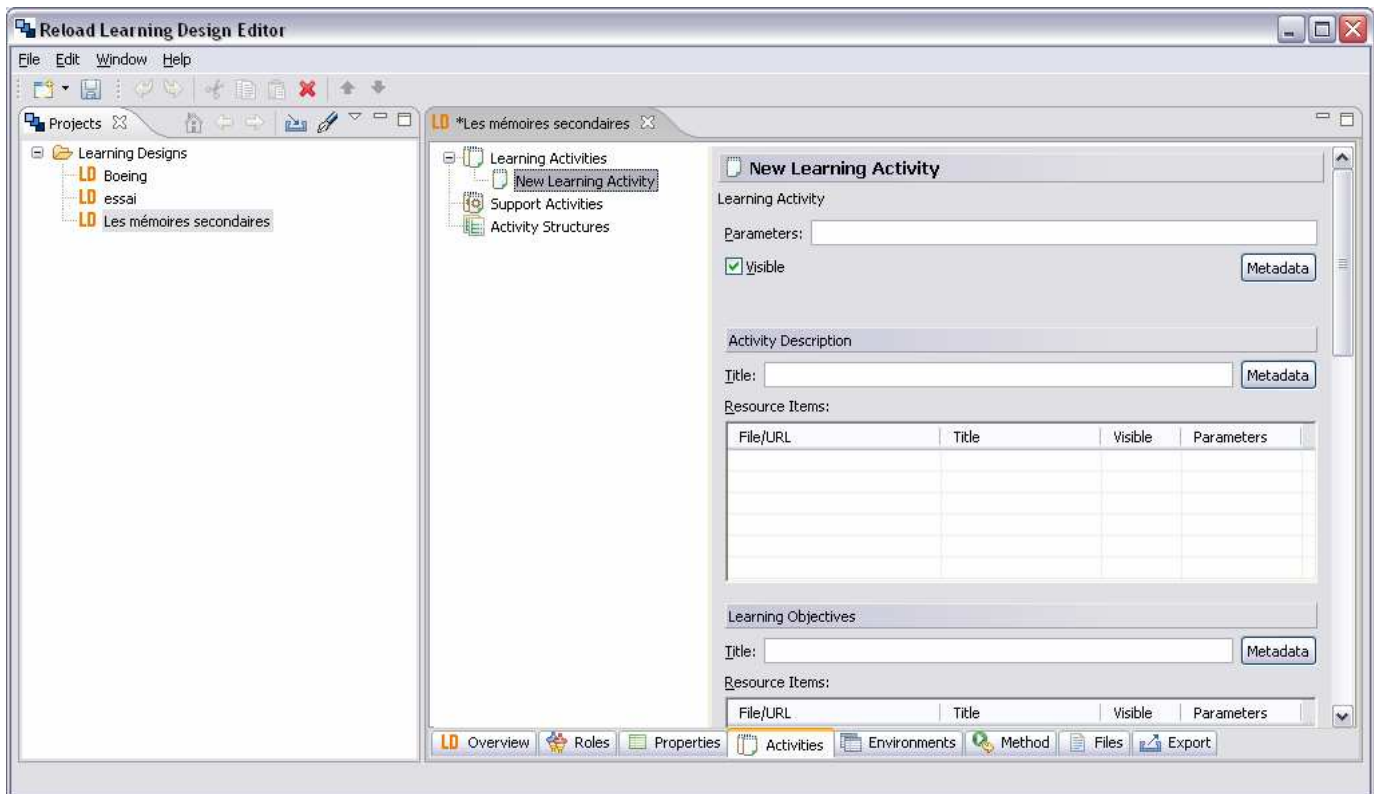


Figure II.21 onglet *Activities* [Reload LD Editor, ver 2.1.0]

**Activities** : permet la définition des activités.

Les activités de type *learning activity* contiennent plusieurs attributs principaux :

- La description (permettant d'indiquer au rôle concerné ce qui est attendu),
- Des pré requis/objectifs d'apprentissage,
- Un *feedback* à donner après achèvement de l'activité, et des *environnements* qui seront proposés (tels des outils) en corrélation avec l'activité à remplir.

L'attribut *visible* permet de définir la visibilité initiale d'une *activity* – visibilité pouvant être influencée par une *condition* dans les niveaux B et C.

Une *activity* peut en outre (par l'intermédiaire d'un clic droit sur l'*activity* dans Reload) faire pointer vers plusieurs objets d'apprentissage par l'intermédiaire de l'utilisation d'un ou de plusieurs *environnements*, eux-mêmes référant une ou plusieurs unités d'apprentissage et/ou *services* et/ou autres *environnements*.

On peut également dans ce panneau éditer intuitivement les modèles *complete-activity* et *on-completion*, qui permettent de déterminer quand l'activité est terminée, et l'action à entreprendre une fois qu'elle est terminée.

On peut par exemple déclencher une *notification* ou changer la valeur d'une *property* (automatiquement à la fin d'une activité).

Une *support-activity* comprend en outre l'attribut *roles*, qui permet de définir à qui est destiné le support.

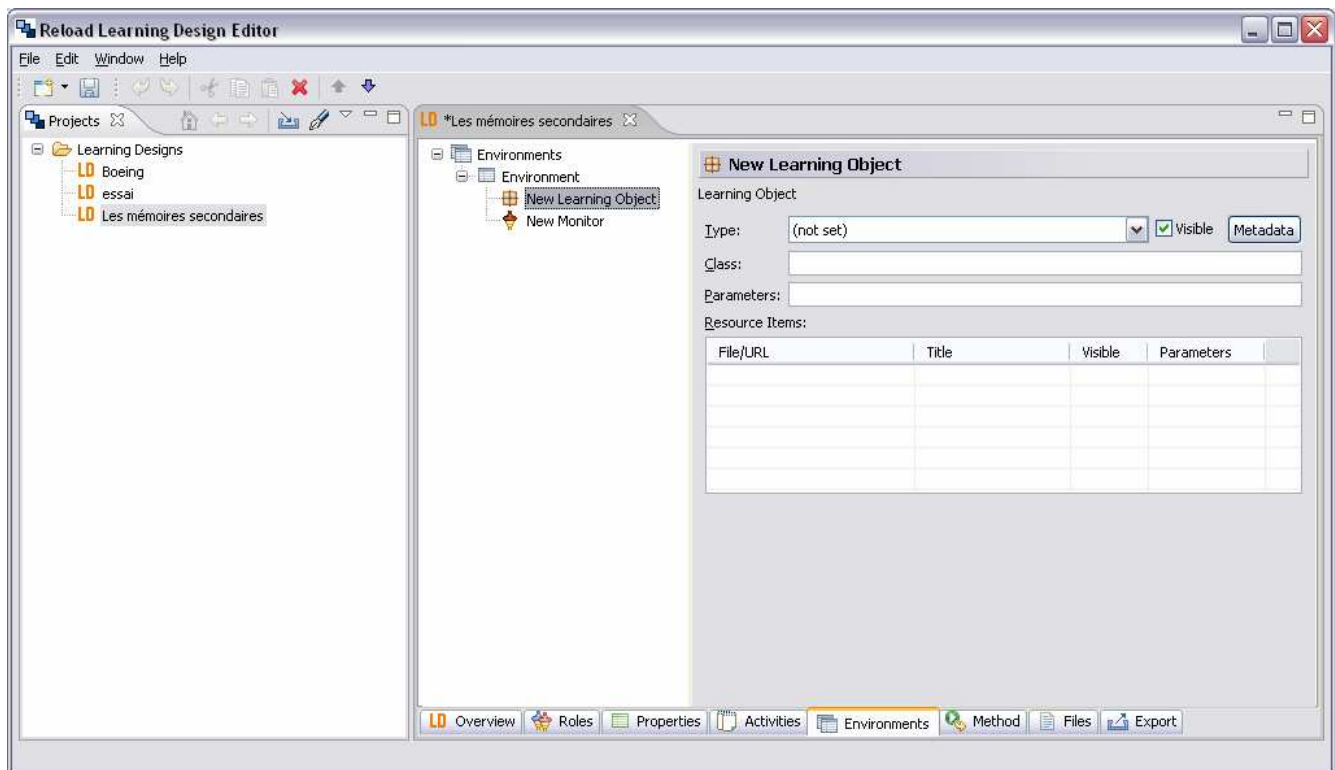


Figure II.22: onglet *Environments* [Reload LD Editor, ver 2.1.0]

***Environments*** : permet la construction des environnements.

Ces *environnements* (par l'intermédiaire d'un clic droit, dans Reload) peuvent faire référence à des objets d'apprentissage et/ou à des *services* et/ou à d'autres *environnements*.

Les *services* disponibles sont : *send-mail*, *conference*, *indexsearch*, plus *monitor* dans le niveau B.

Les objets d'apprentissage sont définis par une ressource (locale ou distante) pointant vers l'objet en question.

Les services en question (conférence, etc.) sont définis explicitement de manière à pouvoir être implémentés dans l'environnement d'exécution (*runtime environment*) d'une façon automatique, ou manuelle par un responsable au moment de l'instanciation.

De plus amples détails au sujet des attributs des objets et différents *services* sont disponibles dans l'aide en ligne de Reload.



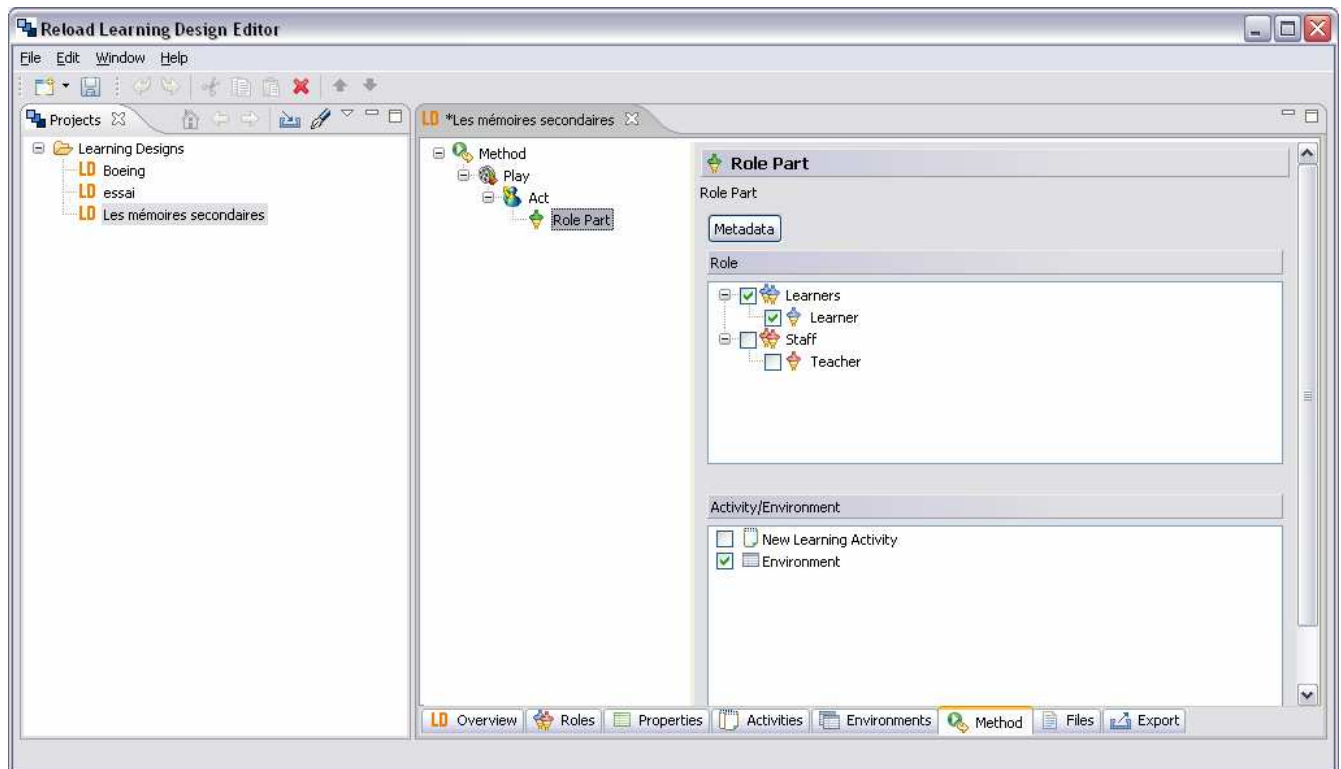


Figure II.23: onglet *Method* [Reload LD Editor, ver 2.1.0]

**Method** : permet la création du scénario lui-même.

La *method* est divisée en *plays* indépendantes, elles-mêmes divisées *acts* (ayant lieu successivement), divisés en *role-plays* – ici les éléments de niveau hiérarchique inférieur.

Le panneau de droite permet ici de définir les attributs de complétion de l'élément sélectionné, ainsi que le modèle *on-completion*, c'est-à-dire : comment considérer l'élément en question comme complété et que faire le cas échéant.

**Files** : liste des fichiers utilisés par le Learning Design.

Cet onglet permet d'ajouter des fichiers locaux au répertoire de travail de Reload (ce qui peut aussi être fait manuellement à l'extérieur de Reload), de façon à pouvoir ensuite les « appeler » simplement par leur nom de fichier, en utilisant éventuellement un nom de sous-répertoire.

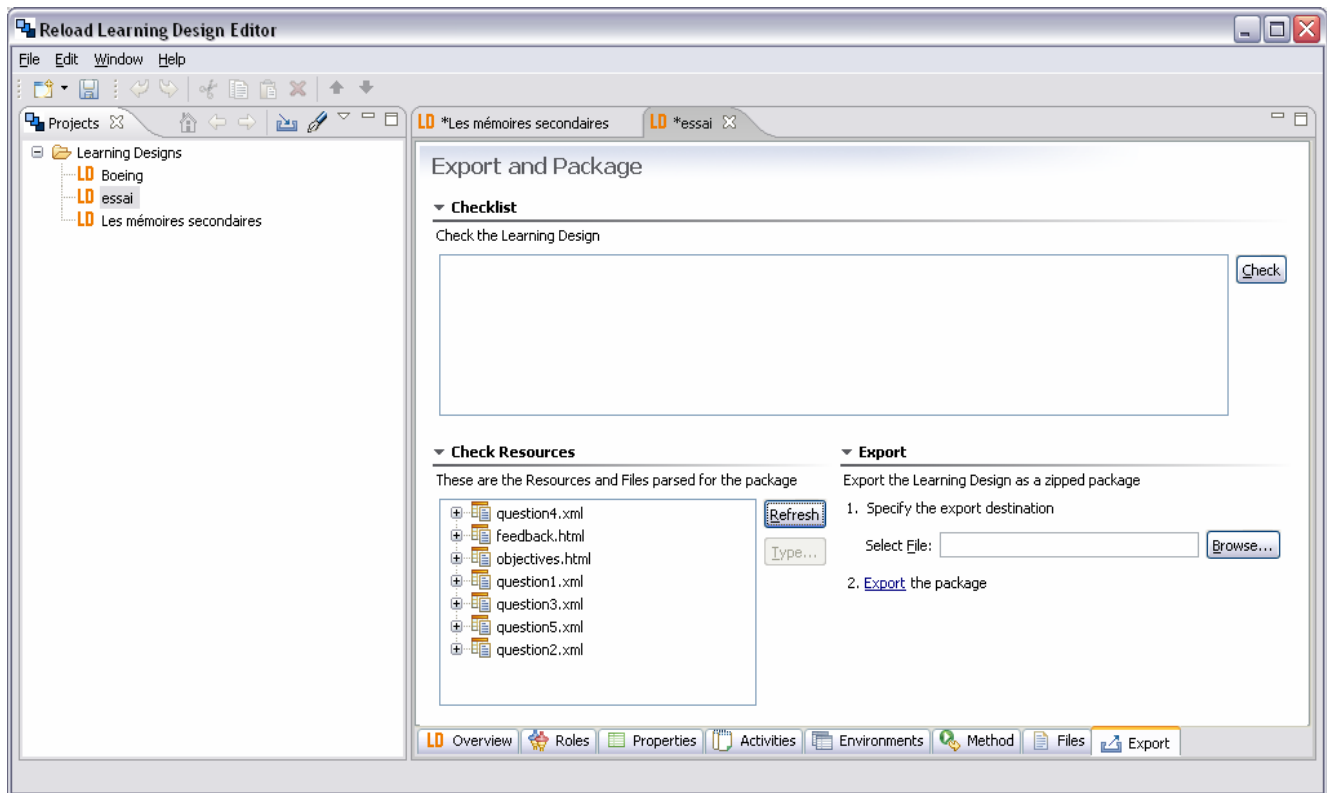


Figure II.24: onglet *Export* [Reload LD Editor, ver 2.1.0]

**Export** : création de l'unité d'apprentissage (*Unit of Learning*).

Cet onglet permet la validation du Learning Design par rapport au niveau spécifié dans l'onglet *Overview* grâce au bouton « Check ».

On peut ensuite définir le type de chacune des ressources utilisées : *webcontent* ou *imsldcontent*.

Il est conseillé de cliquer une ou deux fois sur le bouton « Refresh » avant, à cause d'un léger bug du programme n'affichant pas immédiatement et complètement toutes les ressources.

On peut ensuite et finalement procéder à la création de l'*Unit of Learning*, cette dernière incorporera le fichier manifeste et tous les fichiers du répertoire de travail de Reload dans un fichier ZIP compressé.

### 4.3. Implémentation :

L'implémentation de IMS LD passe par :

1. l'étape **conception**, on conçoit le modèle d'unité d'apprentissage ; C'est l'**environnement autour IMS LD**
2. l'étape **production**, ce sont la validation de conformité, la publication vers l'environnement cible et le lien à la population cible ; c'est l'**instanciation**

3. l'étape de **diffusion** ; ce sont la personnalisation, la planification des rôles et la liaison à d'autres unités d'apprentissage ; C'est l'**exécution** / l'**interprétation**

Ces trois étapes font appel à des **systèmes externes** :

Portails, Services, Constitution des populations cibles.....

## Chapitre III : Adaptation

Ce troisième chapitre introduit la notion d'adaptation, concept fondamental de la problématique qu'on s'est proposée d'étudier, nous mettons en évidence cette notion en décrivant son utilité, ses stratégies et ces types. Nous aborderons également l'aspect adaptation sous l'angle IMS-LD.

## 1. Introduction

Il est connu, et ce depuis longtemps, que les apprenants diffèrent dans leurs apprentissages.

Certains apprenants sont beaucoup plus à l'aise face à des images, et arrivent ainsi à mieux comprendre une situation donnée, alors que d'autres préfèrent des lignes de textes qui stimulent beaucoup mieux leurs compréhensions.

Certains apprenants aimeraient disposer d'instructions détaillées sur une nouvelle tâche avant de l'entamer, alors que d'autres préfèrent y « plonger » et essayer avec leurs propres moyens.

Certains apprenants sont ravis de donner des réponses à n'importe quelles questions, alors que d'autres sont beaucoup plus réticents et ne participent que s'ils sont sollicités.

A travers ces trois cas de figure, nous voyons bien que les apprenants diffèrent dans leurs manières d'appréhender les problèmes, ainsi pour leur éviter des échecs, il serait très bénéfique pour eux de recevoir des apprentissages personnalisés ou individualisés.

L'un des principaux défis de l'e-learning est de permettre l'adaptation aux intérêts personnels des apprenants, à leurs caractéristiques et à leurs objectifs. Les apprenants exigent des contenus et des activités basées sur leurs préférences et leurs connaissances antérieures.

Plusieurs recherches ont été faites dans ce domaine, il en est résulté des catégorisations des apprenants auxquelles sont associés des schémas pédagogiques différents. Chaque schéma décrit comment traiter les apprenants afin d'atteindre les meilleurs objectifs [Towle & Halm, 05].

## 2. C'est quoi l'adaptation

L'adaptation dans le contexte du e-learning vise à créer l'expérience de l'apprenant. Les apprenants doivent être capable d'acquérir de nouveaux schémas qui guideront leurs actions dans des nouvelles situations.

L'adaptation est une méthode pour générer une expérience d'apprentissage pour chaque personne ou rôle impliqué (par exemple, apprenant, tuteur) dans le processus d'apprentissage, durant une période spécifique, avec l'objectif d'augmenter le rendement ou l'efficacité, selon une série de critères prédéfinis [Van Rosmalen et al, 2006].

Des recherches récentes ont donné des définitions sur l'adaptativité (*adaptivity*), toutes centrées sur l'apprentissage personnalisé. Généralement, il s'établit une différence entre adaptativité et adaptabilité (*adaptability*) [Burgos et al., 2006c].

L'adaptativité est la capacité de modifier des cours eLearning en utilisant différents paramètres et un ensemble de règles prédéfinies.

Alors que l'adaptabilité est la possibilité qu'ont les utilisateurs (apprenants ou/et tuteurs) à personnaliser eux-mêmes les cours qu'ils suivent.

Ces deux approches vont d'un système centré machine (*adaptivity*) vers un système centrée utilisateur (*adaptability*).

On peut ainsi dire que l'adaptativité est une adaptation statique, les règles qui régissent l'adaptation sont faites pendant la phase de conception (*design-time*).

L'adaptabilité est une adaptation dynamique, l'adaptation est guidée par l'intervention de l'utilisateur qui se fait pendant le temps d'exécution (*run-time*).

Ceci n'implique pas nécessairement que l'utilisateur, plus précisément l'apprenant, devrait avoir un contrôle complet sur son propre apprentissage, car ceci voudrait dire que [Burgos et al., 2006c]:

- L'apprenant sait ce qui est le meilleur pour lui durant tout le processus d'apprentissage.
- L'apprenant est conscient, sait et contrôle toutes les contributions qui sont faites à son propre apprentissage.
- L'apprenant est capable de prendre la bonne décision quand toutes ces informations sont collectées.

Cependant, ces deux approches interviennent communément dans un processus d'apprentissage et sont en étroite relation, elles sont même combinées, l'une complète l'autre et vis versa. Il devient ainsi assez difficile d'isoler l'une de l'autre.

Les deux termes sont tous les deux utilisés pour l'apprentissage personnalisé. Daniel Burgos, Colin Tattersall et Rob Koper utilisent indifféremment les termes (*adaptivity*, *adaptability*, *adaptation* et apprentissage personnalisé) pour désigner un processus d'adaptation, nous choisirons le terme adaptation.

### 3. Rôles et entrées (inputs)

Bien que l'apprenant soit l'acteur principal et le leitmotiv de son propre processus d'apprentissage, il existe d'autres acteurs et facteurs à prendre en considération. Entre les acteurs ou les rôles que l'on considère, il en existe trois [Van Rosmalen et al., 2006]:

- l'apprenant,
- l'enseignant et
- le système en ligne créé par un auteur/expert en éducation, communément appelé l'ensemble de règles ou le moteur d'adaptation.

On considère un certain nombre de facteurs répartis entre ces trois rôles [Burgos, 2007] (cf. **Figure III.1**):

- 1) La charge cognitive de l'apprenant, c'est-à-dire sa capacité à prendre des décisions sur l'itinéraire de son processus d'apprentissage.
- 2) Le style d'apprentissage de l'apprenant.
- 3) Le comportement de l'apprenant, c'est-à-dire sa façon d'agir, son temps de réponse, toutes les actions qu'il considère avant de se décider pour un choix d'exécution.
- 4) La performance de l'apprenant, la capacité d'apprentissage qu'il présente au cours de l'exécution de l'unité d'apprentissage.
- 5) Les connaissances préalables de l'apprenant (prérequis).
- 6) Le comportement du tuteur, quel est son temps de réponse, quel rythme impose-t-il à la communication avec l'apprenant.
- 7) La décision du tuteur, en vue de l'ensemble des données collectées.
- 8) La contribution du pédagogue ou le concepteur de l'apprentissage, en décrivant le modèle pédagogique ou le méthodologie didactique.
- 9) La contribution du pédagogue ou le concepteur de l'apprentissage, en définissant l'ensemble des règles que régit le système en ligne.
- 10) L'état du groupe d'apprentissage auquel appartient l'apprenant (en se basant sur la décision, le comportement et les performances de tous ses membres).

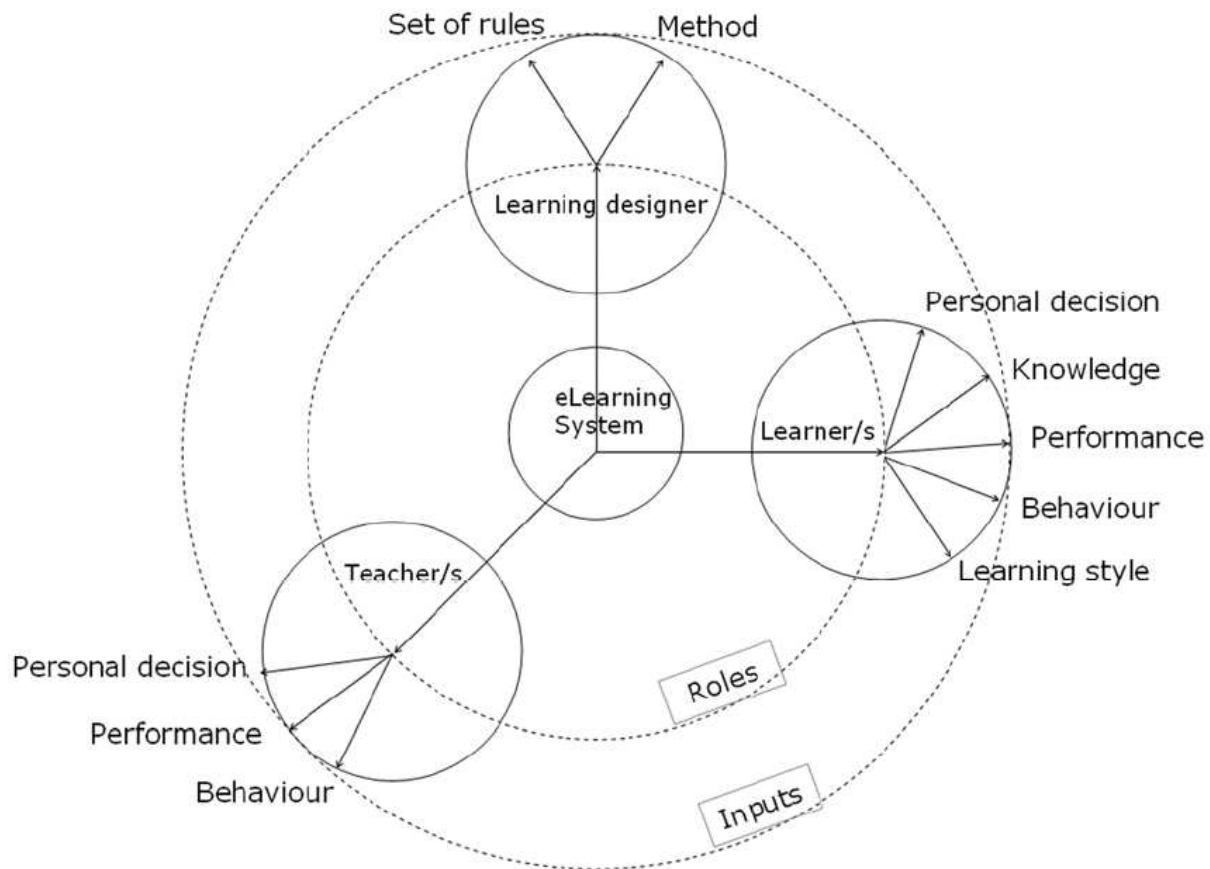


Figure III.1 : Inputs et rôles dans un système d'apprentissage adaptatif [Burgos, 2008]

Nous pouvons ainsi conclure que :

Un processus d'adaptation est vu comme le résultat d'un processus pouvant relever de l'adaptabilité, de l'adaptativité ou de processus qui combinent leurs caractéristiques.

Les acteurs intervenant dans le processus d'adaptation sont :

1. L'utilisateur, par lequel il faut comprendre l'apprenant ou le tuteur.
2. Le système, par lequel il faut comprendre l'ensemble de règles prédéfinies par l'auteur ou le concepteur pédagogique.

Les auteurs produisent des cours, utilisés par les tuteurs qui guident les apprenants à assimiler le contenu sémantique des cours.

Ces trois acteurs s'articulent autour du système d'apprentissage, et interviennent dans le processus d'adaptation.

#### 4. Niveaux d'adaptation

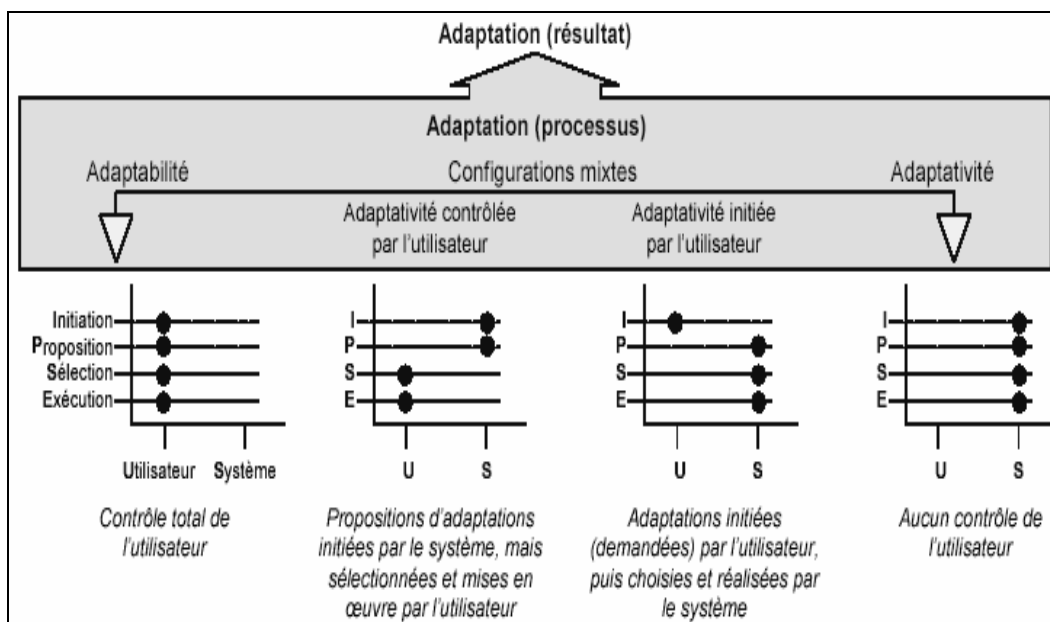
Quatre niveaux de contrôle d'adaptation différents sont introduits [Bekka, 2007] :

1. Initiation : décision d'un acteur de suggérer une adaptation.
2. Proposition : représente les suggestions ou les recommandations.
3. Sélection : Correspond au processus de choix parmi les propositions ou les recommandations.
4. Exécution : la mise en œuvre de la réaction choisie.

L'implication de l'utilisateur et/ou du système dans le processus d'adaptation est selon le degré de contrôle qu'exerce l'acteur (système ou utilisateur) sur les quatre niveaux.

Comme illustré dans la Figure ci-dessous (Cf **Figure III.2**) :

- L'utilisateur ou le système : représenté par un axe horizontal
- Le contrôle des actions (initiation, proposition, sélection, exécution) : représenté par axe vertical.
- Le contrôle : symbolisé par un cercle plein.



**Figure III.2** : Niveaux d'adaptation [Bekka, 2007]

#### 5. Principales approches de conception d'UA adaptatives (Etat de l'art)

Nous examinerons la stratégie de conception de l'apprentissage adaptatif dans le Learning Design, qu'on peut scinder en deux catégories :

- La première repose sur deux axes séparés, le premier axe prend en compte la logique d'implémentation alors que le second axe concerne ce qui est adapté exactement [Towle & Halm, 05].
- Quant à la deuxième catégorie elle s'intéresse au moment où l'adaptation est effectuée ; est-ce pendant la conception (*design-time*) ou est-ce pendant le déroulement de l'unité d'apprentissage (*runtime*) [Zarraonandia & al, 06].



1<sup>ière</sup> catégorie :

Il y a au moins deux axes séparés sur lesquels la stratégie d'implémentation de l'apprentissage adaptatif dans le Learning Design, peut être décrite [Koper and Tattersall, 2005].

La première repose sur la logique qui implémente la stratégie adaptative :

- Est-elle encapsulée à l'intérieur de l'unité d'apprentissage ?
- Est-elle externe à l'unité d'apprentissage ?

La seconde concerne ce qui est adapté exactement :

- Est-ce le contenu de l'UA ?
- Est-ce l'interface de l'UA ?
- Ou bien est-ce carrément autre chose ?

En ce qui concerne la première stratégie, celle où l'implémentation de la logique adaptative se fait complètement à l'intérieur de l'UA, il y a au moins quatre (04) champs où l'UA peut être adaptée à des apprenants individuels sur la base de leurs caractéristiques d'apprentissage [Towle & Halm, 05] :

- Changer l'environnement pour des apprenants différents – en fournissant des ressources différentes, ou les mêmes ressources dans un ordre différent.
- Changer la méthode pour des apprenants différents.
- Mettre les apprenants différents dans des rôles différents, ou bien fournir des supports à partir de rôles différents pour des apprenants différents.
- Changer les activités données aux apprenants différents.

Les éléments IMS-LD impliqués dans l'adaptation selon la stratégie adoptée par Halm et Towle sont :

- a. *environment*,
- b. *method*,
- c. *roles* et
- d. *activities*

Quant à Van Rosmalen et Boticario, ils examinent une adaptation externe à l'UA, en apportant des modifications à la fois aux éléments internes de l'UA et à la couche d'exécution à travers laquelle l'UA est livrée (player).

Cette logique d'adaptation relève du projet ALFANET (*Active Learning For Adaptive interNET*), qui vise à développer un environnement d'apprentissage qui intègre de nouveaux principes et outils dans le domaine de la conception d'apprentissage et de l'intelligence artificielle.

La figure suivante présente le cycle de vie complet de l'adaptation dans ALFANET.

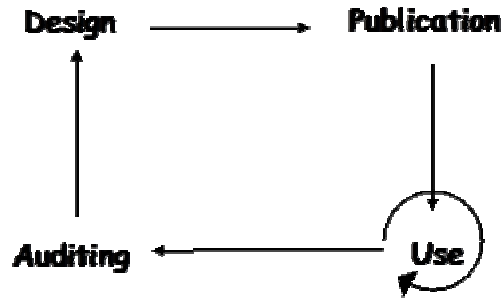


Figure III.3 : Les quatre phases du cycle de vie de l'adaptation dans ALFANET [Van Rosmalen et al., 2006]

**Design** : Le design contient la logique des adaptations préconçues et fournit les supports et l'information sur lesquels se base le raisonnement de l'adaptation pendant le runtime. Afin de concevoir un cours, l'auteur peut sélectionner un ou plusieurs modèles pédagogiques et les appliquer sur le cours en question. Ces modèles sont une traduction des résultats de recherche dans l'apprentissage et le design pédagogique. Leur objectif est de simplifier les tâches complexes qu'a l'auteur lors de la conception de son cours.

**Publication** : Dans cette phase, les apprenants (ou tuteurs) se voient attribuer leurs rôles, leurs groupes et leurs profils personnels ainsi que les droits qu'ils ont dans le cours.

**Use** : Cette phase est axée sur l'environnement et les outils disponibles aux apprenant et aux tuteurs dans l'accomplissement de leurs activités au moment du déroulement du cours. Cet environnement est construit sur une architecture qui intègre un certain nombre de composants qui sont :

- Couche présentation (*presentation layer*) : Elle offre la possibilité d'adapter l'interface aux besoins de l'utilisateur (présentation personnalisée basée sur la présentation de modèles et des présentations adaptatives basées un interpréteur de règles des attributs IMS LIP et d'autres metadata).
- Moteur LD (*LD-engine*) : Il lit et exécute le design pédagogique tel que donné par l'auteur dans le LD, en incluant la maintenance et la prise en main des LD-properties.
- Agents d'adaptation (*Adaptation agents*) : Ils surveillent les interactions et les résultats des interactions des apprenants leur permettant de suggérer des recommandations aux utilisateurs.
- Serveur QTI (*QTI server*) : Il génère dynamiquement les questionnaires qui s'adaptent à l'évolution des apprenant dans leur cours et aux intérêts des apprenants. Il inter opère avec les données provenant d'autres standards (IMS-LD, IMS-LOM et IMS-LIP).

**Audit** (*Auditing*) : Il clôt le cycle de vie de l'adaptation. Il collecte les données dépendant des besoins de l'auteur dans l'utilisation effective du cours et les lui présente d'une manière claire. L'auteur fait des rapports montrant combien le design de ce cours a eu comme succès. Par conséquent, en fonction du résultat l'audit peut décider s'il y a lieu de reconsidérer (mise à jour) le design.

#### 2<sup>ème</sup> catégorie:

- Un processus d'adaptation appelé : « *late modelling of learning process* », une modélisation tardive du processus d'apprentissage est décrit ainsi [Zarraonandia & al, 06] :

Au moment de la conception d'un processus pédagogique, il n'est toujours pas possible de connaître toutes les caractéristiques des apprenants, donc de prévoir toutes les règles d'adaptation.

L'application d'un processus d'apprentissage est dans la pratique assez flexible puisqu'il n'est pas possible de prévoir toutes les réactions potentielles de la part des apprenants.

Les instructeurs prennent le processus d'apprentissage comme une base, et par la suite en observant les réactions des apprenants, ils peuvent intervenir en leur donnant des exemples additionnels, des explications

pour renforcer des concepts particuliers, ils peuvent également leur faire répéter des activités ou bien modifier les temps d'achèvement des évaluations, etc.

L'expérience gagnée des exécutions précédentes est comprise dans la définition du processus et un ensemble vaste de réponses des réactions d'apprenants est capturé. Ceci veut dire que la définition du modèle de cours n'est pas terminée pendant la phase de conception mais plutôt quand il n'y a plus de modifications requises à être appliquées.

- On s'intéresse aux possibilités d'adaptation qui sont offertes à la phase d'instanciation de l'unité d'apprentissage [Smahi, 2008].

L'idée initiale est inspirée du travail préparatif pour les scènes théâtrales, où le scénario, initial, de la pièce doit être joué plusieurs fois avant sa présentation finale, et pendant ces répétitions plusieurs modifications et améliorations peuvent être introduites. Par le même principe, sur un scénario d'apprentissage, il faut à plusieurs modifications, pour des fins d'adaptation, avant le déroulement final de l'unité d'apprentissage. D'où, à cette phase précise, il sera effectué aux modifications à l'intérieur de l'unité d'apprentissage et non pas à l'extérieur.

On peut résumer l'état de l'art sur les des stratégies de conception d'UA adaptatives et les différents travaux s'y référant dans le tableau suivant (cf. **Tableau IV.1**) :

Stratégies de conception d'une UA. Adaptative		Logique d'implémentation de l'adaptation		Qu'est ce qui est adapté ? Types d'adaptation			Moment de l'adaptation		
		Encapsulée dans l'UA	Externe à l'UA	Contenu	Interface	Autre chose (à préciser)	Pendant la conception (design-time)	Pendant l'instanciation	Pendant le déroulement (runtime)
Travaux réalisés par :	Towle et Halm	X <sup>4</sup>		X		X	X		
	Van Rosmalen et Boticario	X	X	X	X		X		X
	Smahi	X				X		X	
	Zarraonandia	X	X			X			X

Tableau IV.1 : Tableau résumant les stratégies de conception d'une UA.

## 6. Types d'adaptation

Traditionnellement, il existe trois types d'adaptations:

- a. Le premier est basé sur l'interface (*Interface-based*)
- b. Le second est basé sur l'itinéraire ou le parcours d'apprentissage ou encore le flux d'apprentissage (*Learning flow-based*)
- c. Le troisième est basé sur le contenu (*Content-based*).

<sup>4</sup> : Le X veut dire que les travaux réalisés par le chercheur en question se situent dans la stratégie de conception adaptative correspondante à la (aux) colonne(s) où se trouve(ent) le(s) X.

Cependant, dans la littérature, huit différents types d'adaptation mis en œuvre dans les systèmes de e-learning sont identifiés [Burgos et al, 2006b].

Tous utilisent des entrées différentes fournies au cours du processus d'apprentissage et visent à mettre au point les activités et les actions de l'apprenant pour obtenir la meilleure expérience d'apprentissage possible.

1) Interface-based: (Adaptation basée sur l'interface, appelée aussi la navigation personnalisée et est associée à l'accessibilité ou encore à la possibilité d'utilisation et à l'adaptabilité):

Les éléments et les options de l'interface, sont positionnés sur l'écran et leurs propriétés sont définies (couleur, taille, ombrage etc.).

Ce type d'adaptation est étroitement lié à la personnalisation générale et à la prise en charge d'individus présentant des handicaps ou ayant par exemple des difficultés concernant une insuffisance de vue ou de l'ouïe.

2) Learning flow-based: (Adaptation basée sur le flux d'apprentissage)

Les contenus du cours sont séquencés de différentes manières afin d'adapter le processus d'apprentissage aux caractéristiques de l'apprenant.

Le chemin d'apprentissage possède plusieurs itinéraires et est personnalisé pour chaque apprenant, à chaque fois que le cours est lancé, de telle sorte que l'apprenant puisse prendre des chemins différents qui dépendent de sa performance.

3) Content-based: (Adaptation basée sur le contenu)

Les ressources et les activités changent leurs contenus effectifs.

Par exemple, l'information à l'intérieur de l'activité d'apprentissage peut être classée selon trois niveaux de profondeur, chaque niveau est visualisé sur la base d'un certain nombre de facteurs. Le premier block des types d'adaptation sera la base des suivants.

Il existe également d'autres genres d'adaptation qui sont [Brusilovsky and Paylo, 2003]:

4) Interactive problem solving support: (Encadrement interactif de résolution des problèmes):

Lors de la résolution de problèmes, l'utilisateur est guidé dans le prochain pas à prendre afin d'arriver à la bonne solution du problème.

Ceci est réalisé grâce à un tuteur en ligne ou hors ligne, ou d'un ensemble de règles prédéfinies.

5) Adaptive information filtering: (Filtrage d'informations adaptatif)

Prise en charge de la récupération d'informations appropriées ne fournissant que des résultats pertinents et catégorisés à l'utilisateur.

Bien que ce type fournit des informations adaptées, il pourrait être considéré comme une facilité externe liée à une activité d'apprentissage et non comme une partie réelle de cette activité d'apprentissage elle-même.

6) Adaptive user grouping: (Groupement d'utilisateurs appropriés)

Permettre la création ad hoc de groupes d'utilisateurs et d'encadrement collaboratif exécutant des tâches spécifiques.

Par exemple, à la suite d'une série de questions, deux groupes d'utilisateurs débutants et avancés sont construits.

7) Adaptive evaluation: (Evaluation adaptative)

Le modèle d'évaluation, le contenu effectif et l'exécution du test peuvent changer en fonction de la performance de l'étudiant et de l'orientation du tuteur [Van Rosmalen, Vogten et al., 2006]

8) Changes on the fly : (Modifications à la volée)

La possibilité de modifier ou d'adapter un cours on-the-fly (à la volée) par le tuteur ou par l'auteur du cours pendant le temps d'exécution [Van Rosmalen and Boticario, 2005], en déplaçant en avant les types prévues qui ont été configurés et définis dans la phase de conception.

## 7. Adaptation sous l'angle IMS-LD

IMS-LD fournit un langage de modélisation capable de concevoir des unités d'apprentissage (UA) exécutables.

Daniel Burgos, Colin Tattersall et Rob Koper (2006) examinent comment IMS-LD peut être utilisé pour représenter chacune des huit types d'adaptations précédentes [Burgos et al., 2006b].

### 7.1. Adaptation de l'interface

L'adaptation de l'interface est basée sur des options de menu, des facilités de navigation et des facilités de visualisation. Cette question se rapporte à l'interface de l'utilisateur fournie avec les players IMS-LD tels que le player inclus avec *Reload player* [Reload, 2004] *CopperCore* et *Sled* [Burgos et al., 2006b].

Le player : est l'enveloppe qui contient l'UA, il permet son interprétation et sa visualisation.

La génération actuelle de ces outils ne fournit pas des facilités permettant l'adaptation de l'interface pendant l'exécution. Quoique *Sled* peut être personnalisé pendant la configuration en utilisant des feuilles de style.

Les players IMS-LD actuels ne peuvent pas changer la taille et la position de leurs cadres ou espaces de travail, la définition de leurs fenêtres ou n'importe quelle autre facilité de navigation.

Ces players ne peuvent pas changer des caractéristiques fondamentales, comme la taille de la fonte, le type de la fonte ou l'alignement.

Bien que l'adaptation de l'interface ne puisse pas être faite avec les players courants, certaines adaptations sont possible à l'intérieur de l'UA, en utilisant deux ressources : les couches DIV et les environnements.

Les couches DIV :

Il est possible de travailler avec les couches DIV qui peuvent être cachées ou visualisées pendant le temps d'exécution par n'importe quel acteur (participant) dans le processus d'apprentissage (l'apprenant, le tuteur ou l'ensemble des règles). A l'intérieur d'une couche DIV, des options différentes peuvent être définies..

Les environnements :

De même, il est possible d'utiliser plusieurs environnements pour fournir des configurations différentes (contenus, approches, vues) se rapportant à la même UA, chacune d'elles conduisant à une interface personnalisée.

Bien qu'aucune de ces deux solutions (les couches DIV et les environnements) ne soit basée sur un player, une autre vue de l'adaptation de l'interface peut être fournie ainsi.

### 7.2. Adaptation basée sur le flux d'apprentissage :

La description d'un flux d'apprentissage adaptatif est principalement basée sur quatre parmi les cinq différents éléments disponibles d'IMS-LD au niveau B [Koper and Burgos, 2005]:

- a. *properties*,
- b. *calculations*,
- c. *global elements* et
- d. *conditions*.

En plus, les services de surveillance (*monitoring services*) peuvent être rajoutés pour suivre le comportement des apprenants et permettre au tuteur d'adapter le flux dynamiquement.

Un exemple de ces caractéristiques est celui d'un apprenant qui peut étudier un cours en choisissant entre un itinéraire thématique et un itinéraire historique.

Un autre exemple est celui où les activités sont définies par la performance d'un étudiant après qu'il ait répondu à une formulaire d'évaluation. Dépendant du score final et en relation avec le niveau acquis, une ou une autre activité est visualisée.

Un dernier exemple où l'apprenant sélectionne directement quelle est la meilleure méthode à étudier parmi plusieurs propositions différentes.

### 7.3. Adaptation basée sur le contenu

Le contenu d'une activité nécessite une ressource liée à l'élément *Activity description*.

Bien que ce lien ne puisse pas être changé pendant le temps d'exécution (*runtime environment*), trois autres éléments peuvent être modifiés dynamiquement :

- a. Le contenu à l'intérieur d'une ressource XHTML, en définissant des classes et des couches DIV qui peuvent être cachées et visualisées sur la base de certains paramètres.
- b. Le contenu des propriétés/variables prédéfinies, qui peuvent être remplacées avec un autre contenu saisi dynamiquement.
- c. Le contenu d'une activité peut être adapté en permutant entre la visualisation ou l'inhibition d'un des plusieurs environnements liés.

Un exemple d'utilisation des environnements est celui où un apprenant décide du niveau de granularité qu'il veut et, où encore un apprenant prend le contrôle et permute entre l'activation ou la désactivation du support audio de l'unité d'apprentissage.

Une autre manière de l'adaptation basée sur le contenu est la modification des contenus liés aux ressources fixées et basée sur des outils externes.

Par exemple, une ressource liée à un service wiki hébergé en-dehors de l'UA pourrait adapter son contenu dynamiquement en se basant sur les contributions des utilisateurs, des tuteurs et des auteurs.

### 7.4. Encadrement interactif de la résolution de problèmes

Ce type d'adaptation peut être considéré comme une extension du *learning flow based*, avec la définition appropriée des propriétés et des conditions modélisant l'itinéraire, et l'incorporation d'un service de surveillance permettant le suivi du processus d'apprentissage de l'apprenant, en faisant des remarques ad hoc et en changeant le processus comme il convient. Ces changements peuvent être exécutés :

- a. Par la modification des arguments spécifiques par le tuteur.
- b. Par l'exécution de règles prédéfinies spécifiques pendant la conception.
- c. Par la combinaison de tous ces mécanismes.

Un exemple est où le tuteur modère les contributions d'un groupe d'apprenants dans une question ouverte, donnant accès à la prochaine étape quand le tuteur pense que l'étape courante est terminée. Le tuteur a le droit de fermer et de bloquer toutes les étapes et fournit un feedback contextuel.

### 7.5. Filtrage d'informations adaptatif

Quelques facilités rudimentaires sont disponibles au travers *index-search service*.

Plus concrètement IMS-LD peut signaler à un service de recherche externe de fournir le conteneur de l'exécution de cette application et aussi la visualisation des résultats.

### 7.6. Groupement d'utilisateurs adaptatif

L'administrateur (*User management*) a deux approches :

- a. Une, basée sur la création du rôle, où les utilisateurs y sont assignés,
- b. L'autre basée sur la création de l'utilisateur lui-même.

En utilisant le système d'administration fourni par plusieurs outils et moteurs (players), à savoir *Coppercore*, *CopperAuthor*, *Reload*, une fois l'UA publiée, l'administrateur (ou l'enseignant lui-même) peut rajouter des utilisateurs et leur assigner une exécution spécifique de cette UA.

Cependant, la création dynamique des rôles après le processus de publication est actuellement impossible.

Une fois la définition des rôles disponible et une exécution d'un UA définie, des utilisateurs spécifiques peuvent y être rajoutés, ou y être enlevés, n'importe quels de ces groupes ou de ces utilisateurs peut faire partie de l'exécution.

Quelques facilités de représentation sont disponibles dans IMS-LD pour supporter la création des groupes (*min-persons* et *max-persons*) et bien que l'assignation des utilisateurs aux groupes puisse être atteinte, une création des groupes on-the-fly complètement automatique nécessite des dispositifs de représentation additionnels.

### 7.7. Evaluation adaptative

En prenant la performance d'un apprenant dans une unité d'apprentissage comme une entrée, un ensemble complet de paramètres peut être stocké dans des propriétés locales pour être utilisé dans l'adaptation des évaluations formatives ou sommatives.

Certaines actions ou réponses d'un utilisateur peuvent être affectées dans des variables prédéfinies dans le *design-time* et peuvent être ainsi interprétées dans le *run-time* en suivant un ensemble de règles.

De cette manière, à la fois le système d'évaluation et le contenu lui-même et même l'interprétation des résultats, peuvent changer pour chaque utilisateur.

Un exemple où un questionnaire peut être complètement configuré avec des questions, des réponses, des seuils et des feedback définies dans le *run-time*.

Encore, l'obstacle principal à surmonter est la modification pendant le *run-time* du squelette (l'ossature) lui-même; ce qui n'est pas possible à ce jour avec l'état actuel de l'outillage.

Toutefois, il est possible de définir un ensemble de questions qui peuvent être cachées et montrées sur demande, fournissant une simulation hiérarchique de l'extensibilité de l'adaptation.

### 7.8. Faire des changements à la volée

Toutes les UA ont trois phases différentes dans leurs cycles de vie [Koper and Tattersall, 2005]:

- a. Design-time, (phase de conception)
- b. Publishing-time (phase d'instanciation ou de publication) et
- c. Run-time (Phase d'exécution).

Avec les outils actuels, une fois l'UA publiée, il n'est plus possible de changer sa structure, sa méthode ou la définition des paramètres de base (telles que les conditions ou les propriétés par exemple).

Bien sûr, si une UA est conçue, le tuteur est capable de modifier la manière dont l'apprenant perçoit le cours et le flux d'apprentissage :

- a. Les tuteurs peuvent mettre à jour le contenu, sur la base de contenus prédéfinis ou sur de nouvelles contributions.
- b. Le tuteur peut aussi influencer l'itinéraire d'apprentissage, en téléchargeant des fichiers, en montrant et en cachant les éléments de contenu et les éléments de structure, etc.

Cela signifie que le tuteur est capable de changer des choses pendant l'exécution, tant que cette possibilité a été antérieurement définie pendant le design-time.



## **Chapitre IV : Vers un design pédagogique adaptatif, extension de la spécification IMS-LD**

Ce chapitre présente une première analyse et une analyse plus détaillée de la solution à la problématique, il aborde les caractéristiques pertinentes qui définissent le profil de l'apprenant, et les modèles IMS-PAPI et IMS-LIP de l'apprenant.

Il présente les changements qui peuvent être apportés au Learning design afin que ce dernier puisse s'adapter au profil de l'apprenant ainsi que les éléments de la spécification IMS LD touchés par ces changements.

Finalement, on présente le modèle étendu de la spécification IMS LD.

## 1. Introduction

On s'intéresse dans notre sujet plus particulièrement aux cours dont le public apprenant cible n'est pas connu à priori avec précision.

Par conséquent les éléments pédagogiques contenus dans le LD, susceptibles de changer selon le public cible apprenant doivent être ainsi suffisamment adaptés en vue de prendre en compte une variété de choix et de préférences.

Notre travail consiste alors à étendre le modèle de spécification IMS-LD [IMSLD 03] en vue de le rendre capable de prendre en charge un design pédagogique adaptable à un public hétérogène.

## 2. Analyse

### 2.1. Première analyse

- On doit tenir compte des différences individuelles des apprenants : styles cognitifs, connaissances antérieures, âge, situation de travail, motivations et objectifs vis-à-vis de la formation qu'ils vont suivre etc.
- On utilisera le terme « **profil** » de l'apprenant pour englober toutes ces différences individuelles.

Le Learning Design (LD) est un formalisme, il permet de décrire un design pédagogique.

A première vue les éléments pédagogiques contenus dans le LD permettant de prendre en compte les caractéristiques personnelles des apprenants sont :

1. L'apprenant (**learner**) (cf. **Figure II.4**) : Placer les apprenants dans des rôles différents selon leurs profils respectifs.
2. L'activité des apprenants (**learning-activity**) (cf. **Figure II.8**) : changer les activités données aux apprenants n'ayant pas le même profil.
3. Le scénario (**play**) (cf. **Figure II.6**) : Proposer des cheminements alternatifs ou variables qui tiennent compte des profils des apprenants.
4. L'environnement (**environments**) (cf. **Figure II.9**) : Fournir à l'apprenant selon son profil des ressources différentes, ou les mêmes ressources dans un ordre différent.

Nous remarquons que les quatre points qu'on vient de citer tournent tous autour du terme **profil**, qui semble être l'élément pivot et central, en le sens où tous les éléments pédagogiques censés être adaptés ou changés s'y réfèrent expressément.

### Suggestions :

1. Faut-il alors rajouter l'élément « profil », dont la structure serait composée d'éléments qui vont recueillir toutes les différences possibles pouvant exister entre des apprenants, et de par là concevoir une structure de profil type d'apprenant?
2. Si c'est le cas, à quel niveau du modèle LD faut-il l'intégrer? Et quelle serait sa structure?
3. L'élément prérequis (**prerequisites**) (cf. **Figure II.2**), exige des apprenants certaines connaissances antérieures nécessaires pour atteindre l'objectif d'apprentissage. Il prend un seul aspect de différenciation entre les apprenants, c'est celui de connaissances antérieures. Pourrait-il être étendu

de façon à contenir le profil de l'apprenant susceptible d'accomplir l'activité en question et utiliser sans difficulté les ressources associées à cette activité?

4. Dans ce cas, on se contenterait de l'élément *prerequisites* auquel on apportera certaines modifications.

## 2.2. Analyse détaillée

Il convient alors à partir de la première analyse que :

- a. Les éléments du Learning Design (LD) susceptibles d'être adaptés seront rattachés au profil de l'apprenant.
- b. Les valeurs constantes demeurent dans leurs éléments de design
- c. Les valeurs qui changent peuvent être rangées dans les associations (relations) entre ces éléments et l'élément profil de l'apprenant.

Dans un premier temps il faut donc : identifier tous les éléments du LD qui peuvent changer en fonction du profil de l'apprenant.

On sait que tous les aspects du LD sont contenus dans le fichier XML « imsmanifest.xml ».

Pour ce faire on peut suivre l'algorithme suivant:

POUR chaque élément du LD

FAIRE

SI (il existe une relation entre l'élément courant du LD et le profil de l'apprenant)

ALORS

- Les valeurs constantes de l'élément du LD seront maintenues dans le LD.
- Les valeurs qui changent seront rangées dans l'association qui existe entre l'élément du LD et l'élément profil de l'apprenant.

FSI

FFPOUR

Il en résulte que l'adaptation ne se fera que pour les éléments du LD ayant une relation avec le profil de l'apprenant.

### Quels sont les éléments du LD qui peuvent être en relation avec le profil de l'apprenant ?

De la première analyse nous avons identifié quatre éléments du LD (04) :

1. *Learner*
2. *learning-activity*
3. *play*
4. *environment*

Examinons les faits suivants en nous basant sur les niveaux A et B d'IMS-LD :

1. Le niveau A de la spécification IMS-LD contient le noyau de la conception pédagogique d'IMS-LD (les personnes, les activités élémentaires et les ressources) et leur coordination grâce aux éléments méthode, scénario et acte. Ceci prévoit simplement une série d'activités d'apprentissage ordonnées temporellement pour être exécutées par des apprenants dans un environnement, en utilisant les objets et/ou les services d'apprentissage.

Seulement les éléments *Learner*, *activity* et *environment* peuvent être adaptés, ce qui rejoint la première analyse, l'élément *play* ne peut pas l'être à ce niveau, car au niveau A, on ne peut concevoir que des scénarios prédictifs sans tenir compte dans l'enchaînement des activités des caractéristiques des apprenants.

2. Quant au niveau B de la spécification IMS-LD, il rajoute d'autres éléments au niveau A, à savoir :

- *Properties*
- *Conditions* et
- Les valeurs des éléments *when-property-value-is-set* et *change-property-value*.

Ces éléments additionnels permettent un séquençage plus élaboré et des interactions basées sur le profil de l'apprenant ainsi qu'un plus grand contrôle et une plus grande complexité dans le sens où ces éléments permettent de décider de l'évolution d'un scénario pédagogique à un moment donné ; par l'évaluation d'une expression on peut décider, en fonction de son résultat, quel parcours suivra le scénario.

Outre les éléments déjà mentionnés dans le niveau A on peut rajouter l'élément *play*.

Le niveau B s'impose comme le niveau le plus approprié pour interagir avec le profil de l'apprenant, donc avoir des relations avec ce dernier.

Fournir à l'apprenant un ensemble d'activités d'apprentissage et des ressources qui correspondent à ses caractéristiques individuelles telles que ses objectifs d'apprentissage, ses connaissances préalables font partie d'une approche pédagogique appelée l'apprentissage adaptatif dont nous avons abordés les principes, dans le chapitre III.

### 3. Démarche à suivre

Un apprentissage vise un objectif pédagogique exprimé par des verbes d'actions, ces derniers traduisent ce dont l'apprenant doit être capable d'accomplir une fois l'apprentissage achevé.

Cette finalité suppose au préalable des prérequis que l'apprenant doit avoir avant d'entamer cet apprentissage. Il en résulte donc, qu'à chaque apprentissage correspond un couple (objectif, prérequis)

Concevoir un cours dont le couple (objectif, prérequis) est défini, pour un public cible connu et globalement homogène est modélisé en invoquant juste le niveau A de la spécification IMS-LD.

Ceci n'est toujours pas le cas, les apprenants se distinguent l'un de l'autre par leurs connaissances du domaine à étudier, leurs préférences, leurs objectifs et surtout leurs connaissances antérieures. Cet ensemble d'informations pertinentes sur l'apprenant constitue ce que l'on peut appeler le « profil de l'apprenant ». Ce dernier constitue l'élément pivot. Il s'agit bien entendu de l'apprenant potentiel.

Notre démarche est la suivante :

- Nous présenterons au début les caractéristiques pertinentes qui nous semblent définir le profil de l'apprenant, et choisirons un modèle de l'apprenant.

- On tiendra compte des types de changements qui peuvent être apportés au Learning Design afin que ce dernier puisse s'adapter au profil de l'apprenant.
- Bien entendu, ces changements utilisent des éléments de la spécification IMS LD qu'on va identifier et voir comment peut-on les utiliser dans les phases conception (design-time) et exécution (runtime) afin qu'ils puissent prendre en charge les types d'adaptations.
- Finalement, et compte tenu de l'étroite relation qui existe entre ces éléments et le profil de l'apprenant, élément pivot sur lequel s'articule l'adaptation, nous proposons une extension du modèle de la spécification IMS-LD.

#### 4. Modèle de données du Profil de l'apprenant:

Quelles sont les informations pertinentes qu'il faut fournir au système d'apprentissage pour adapter l'apprentissage à des caractéristiques ou préférences de l'apprenant.

Des normes et des standards existent, ils reprennent toutes ces données pertinentes et offrent des modèles appelées : modèles de données des apprenants pour permettre leur représentation et une meilleure réutilisation.

Les types d'informations recueillies dans ces modèles présentent des différences. Un choix d'un modèle de données des apprenants s'impose. Parmi les différents modèles de données des apprenants, les principaux modèles existants sont :

- Le modèle apprenant PAPI (Public And Private Information for Learners)
- Le modèle IMS LIP (IMS Learner Information Package)

#### 5. Principaux modèles existants

##### 5.1. Présentation du modèle apprenant PAPI

**Le modèle PAPI** (*Public And Private Information for Learners*) est un standard développé au sein du groupe Learner Model Working Group (*IEEE P1484.2 Learner Model Working Group*) qui n'a pas été accepté comme une norme par l'ISO.

Son but est de :

- Permettre aux apprenants d'avoir leur modèle apprenant basé sur les standards qu'ils peuvent ainsi utiliser durant tout leur parcours de formation et durant leur expérience d'apprentissage et leur vie de travail.
- Permettre aux enseignants auteurs de développer les matériaux pédagogiques qui fourniront une instruction plus personnalisée et plus efficace.
- Fournir une base pour le développement des normes éducatives complémentaires, et pour faire ainsi un foyer d'apprentissage centré apprenant.
- Et fournir des conseils architecturaux aux concepteurs de système d'apprentissage.

Six types d'informations sont définis par ce standard, qui rend également possible l'extension de chacun d'eux.

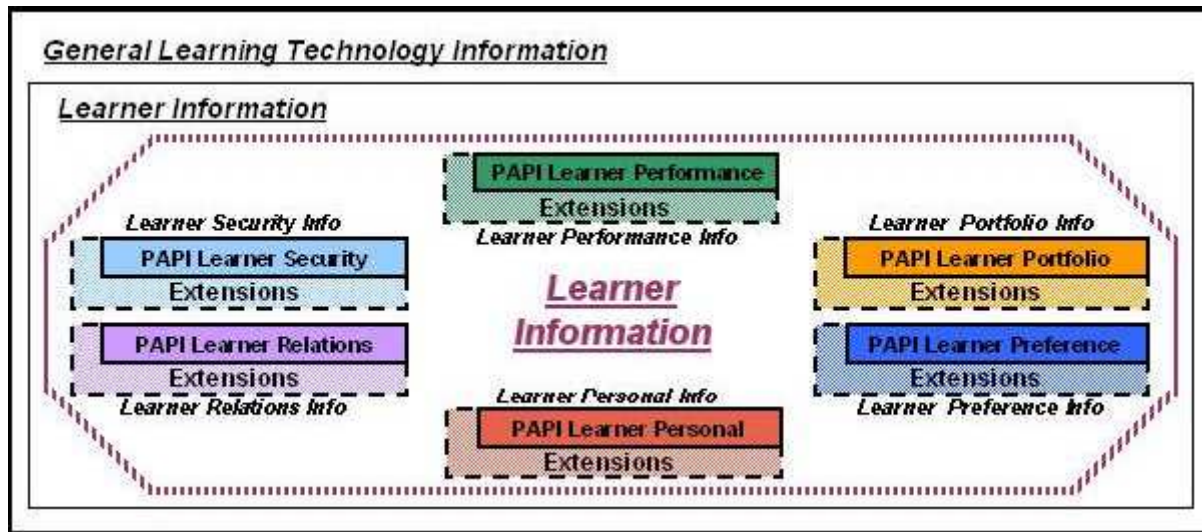


Figure IV.1 : Représentation de la proposition de standard PAPI Learner

Il contient six (06) types d'informations concernant l'apprenant :

- 1) **Learner Personal** : contient les informations personnelles sur l'apprenant. Il s'agit d'informations privées qui n'ont pas de lien direct avec l'apprentissage mais sont plutôt administratives telles que le nom, l'adresse, etc.
- 2) **Learner Relations** : contient les informations relationnelles, qui ont trait aux relations entretenues par l'apprenant avec les autres utilisateurs - formateurs, autres apprenants, etc.
- 3) **Learner Security** : contient les informations portant sur la sécurité : mot de passe, login, etc.
- 4) **Learner Performance** : contient les informations sur la performance de l'apprenant PAPI elles concernent les travaux effectués par l'apprenant, son travail en cours, et ses objectifs futurs.
- 5) **Learner Portfolio** : contient les informations « portfolio » qui constituent une collection représentative de travaux de l'apprenant utilisées en tant qu'illustrations de ses capacités.
- 6) **Learner Preference** : contient les informations liées aux préférences de l'apprenant (destinées à améliorer l'interaction homme-machine et à permettre l'adaptation automatique des systèmes aux besoins spécifiques de l'apprenant. Les préférences incluent des caractéristiques techniques, liées à l'interface, ou à la présentation des contenus. Ces préférences peuvent être explicitement identifiées par l'utilisateur ou être inférées à partir de son comportement.

Le standard PAPI Learner représente l'une des premières propositions qui offre un cadre qui organise les données de l'apprenant. Il existe un ensemble de données apprenant que ce standard ne prend pas en compte, et qui peuvent être échangeables entre les différents systèmes de e-formation. Ceci explique que cette proposition fasse l'objet d'une évolution par IMS dans son nouveau standard IMS LIP. C'est ce que nous présentons dans ce qui suit.

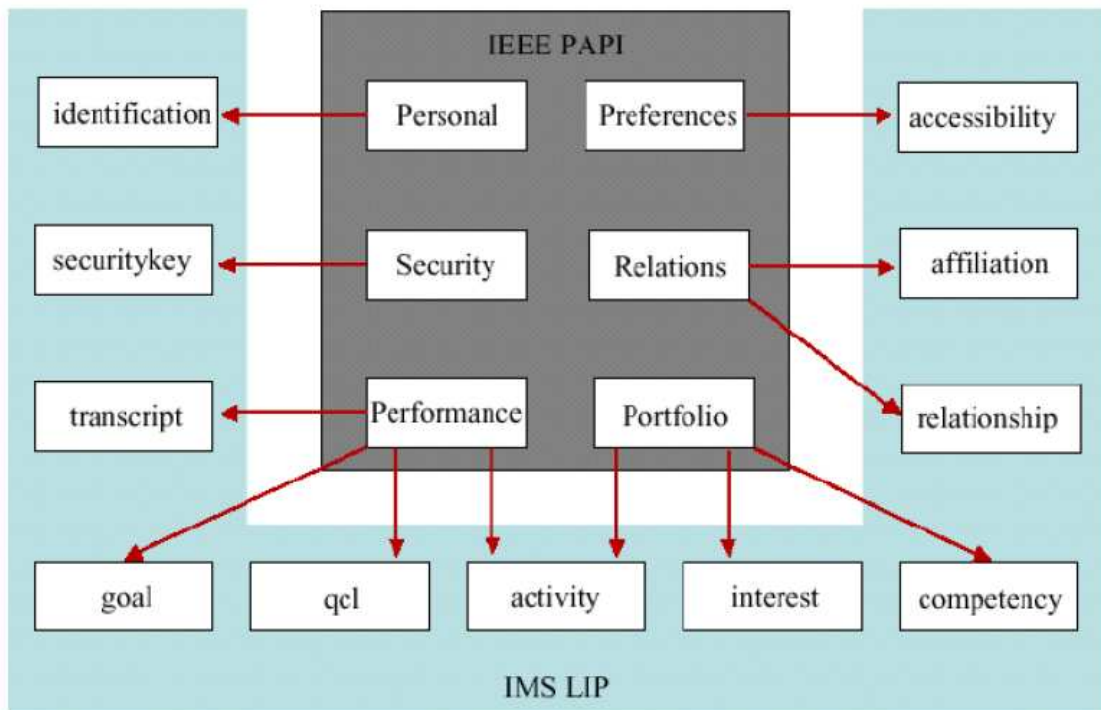
## 5.2. Présentation du modèle IMS LIP (*IMS Learner Information Package*)

### 5.2.1. C'est quoi IMS LIP

Sans aucun doute la partie la plus importante d'un processus d'apprentissage concerne les personnes qui y participent.

IMS LIP comprend les résultats obtenus dans le modèle PAPI (PAPI 2000), proposition de standardisation de l'information de l'apprenant réalisée par l'IEEE.

Les relations entre PAPI et IMS LIP sont détaillées comme suit (cf. **Figure IV.1**) :



**Figure IV.2 :** Relation entre IMS LIP et PAPI [IMSLIP-BEST 2001]

Cette nouvelle spécification montre comment stocker des informations sur un apprenant (ou un groupe d'apprenants) ou encore un producteur de contenu éducatif (auteur, pédagogue) [IMSLIP\_INFO 2001].

L'objectif est de définir une structure qui permet l'échange de paquets d'informations sur n'importe quel acteur impliqué dans le système éducatif.

En outre cette spécification doit permettre de :

- Disposer d'un registre de l'historique, des objectifs et des réalisations des apprenants.
- Obtenir un registre des apprenants et des enseignants dans le système.
- Ajuster les possibilités d'apprentissage aux besoins des apprenants.

Cette spécification introduit un nouveau concept appelé: Système d'Information de l'Apprenant (SIA). Ceci suppose toute une architecture montée autour de l'information obtenue des acteurs impliqués dans le processus d'apprentissage. La figure ci-dessous montre les composants d'un SIA.

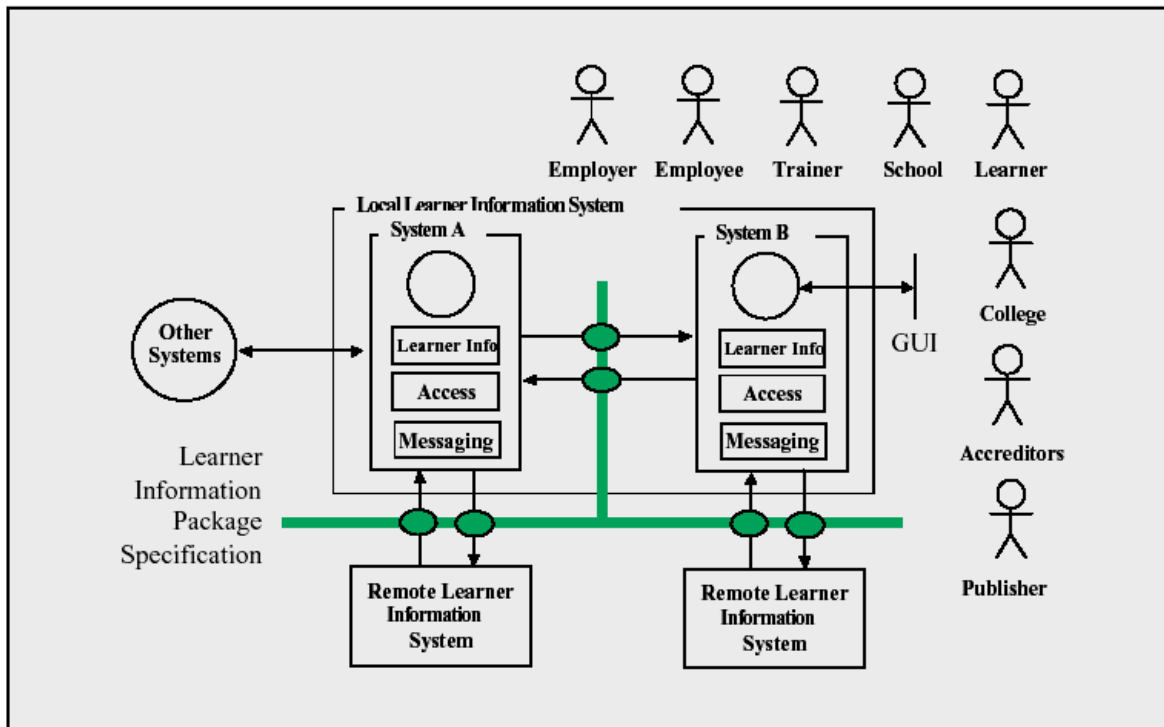


Figure IV.3 : Composants d'un SAI [IMSLIP\_INFO 2001]

Les principaux éléments (cf. Figure IV.2) sont les suivants:

- SIA Local (*Local Learner Information System*): ce sont les serveurs locaux accessibles par les utilisateurs.
- SIA à distance (*Remote Learner Information System*): ce sont les serveurs qui stockent des copies totales ou partielles des informations qui existent dans les serveurs locaux.
- Structures de données :
  - Informations sur l'apprenant : données actuelles sur l'apprenant
  - Accès, permissions pour la visualisation des données.
  - Messagerie, protocole de messagerie destiné à être utilisé pour l'échange de données.

Afin d'inclure dans un système d'apprentissage des informations sur les personnes, le système doit satisfaire certaines conditions, sans ça, la spécification n'est pas valide si le système qui l'implémentera ne satisfait pas les conditions suivantes :

- **Scalability** : Le système doit être prêt à gérer un grand nombre de dossiers qui comprennent des informations sur les utilisateurs. Une bonne façon de le faire serait de le faire en liaison avec la spécification IMS Content Packaging.
- **Privatisation et contrôle des données** : Les données stockées ne sont plus des cours, mais plutôt des données personnelles qui concernent des utilisateurs, et doivent donc être absolument confidentielles. La spécification est conçue de telle sorte qu'il soit facile d'implémenter un système de protection. Un système qui n'est pas en mesure de garantir la sécurité des données ne doit pas utiliser cette spécification.



- **Flexibilité** : Il est sauvegardé un historique sur le type de navigation de l'apprenant dans le cours. Ces données ont été liées à la structure du cours. En effet, les cours peuvent être mis à jour en permanence, le système doit surveiller les changements possibles dans les données de navigation et ce à chaque changement dans l'organisation d'un cours.

Le modèle de données proposé est structuré. En outre, au sein des documents de l'IMS se trouve un document liant ce modèle de données avec XML.

Cependant, il ne faut pas oublier que c'est une proposition, libre à ceux qui veulent utiliser un autre moyen de stockage (une base de données, par exemple)

L'information est stockée selon deux types :

1<sup>er</sup> type de données : les données stockées sont de type nom de l'apprenant, les cours qu'il a accompli, ou ses préférences d'affichage des cours.

2<sup>ème</sup> type de données : les données stockées sont une série de métadonnées relatives à chacun des domaines où les données sont stockées. Ces métadonnées sont du type : informations temporelles sur les données, informations concernant la sécurité et informations sur l'identification et l'indexation.

### 5.2.2. Aspect de IMS LIP

Les informations qui peuvent être stockées dans les registres LIP correspondent aux catégories suivantes: Registres éducatifs : Informations relatives à la réussite des apprenants dans leur vie scolaire.

- Registre des activités : Les activités réalisées par l'apprenant.
- Registre professionnel : Activités professionnelles et affiliations à des groupes.
- Curriculum : Expérience de travail, qualifications et des diplômes scolaires.
- Registre de formation continue : Réalisations de l'apprenant classées par date.
- Registre de services à la communauté : Les activités réalisées par l'apprenant supportées par la communauté.

Il est important de noter que le modèle complet de données de l'ensemble des informations sur les apprenants se compose de trois parties, alors que cette spécification traite uniquement la première partie :

- Learner Information Package : contient les structures de données avec des informations sur l'apprenant.
- Learner information exchange : Offre des facilités pour ajouter des paquets, procéder à des transactions et la messagerie.
- Learner information query : fournit le mécanisme de recherche d'information des apprenants.

Les deux derniers sont en dehors du champ d'application du présent cadre d'étude et ne seront pas traités ici.

Learner Information Package offre une structure de données sous-jacente avec onze (11) domaines. Ces champs sont destinés à couvrir toutes les informations utiles provenant d'un apprenant, comme illustré (cf. **Figure IV.4**).

Avant de commencer l'analyse des domaines qui constituent cette spécification, il est à noter que ces champs se trouvent dans un fichier XML qui porte le nom de registre de l'apprenant.

Ainsi, le nom d'un fichier qui répond à cette spécification sera de la forme : *login\_apprenant.xml*.

A l'intérieur de ce fichier, nous trouvons l'élément racine *<learnerinformation>*, d'où partent tous les champs que nous verrons par la suite.

Le modèle IMS LIP définit donc une structure XML en onze catégories pour pouvoir être importée ou exportée entre des systèmes interopérables.

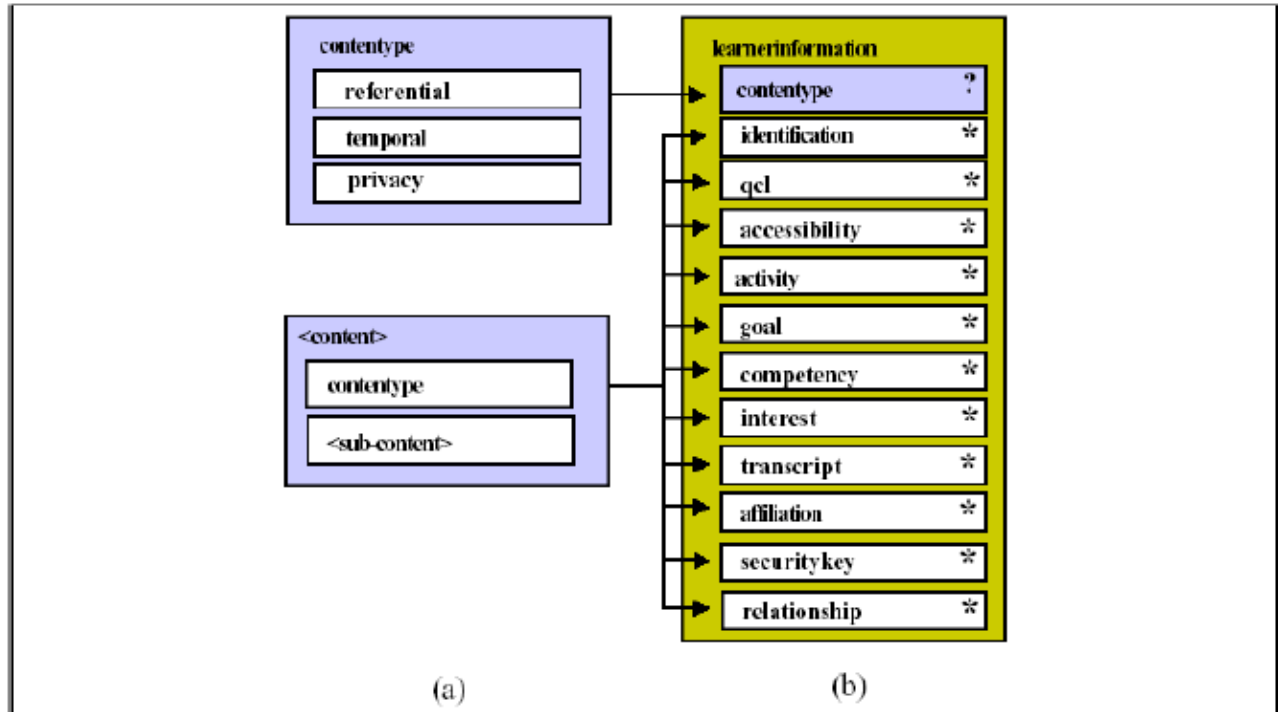


Figure IV.4 : Structure de données de la spécification LIP [IMSLIP\_INFO, 2001]

Comme nous pouvons le constater, il y a onze (11) champs clés qui composent le modèle de données de la présente spécification. Ces fichiers stockent des informations concernant l'apprenant. Nous allons les examiner une par une :

**1- Identification :** Stocke les informations nécessaires à l'identification de l'apprenant. Y compris, entre autres, les données composées de type :

- *<name>*, dans lequel est stocké l'identité de l'apprenant.
- *<address>*, qui comprend tout type d'informations sur la domiciliation de l'apprenant.
- *<contactinfo>*, où l'on trouve toutes sortes de détails pour contacter le candidat.
- *<demographics>*, stocke les données relatives à la naissance de la personne.

**2- Goal :** Contient les objectifs et les aspirations personnelles de l'apprenant. Il peut également inclure un historique des résultats obtenus par l'apprenant. Pour rendre ceci plus intuitif, les objectifs sont partagés en d'autres objectifs plus petits pour que le suivi de cet historique soit plus détaillé. Ne comprend pas des éléments composés, mais l'imbrication des objectifs est permise. Inclut des éléments tels que :

- *<contentype>*, qui donne un identificateur à l'objectif.
- *<priority>*, qui assigne une priorité.
- *<date>*, qui stocke les dates pertinentes pour l'objectif.
- *<status>*, qui informe de l'état dans lequel se trouve l'objectif en ce moment.

**3- QCL** (Qualifications, Certifications and Licenses). Ici, il est fait référence aux qualifications, aux certifications et aux licences obtenues par l'apprenant tout au long de sa formation scolaire et professionnelle. N'inclut pas des éléments composés. Inclut des éléments du type:

- <contentype>, qui attribue un identifiant à l'élément parent QCL.
- <type>, informe du type de l'élément QCL qui se traite.
- <organization> organisme qui a accordé le QCL.
- <date>, date à laquelle il a été accordé.

**4- Activity** : Stocker des informations relatives aux activités liées au travail et à la formation d'une personne. Inclut les éléments suivants:

- <contentype>, de nouveau utilisé pour assigner un identifiant à l'activité.
- <status>, nous informe sur l'état actuel de l'activité.
- <units>, fournit le nombre de crédits (ou similaire), ce qui est équivalent.
- <learningactivityref>, comprend une référence externe à l'activité à laquelle il se réfère.
- <Definition> stocke une description des outils pédagogiques qui ont été utilisés lors de l'activité.
- <Testimonial> présente les capacités de l'apprenant du point de vue du professeur ou du tuteur de l'activité.
- <evaluation>, stocke les résultats obtenus par l'apprenant en formation. Il est important de noter qu'il est compatible avec les résultats obtenus dans le cadre de la spécification Question & Test Interoperability.

**5- Competency** : stocke une description des compétences acquises par l'apprenant. Ces compétences peuvent être liées une formation professionnelle (décrite dans <activity>) ou un processus universitaire (décrit dans <QCL>). Il s'agit d'un élément très simple qui n'a pas d'autres éléments composés. Parmi les éléments les plus intéressants on peut citer:

- <contentype>, ajoute un identifiant à l'habileté.
- <exrefrecord>, Ici est stockée l'information concernant la compétence, ou un lien vers un archive qui contient cette information.

**6- Transcript** : Cet élément est utilisé pour stocker les données sur les bulletins de notes de l'apprenant, les résultats qu'il a obtenu dans une institution académique déterminée (Bulletins de notes). Ces résultats dépendent de l'institution et se présentent sous des formes très variées. Il n'est pas obligatoire d'utiliser un schéma rigide, bien au contraire, cet élément accepte n'importe quel niveau de détail. Les éléments les plus significatifs sont les mêmes que ceux déjà cités dans l'élément competency.

**7- Accesibility** : Détaille les options d'accessibilité de l'apprenant. Parmi ces options, on peut citer, les préférences physiques, techniques et cognitives, comme le langage préféré ou les handicaps éventuels de l'apprenant. En plus on peut trouver également les capacités que possède l'apprenant pour interagir avec les médias éducatifs. Les éléments les plus importants sont :

- <language>, conserve une évaluation de la capacité d'expression orale, écrite et de lecture de l'apprenant et de chaque langage que connaît l'apprenant.
- <preference>, stocke les préférences de l'apprenant pour interagir avec l'environnement éducatif, tels que la technologie d'entrée/sortie nécessaire, les styles d'apprentissage qui lui correspondent le mieux et l'environnement physique préféré.

- <disability>, c'est ici que sont énumérées les incapacités de l'apprenant, par exemple, si un apprenant ne voit pas bien, cette incapacité sera stockée pour agrandir la police des lettres dans l'unité d'apprentissage en cours.

**8- Interest** : inclut une description des intérêts de l'apprenant. Chaque intérêt est stocké avec sa propre structure, ses éléments les plus importants sont :

- <contenttype>, qui contient l'identificateur de l'intérêt, marcas de tiempo , et les droits d'accès à ces données.
- <product>, inclut un matériel concret significatif sur l'intérêt de l'apprenant, i.e, une photo d'un œuvre si son intérêt est la peinture.
- <description>, présente une description de l'intérêt.

**9- Affiliation** : Cet élément est utilisé pour conserver les données correspondant aux organisations auxquelles est affilié l'apprenant. Ces organisations peuvent être de type civil, social, militaire, etc. Il est important de noter qu'il s'agit d'un élément unique, ce qui veut dire qu'il existe un seul élément par affiliation. Les éléments les plus significatifs sont :

- <affiliationid>, conserve le numéro de l'affiliation à l'organisation.
- <role>, est un élément composé qui stocke le rôle que tient l'apprenant dans l'organisation à laquelle il est affilié. Possède également des données en référence de l'affiliation de type date, état, etc.
- <organization>, nom de l'organisation ou de la compagnie.

**10- Securitykey** : est utilisée pour stocker les clefs et les codes de sécurité affectés à l'apprenant et nécessaire à son interaction avec le système. Son champ le plus remarquable est :

- <keyfields>, Dans cet élément est définie le type de la clé qui est stockée, ainsi que la clé elle-même.

**11- Relation** : cet élément peut être considéré comme un joker dans lequel sont stocké toutes les relations qui peuvent exister entre les dix autres éléments. Au lieu d'inclure ces relations dans chaque élément, elles sont groupées dans cet élément. L'élément le plus significatif est :

- <tuple>, est un élément composé qui contient : <tuplesource>, qui indique le premier composant de la relation; <tupledest>, qui stocke le deuxième composant de la relation; <tuplrelation>, qui indique la relation entre les deux.

Il y a un élément commun à tous les champs (cf. **Figure IV.3**). Il s'agit de l'élément <contenttype>. Cet élément est le contenant de l'information de contrôle utilisé pour décrire les informations de l'apprenant [IMSLIP\_BIND 2001]. Cette information est du type:

- **Referential**. Information qui attribue un identificateur unique à la donnée composée. Grâce à ceci, des recherches et des manipulations sur les données peuvent être faites de manière univoque.
- **Temporal**. Information axée sur le développement temporelle de la donnée.
- **Privacy**. Les informations utilisées pour décrire la privatisation et garantir l'intégrité de la donnée.

L'IMS a également développé une DTD XML pour valider la conformité d'une spécification avec LIP.

### 5.2.3. IMS- IP comme support de personnalisation d'un apprenant

Il n'est pas impératif que les dossiers de personnalisation des apprenants possèdent tous les champs que couvre la spécification IMS-LIP.

On peut utiliser à titre d'exemple les champs <identification>, <qcl>, <activity>, <securitykey> dans le fichier des profils des utilisateurs. Une sauvegarde de la navigation réalisée par l'apprenant au cours de chaque cours est faite à l'intérieur des fichiers de personnalisation des apprenants.

Ceci implique une référence à chaque cours des archives visitées par l'apprenant.

Au cas où l'on veut modifier un cours, ceci se répercutera sur l'organisation du cours et se verra dans le manifest correspondant au cours (contenu dans le IMS-CP) alors que le fichier de personnalisation de l'apprenant contenant la sauvegarde de la navigation restera inchangé [Borja, 2003].

Il se pose alors un problème : tous les dossiers des étudiants qui ont visité un cours dont la structure a changé deviennent inutilisables.

Ce problème a une solution difficile et coûteuse ; la modification d'un cours entraîne à écarter tous les fichiers de personnalisation des apprenants qui l'ont visité, ou du moins, éliminer les arbres de navigation qui les contiennent.

Une solution acceptable serait [Borja, 2003]: Les contenus existants des cours sont fixés. C'est-à-dire que les contenus qui existent ne peuvent ni être supprimés ni être modifiés. Ce qui est possible, c'est de rajouter des parties à un cours, à condition que cela n'implique pas de changement dans son organisation.

## 6. Identification des éléments de la spécification IMS LD touchés par les changements :

L'adaptation des parcours pédagogiques (learning flow), des contenus (contents) et des présentations (interface) aux besoins des apprenants nécessitent des données reconnues pertinentes sur ces apprenants.

### 6.1. Premier type de changement apporté par le Learning flow :

Pour un apprenant, une leçon, un cours, un module... est un ensemble d'activités ordonnées et structurées qu'il doit réaliser.

<u>Compte tenu de:</u>	- La compétence de l'apprenant	et/ou
	- Son choix de parcours,	
<u>Certaines activités :</u>	- Remplaceront d'autres,	
	- S'ajouteront à celles déjà prévues	ou
	- Deviendront invisibles à l'apprenant.	
<u>Produisant ainsi :</u>	- Différentes manières des séquences des contenus du cours.	
<u>Le parcours d'apprentissage est alors :</u>	- Dynamique	et
	- Est personnalisé pour chaque utilisateur de sorte qu'il puisse prendre des <b>itinéraires différents qui dépendent de son profil.</b>	

### Exemples :

#### 1) Compte tenu de la performance de l'apprenant et de son choix de parcours :

Proposer à un apprenant deux parcours d'apprentissages différents le premier est thématique, le second est historique, les deux parcours d'apprentissage visent le même objectif d'apprentissage.

2) Compte tenu de la compétence de l'apprenant :

A la suite du résultat du test d'évaluation proposé à l'apprenant, certaines activités seront affichées cachées ou remplacées. Cela suppose évidemment que toutes ces activités ont été prévues donc existent au niveau du LD.

3) Compte tenu du choix de parcours de l'apprenant :

Un full contrôle est donné à l'apprenant où il peut choisir à volonté parmi un ensemble de choix la méthode qui lui semble le mieux convenir à son profil pour suivre un cours.

Les inputs d'un tel système d'apprentissage seraient :

- L'utilisateur et un ensemble de règles prédéfinies

Les outputs seraient :

- Un parcours d'apprentissage

La description du learning flow adaptatif est principalement basée sur quatre (04) éléments de la spécification IMS-LD disponibles au niveau B de la spécification, ces éléments sont :

- *Properties*
- *Calculations*
- *Global elements*
- *Conditions*

L'élément *monitoring services* peut être rajouté pour suivre la trace du comportement de l'apprenant et donc adapter le flow dynamiquement.

Les *properties* dans IMS-LD sont des variables qui sont déclarées dans le LD (Le fichier imsmanifest.xml) et sont :

- Utilisés et visualisés par la le biais d'éléments appelés *Global elements* qui se trouvent dans des fichiers XHTML.
- Utilisés à l'intérieur du LD dans les *conditions* ou les *calculations*.

*Global elements*, *Conditions* et *Calculations* seront détaillés par la suite.

Les valeurs dans les *properties* peuvent être de différents types comme Integer, Boolean, Float, File, etc.

Il y a cinq (05) types de *properties* déjà définies dans le premier chapitre, on va donner des exemples pour chaque type :

1. *Local properties*. Peuvent par exemple représenter une **question dans un questionnaire**, ou une **introduction commune à un cours**.
2. *Local personal properties*. Peuvent par exemple contenir le **nom de l'apprenant** ou la **réponse donné par l'apprenant à une question d'examen**.
3. *Local role property*. Peuvent par exemple représenter **deux rôles**, chaque rôle indiquant un groupe différent. Chaque groupe doit formuler une proposition de la solution d'un problème.
4. *Global personal properties*. Peuvent par exemple contenir des informations personnelles d'un étudiant, **son nom, son adresse, son email, ses scores, sa compétence, son niveau, ses réalisations**, etc...
5. *Global properties*. Peuvent contenir par exemple une **valeur numérique seuil pour une évaluation**.

Nous pouvons conclure que :

Le profil de l'apprenant, notre élément pivot, modélisé par IMS-LIP, sera stocké dans des *properties* de type *Local personal properties* ou *Global personal properties*.

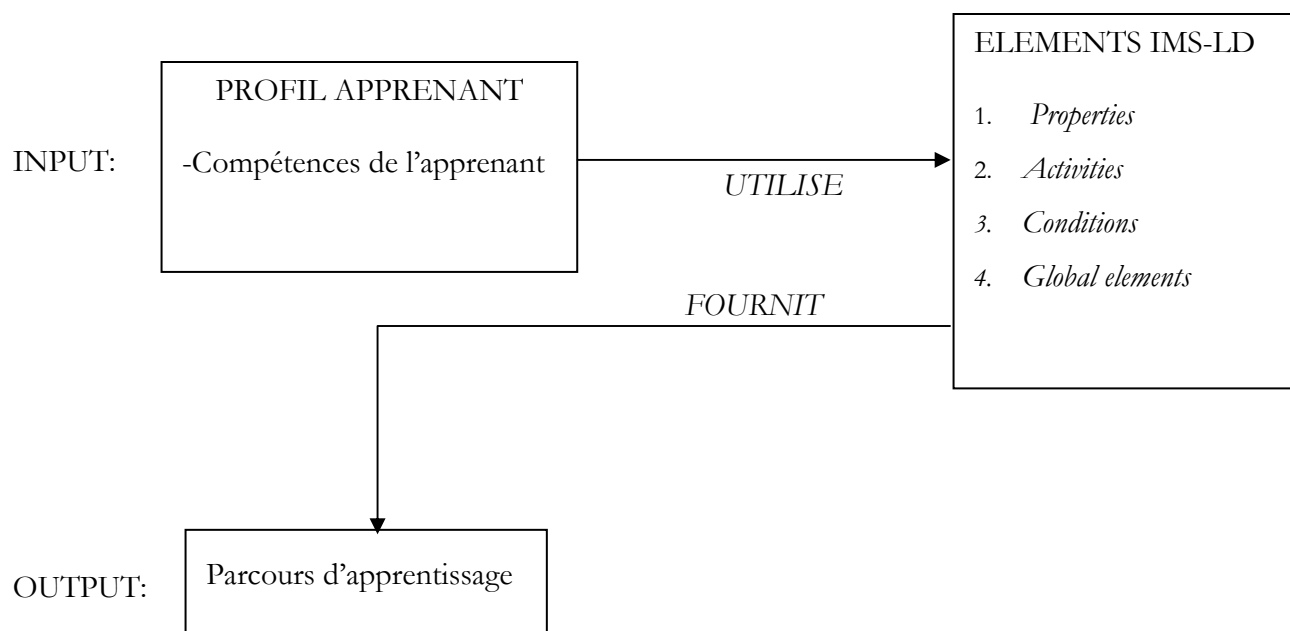
Et selon les valeurs de ces *properties*, auxquelles on pourra y accéder à travers l'élément *global elements*, plus précisément par le biais des éléments *view-property* et *set-property* et qui seront testées par l'élément *Conditions*, le parcours d'apprentissage (learning flow) sera modifié en conséquence.

Nous sommes dans les cas où le processus d'apprentissage est dynamiquement adapté afin d'expliquer le contenu du cours de différentes manières.

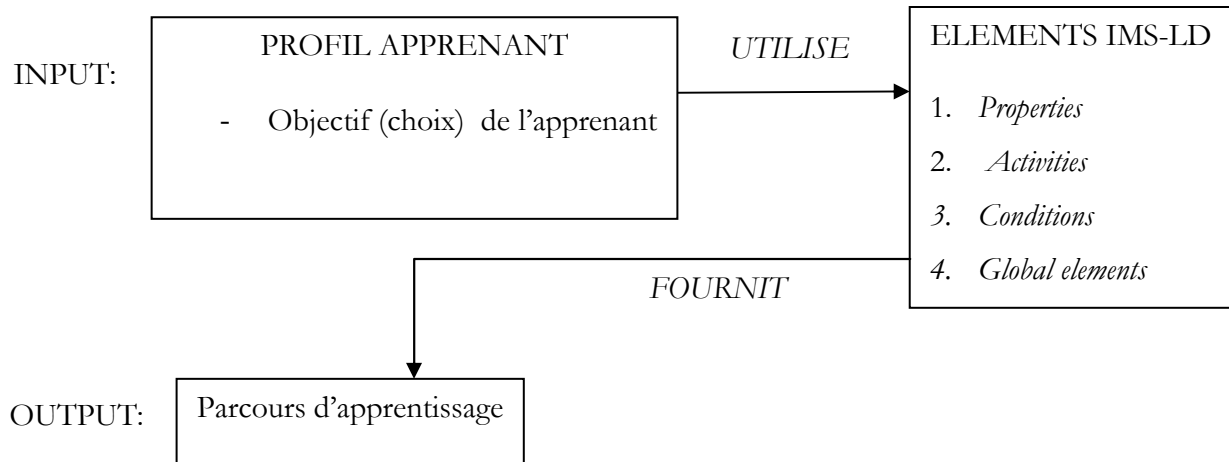
Il faudrait alors:

- Visualiser/ Cacher des éléments dans le learning flow. Certaines activités décrites par leurs environnements (Services, objets d'apprentissage) seront visibles ou cachées à l'apprenant. Les éléments IMS-LD utilisés sont : *Show* et *Hide*. On visualisera des activités et on cachera d'autres.
- Offrir une sélection ou une séquence d'activités ; l'attribut *structure-type* de l'élément *activity-structure* qui indique s'il s'agit d'une *sequence* ou d'une *selection* d'activités
- Fournir des nouveaux itinéraires *on-the-fly*, (à la volée) selon l'intervention de l'apprenant ou du tuteur par le biais du l'élément *monitoring*.

### Un ensemble de règles adapte le learning flow selon les performances de l'apprenant



**Figure IV.5 :** ensemble de règles adapte le learning flow selon les performances de l'apprenant

**L'utilisateur lui même modifie le learning flow selon ses décisions personnelles**

**Figure IV.6 :** L'utilisateur lui même modifie le learning flow selon ses décisions personnelles

Ce sont les valeurs des *properties* testées au niveau des éléments *conditions* qui modifient le learning flow et le contenu.

On déclare la propriété *locpers-property* au niveau de l'élément IMS-LD *component*, à laquelle on peut affecter le niveau de l'apprenant, sa qualification ou une autre caractéristique du profil de l'apprenant obtenue directement ou suite à un test subi par l'apprenant.

Et selon la valeur prise par cette propriété qui sera testée par l'élément IMS-LD *conditions*, le *learning flow* changera en conséquence, comme le montre les portions de programmes XML du fichier *manifest.xml*

De même que le contenu de cette propriété pourrait changer dynamiquement en donnant la main à l'apprenant pour modifier le contenu de la propriété pendant l'exécution par le biais de l'élément *Global elements* plus précisément *set-property* se trouvant dans le fichier XHTML.

```

<locpers-property identifier= "constante chaîne de caractère">
  <title> titre </title>
  <datatype= "type"/>
  <initial-value> valeur initiale </initial-value>
</locpers-property>
  
```



```

<if>
  <is>
    <property-ref ref="constante chaîne de caractère"/>
    <property-value>valeur</property-value>
  </is>
</if>
<then>
  <show>
    <activity-structure-ref ref="identifiant de la structure d'activité à visualiser"/>
  </show>
  <hide>
    <activity-structure-ref ref="identifiant de la structure d'activité à cacher"/>
  </hide>
</then>

```

## 6.2. Deuxième type de changement apporté par le Contenu

Dans ce qui précède, nous avons vu que le learning flow est principalement centré sur la séquence des activités définies dans le LD.

L'adaptation basée contenu est quant à elle :

- Centrée sur les informations contenues dans les activités, et
- Sur l'activité elle-même.

Les ressources et les activités changent dynamiquement leurs contenus actuels.

Le changement s'effectue à l'intérieur de l'activité.

Ce changement peut être déclenché par :

- Le choix de l'utilisateur qui décidera du niveau de granularité qu'il veut, ou
- Compte tenu de sa compétence, un niveau de granularité lui sera visualisé.

### **Exemple :**

Deux d'apprenants qui veulent suivre un cours sur l'écriture d'instructions de boucle en langage java, l'un deux a un profil d'informaticien programmeur et connaît parfaitement les boucles, l'autre a un profil de mathématicien et n'a aucune notion sur les boucles, les deux apprenants ont un même objectif pédagogique qui est celui d'être capable d'utiliser les boucles en java. En ce qui concerne le premier apprenant, les activités qui lui seront assignées ne contiendront pas des notions sur les boucles alors que pour le second ces notions sont plus que nécessaires et feront partie des activités à réaliser.

Le contenu d'une activité nécessite une ressource **liée** à l'élément *activity-description*.

Bien que ce link ne puisse pas changer pendant l'exécution, trois autres éléments peuvent être modifiés dynamiquement :

- Le contenu se trouvant dans les ressources XHTML, en définissant des classes et des couches DIV qui peuvent être visualisées ou cachées selon certains paramètres.
- Le contenu des *properties* prédéfinies qui peut être remplacé à la volée (*on-the-fly*) par un autre contenu.
- Le contenu d'une activité peut être adapté en visualisant ou en cachant un ou plusieurs environnements liés à l'activité.

La modification des contenus liés à des ressources fixes basée sur des outils externes. Par exemple une ressource liée à un service wiki hébergée à l'extérieur de l'unité d'apprentissage IMS-LD, pourrait adapter son contenu sur la base des contributions des utilisateurs, tuteurs ou auteurs.

- 1) La première approche est centralisée sur l'utilisation des *flag properties* qui visualisent ou cachent une certaine couche d'information (telle que DIV layer dans le fichier XHTML).

#### Démarche :

- *Flag propriété* = 'V1' (initialisation)
- Une activité n'est terminée que si *flag propriété* = 'V1' (*complete-activity*)
- L'utilisateur interagit avec le LD par le biais des *global elements (set-property)* contenu dans des fichiers externes XHTML en répondant à des questions, ses réponses seront stockées dans d'autres *properties*.
- Cet ensemble de réponse contenu dans ces *properties* sera soumis à un ensemble de règles définies dans la partie *conditions* de l'élément *method*, qui permettront de cacher ou de visualiser certaines couches d'information (telle que DIV layer dans XHTML) et d'affecter à *Flag propriété* la valeur 'V1', ce qui permettra l'achèvement de l'activité en cours et le passage à l'autre activité.

- 2) La seconde approche de l'adaptation permet la modification du contenu de la propriété prédéfinie à l'intérieur de l'activité.

#### Démarche :

- Dans un premier temps, pendant la conception du LD, dans le fichier *imsmanifest*, la *propriété* est définie et est configurée.
- Dans un second temps, pendant l'exécution, on change le contenu de la *propriété* en utilisant un fichier externe XML par le biais des *global elements (set-property et view-property)* pour configurer et visualiser le contenu de la *propriété*.

### 6.3. Troisième type de changement apporté sur l'interface

Ce type d'adaptation est basé sur des options, des types de navigation et des facilités de visualisation. Pour une facilité d'utilisation et une adaptabilité de l'interface, les éléments et les options de l'interface peuvent être redéfinis par l'utilisateur selon son profil.

Grâce à des menus d'options, l'apprenant pourra choisir le type de navigation pour une facilité de manipulation ou personnaliser les couleurs et les caractères pour une facilité de visualisation.

Les options et l'apparence de l'interface sont adaptées selon la requête de l'utilisateur.

Les players IMS-LD actuels ne peuvent pas changer la taille et la position de leurs panels, de leurs espaces de travail, de leurs fenêtres ou n'importe quel autre composant qui facilite la navigation.

Ces players ne peuvent pas changer les propriétés de base, telles que la taille de la police, la couleur de la police ou le style de la police.

Néanmoins, au niveau de l'unité d'apprentissage, cette adaptation peut se faire, en faisant appel aux éléments *DIV layers* appartenant aux fichiers XHTML utilisé comme ressources externes par les éléments du LD et l'élément *environments* du LD.

L'adaptation basée interface est tout à fait différente de l'adaptation basée contenu.

L'adaptation basée contenu est basée sur les informations appartenant à une activité qui sont visualisées et traitées. L'adaptation basée interface est basée sur les options, les facilités de navigation et de visualisation.

Les options, le look de l'interface sont adaptés selon les préférences de l'utilisateur.

En ce qui concerne les activités, plusieurs couches DIV ou activités d'apprentissage peuvent être configurées avec des visualisations différentes pour le même contenu.

Par exemple, lier le même fichier à différentes feuilles de style CSS.

Démarche :

Visualiser l'activité sélectionnée (donc le même fichier avec une feuille de style correspondante).

- Définir les différentes activités dans l'élément *components*, chacune d'elles liée à une ressource définie dans l'élément *item* correspondant.
- Dans l'élément *ressource*, les ressources référencées dans les *items* utilisent le **même fichier html** et des feuilles de style différentes.
- Finalement, selon les valeurs des *properties* contenant le choix de l'apprenant quant à l'interface retenue, l'élément conditions par le biais des éléments *hide* et *show*,

**Résultats :**

- **Les valeurs** des *loppers-properties* changent, elles ne sont pas constantes, elles prennent différentes valeurs en fonction du profil de l'apprenant.
- Elles peuvent être paramétrées
- **Leur type** est variable ; il peut être :
  - **réel** (dans le cas où elles représentent par exemple le résultat d'un test indiquant la performance d'un apprenant), l'ensemble des valeurs permises est une note  $\geq 0$  ou un pourcentage compris entre 0 et 100.
  - **texte** (dans le cas où elles représentent par exemple une réponse donnée par l'apprenant, ou un choix répondant à un objectif ou une décision de l'apprenant), l'ensemble des valeurs permises sont des chaînes de caractères.
- **L'intérêt du paramétrage** est de fournir un *learning flow* (parcours d'apprentissage) à l'apprenant, en fonction des valeurs contenues dans les *loppers-property* reflétant son profil.

Ainsi le Learning design étendu peut être modélisé.

**7. Extension du LD****7.1. Introduction**

Sans apprenant, il ne peut y avoir d'apprentissage, c'est l'élément clé sur lequel se base un enseignement, qu'il soit en présentiel, à distance ou mixte.

Le profil de l'apprenant constitue l'élément pivot sur lequel s'articule l'adaptation. L'auteur du design pédagogique doit en tenir compte s'il veut que son cours prenne en considération des apprenants qui se distinguent dans leur façon d'appréhender un enseignement.

Nous proposons alors d'étendre la spécification IMS-LD en y intégrant cet élément.

La structure des éléments *Local personal properties* et *Global personal properties* permettant la personnalisation d'une UA peut être entièrement assurée par IMS Learner Information Package (IMS-LIP)

Le modèle IMS-LIP s'avère être le modèle approprié qui va supporter ce profil. Son intégration dans le Learning Design ne pose pas de problèmes compte tenu des relations qui existent entre la spécification IMS-LD et la spécification IMS-LIP.

**7.2. Extension du LD**

Le modèle de l'apprenant ou profil de l'apprenant est supporté par le modèle IMS-LIP.

1. Le modèle de l'apprenant ne fait pas partie du Learning Design.
2. L'élément **person** du Learning design contient le profil de l'apprenant.
3. L'élément **property** du Learning Design est en relation avec l'élément **person** du Learning design.
4. Ainsi le profil de l'apprenant sera supporté par l'élément **property** du Learning design, plus précisément par les éléments *Local personal properties*.

5. La relation ***affect*** (agit) exprime le fait que l'élément externe **global elements** (qui ne fait pas partie du LD) agit sur l'élément **properties**, plus précisément met à jour ou visualise le contenu de l'élément **properties**.
6. La relation ***select*** (sélectionne) exprime le fait que l'élément **properties**, en utilisant l'élément **condition** du Learning Design sélectionne l'activité qui convient le mieux au profil de l'apprenant.

**Il est à noter** que le modèle de l'apprenant est également mis à jour par les éventuels changements ayant eu lieu pendant le déroulement du LD (runtime).

Le Learning Design (LD)

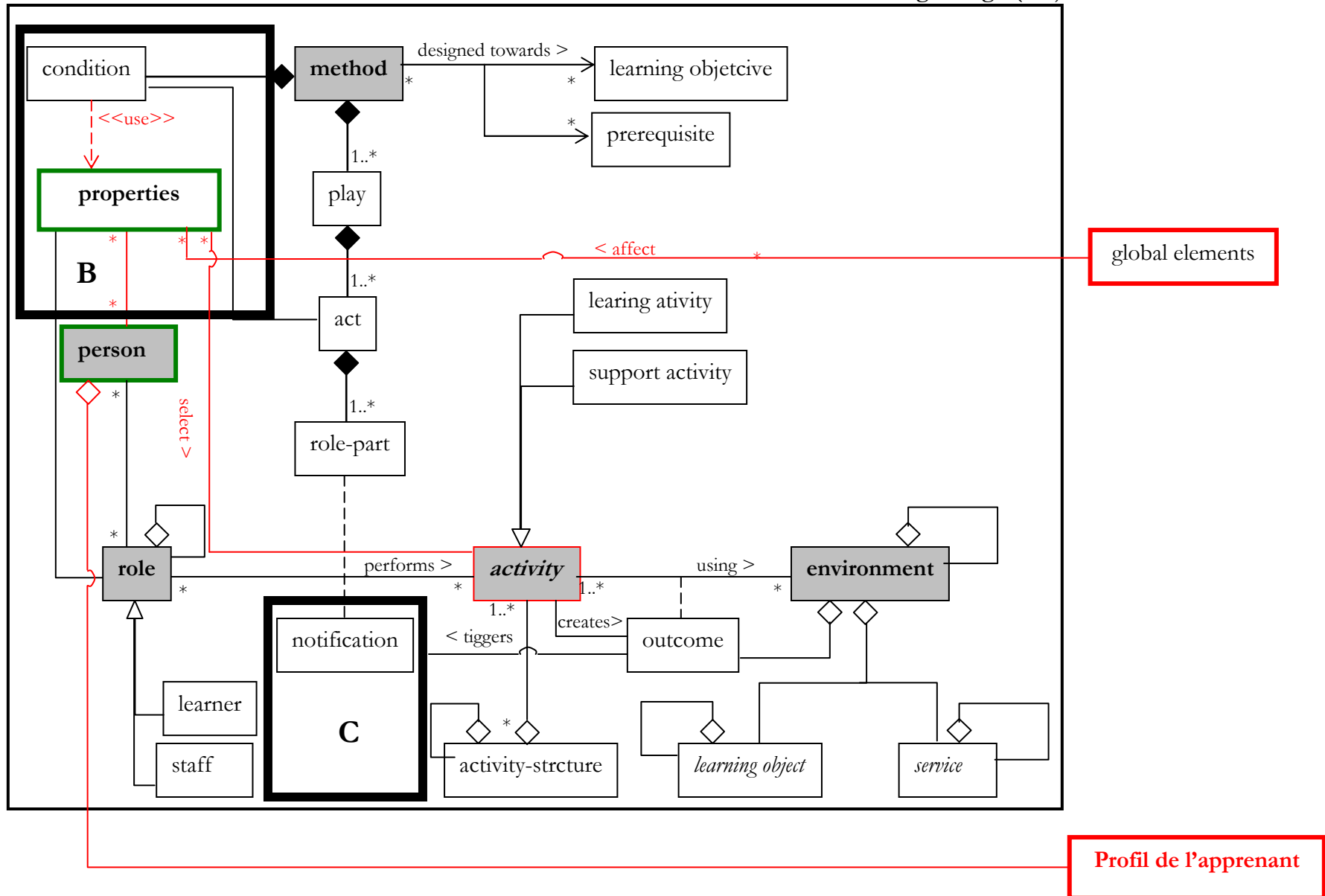


Figure IV.7 : Extension du Learning design

## 8. Conclusion

L'adaptation est un processus assez complexe prenant en compte plusieurs parties prenantes et plusieurs entrées: Apprenant, tuteur et un ensemble de règles.

IMS Learning Design semble être un langage expressif prometteur qui permet plusieurs types d'adaptation. En utilisant les éléments disponibles au niveau B, telles que :

*conditions, properties, calculations, global elements* et *monitoring services* pour modéliser et exécuter des UA riches et adaptatives.

Les possibilités d'adaptation supportées par IMS LD sont diverses. Sur les huit types d'adaptation décrits, nous identifions trois niveaux de support:

- a) *Learning flow, content, evaluation* et *interactive problem solving support* sont convenablement supportés;
- b) *User grouping, interface adaptation, adaptive evaluation* et *full modification of a course onthe-fly* sont partiellement pris en charge;
- c) Enfin, certaines questions restent en suspens, sans support pour toutes et qui sont : *dynamic modification of learning structure* et *method in run-time*, et *adaptive information filtering and retrieval*.

Cependant, une partie de ce manque de support est du à l'état actuel de l'outillage et non pas à la spécification IMS-LD elle-même.

Néanmoins, avec plusieurs types d'adaptation, il est également possible de fournir un support spécifique sur l'adaptation, c'est-à-dire relier une activité d'apprentissage à un outil externe qui fournit un service connexe et maintenir IMS LD comme un conteneur pour adaptation externe. Dans cette mesure, l'adaptation provient de l'extérieur d'IMS LD bien que le LD agisse comme un intégrateur.

Nous pouvons dire en conclusion qu'avec le support approprié (l'outillage), IMS LD peut construire des expériences d'apprentissage pour toutes les parties prenantes.

## Conclusion générale

Dans ce mémoire, nous avons présenté d'une façon détaillée la spécification IMS-LD en analysant la structure et l'organisation de ses éléments.

Nous avons également montré comment construire une unité d'apprentissage en utilisant la spécification IMS-LD.

Nous nous sommes intéressés plus particulièrement aux cours dont le public apprenant cible n'est pas connu à priori avec précision. Par conséquent les éléments pédagogiques susceptibles de changer doivent être ainsi suffisamment adaptés en vue de prendre en compte une variété de choix et de préférences.

Pour ce faire, nous nous sommes penchés sur la notion d'adaptabilité afin de faire ressortir tous les types de changements qui peuvent se produire dans un Learning Design qui prend en compte des profils d'apprenants hétérogènes, à savoir le Learning flow, le contenu et l'interface.

Nous avons identifiés les éléments du Learning design qui interviennent dans chaque type de changement et nous avons montré comment les implémenter en utilisant le niveau B de la spécification IMS-LD.

Finalement, nous avons étendu le modèle de spécification IMS-LD en vue de le rendre capable de prendre en charge un design pédagogique adaptable en faisant toujours appel au niveau B de la spécification IMS-LD.

Nous réfléchissons à faire évoluer cette notion d'adaptation, afin qu'elle prenne une autre dimension, celle du Learning design paramétré destiné aux auteurs de cours, un tel Learning design constituera un modèle suffisamment paramétré destiné à l'auteur et permettra ainsi de prendre en considération une variété d'utilisateurs auteurs caractérisés par des contextes techno-pédagogiques très différents.



## Références bibliographiques

- [Bekka, 2007] « Adaptation d'un dispositif de formation en ligne base sur les activités et les documents ». Mémoire de magister. Université Abou-Bakr-Belkaid Tlemcen. Bekkaoui. M. (2007).
- [Berk, 2008] « Communautés de Pratique du E-learning - CoPE : définition des concepts et proposition d'un langage de spécification des scénarios d'apprentissage ». Mémoire de magister. Institut National de formation en Informatique (INI) oued smar. Alger. Lamia BERKANI (2008)
- [Borja, 2003] « Estudio de la propuesta IMS de estandarización de enseñanza asistida por computadora ». Borja Manero Iglesias (2003)
- [Brusilovsky and Paylo, 2003] « Adaptive and Intelligent Web-based Educational Systems ». International Journal of Artificial Intelligence in Education 13: 156-169. Brusilovsky, P. and C. Paylo (2003).
- [Burgos et al., 2006a] « IMS Learning Design : la flexibilité pédagogique au service des besoins de la e-formation ». Daniel Burgos, Michel Arnaud, Patrick Neuhauser et Rob Koper (2006).
- [Burgos et al., 2006b] « How to represent adaptation in eLearning with IMS Learning Design ». Daniel Burgos, Colin Tattersall and Rob Koper (2006)
- [Burgos et al., 2006c] « Representing adaptive and adaptable Units of Learning ». Burgos, D., Tattersall, C., & Koper, R. (2006).
- [Burgos, 2008] « Extension of the IMS Learning Design Specification based on Adaptation and Integration of Units of Learning ». Thesis Doctoral Daniel Burgos Solans 25 avril (2008)
- [Cons, 2001] « Résolution du conseil sur le e-learning ». Conseil de l'Union Européenne, Journal Officiel des Communautés européennes, Vol. 20, n°7, 2001.
- [Educnet] Site Educnet : ressources pédagogiques numériques pour enseigner avec les technologies de l'information et de la communication. [<http://www.educnet.education.fr/superieur/glossaire.htm>]
- [IMS, 2001] « IMS Global Learning Consortium ». IMS Meta-data Specification Version 1.2.1 Finale Release. Technical report, (2001).
- [IMSLD03] « IMS Learning Design, Information model ». version 1.0 Final specification [http://www.imsglobal.org/learningdesign/ldv1p0/imsl\\_d\\_infov1p0.html](http://www.imsglobal.org/learningdesign/ldv1p0/imsl_d_infov1p0.html). 20 January (2003).
- [IMSLD Best Practice and implementation] « IMS Learning Design Best Practice and implementation Guide, version 1.0 final spec ». IMS Global Learning Consortium, (2003). [[www.imsglobal.org/learningdesign/ldv1p0/imsl\\_d\\_bestv1p0.html](http://www.imsglobal.org/learningdesign/ldv1p0/imsl_d_bestv1p0.html)]
- [IMS Learning Design] « IMS Learning Design Un langage de modélisation pédagogique ». Anne Lejeune, Olga Marino, 3ème École Thématique EIAH - du 1er au 6 juillet (2005).

- [IMSLIP-BEST 2001]** « IMS Learner Information Package Best Practice IMSLIP\_BEST ». Guide. Version 1.0 Final Specification, IMS Global Learning Consortium, Inc. (2001).
- [IMSLIP\_INFO 2001]** « IMS Learner Information Package Information Model ». Version 1.0 Final Specification, IMS Global Learning Consortium, Inc. (2001).
- [IMSLIP\_BIND 2001]** « IMS Learner Information Package XML Binding ». Version 1.0 Final Specification, IMS Global Learning Consortium, Inc. (2001).
- [Koper, 2001]** « Modeling Units of Study from a Pedagogical Perspective, the pedagogical meta-model behind EML ». Koper, R. (2001).  
[<http://eml.ou.nl/introduction/docs/pedmetamodel.pdf>]
- [Koper and Burgos, 2005]** « Developing advanced units of learning using IMS learning design level B ». Rob Koper, Daniel Burgos (2005).
- [Koper and Tattersall, 2005]** « Learning Design ». A Handbook on Modelling and Delivering Networked Education and Training. Heidelberg, Springer Verlag.  
Koper, R. and C. Tattersall, Eds. (2005).
- [Lejeune A., 2004]** « IMS Learning Design ». Lejeune A.nne, Distances et savoirs 2004/4, Volume 2, p. 409-450. (2004).
- [Paquette, 1998]** « L'ingénierie des interactions dans les systèmes d'apprentissage ». Paquette Gilbert ., Revue des sciences de l'Éducation, septembre (1998).
- [Pernin, 2006]** « LOM, SCORM et IMS-Learning Design : ressources, activités et scénarios ». Pernin, Jean-Philippe. Babel - edit -, L'indexation des ressources pédagogiques numériques. ENSSIB - janvier (2006)
- [Pernin, 2004]** « LOM, SCORM et IMS Learning Design : Ressources, activités ou scénarios ? ». Pernin, Jean-Philippe.  
L'indexation des ressources pédagogiques numériques, Lyon, (2004).
- [Reload LD Editor, ver 2.1.0]** « Reload Learning Design Editor, version 2.1.0 ». RELOAD Project, (2005).  
[[www.reload.ac.uk/leditor.html](http://www.reload.ac.uk/leditor.html)]
- [Reload, 2004]** « Reload Editor ». RELOAD Project, (2005).  
[<http://www.reload.ac.uk/editor.html>]
- [Specht et Burgos, 2006]** « Implementing Adaptive Educational Methods with IMS Learning Design ». Marcus Specht, Daniel Burgos Open University of the Netherlands (2006).
- [Smahi, 2008]** « Adaptation des scénarios d'apprentissage au moment de leur instanciation Approche basée sur les systèmes hypermédias adaptatifs ». Mémoire de magister. Université Abou-Bakr-Belkaid Tlemcen.  
Ismail Mohammed SMAHI (2008)
- [Towle & Halm, 05]** « Designing Adaptive Learning Environments with Learning Design ». Brendon Towle and Michael Halm.  
Learning Design, The Netherlands, Springer, pp. 215-226, (2005).

- [Van Rosmalen and Boticario, 2005]** « Using Learning Design to support design- and runtime adaptation ». Learning Design: A Handbook on Modeling and Delivering Networked Education.  
Van Rosmalen, P. and J. Boticario (2005).
- [Van Rosmalen et al., 2006]** « The Full Life Cycle of Adaptation in aLFanet eLearning Environment ». Peter van Rosmalen, Jesús G. Boticario and Olga C. Santos (2006).
- [Van Rosmalen, Vogten et al., 2006]** « Authoring a full life cycle model in standards-based, adaptive e-learning ». Educational Technology & Society 9(1): 72-83.  
Van Rosmalen, P., H. Vogten, R. Van Es, P. Van, H., P. Poelmans and R. Koper (2006).
- [Web3.2]** Glossaire de la e-formation.  
[http : //www.educnet.education.fr/superieur/glossaire.htm](http://www.educnet.education.fr/superieur/glossaire.htm)
- [Zarraonandia & al, 06]** « A late Modelling Approach for the Definition of Computer-Supported Learning Process ». Telmo Zarraonandia, Camino Fernández, Juan Manuel Dodero  
Proceeding of Adaptive Hypermedia, pp.241-251, (2006).