HAL
open science

Un Environnement Informatique pour l'Apprentissage Humain au Service de la Virtualisation et de la Gestion des Objets Pédagogiques

Julien Broisin

To cite this version:

Julien Broisin. Un Environnement Informatique pour l'Apprentissage Humain au Service de la Virtualisation et de la Gestion des Objets Pédagogiques. Informatique [cs]. Université Paul Sabatier - Toulouse III, 2006. Français. NNT : . tel-00367682

HAL Id: tel-00367682

https://theses.hal.science/tel-00367682

Submitted on 12 Mar 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Paul Sabatier - Toulouse III **École Doctorale Informatique et Télécommunications**
UFR Mathématiques, Informatique, Gestion

Un Environnement Informatique pour l'Apprentissage Humain au Service de la Virtualisation et de la Gestion des Objets Pédagogiques

THÈSE

présentée et soutenue publiquement le 29 Mai 2006

pour l'obtention du

Doctorat de l'Université Paul Sabatier – Toulouse III

(spécialité informatique)

par

Julien Broisin

Composition du jury

Président :	Florence Sèdes, Professeur, Université Paul Sabatier
Rapporteurs :	Alain Derycke, Professeur, Université des Sciences et Technologies de Lille Pascal Leroux, Professeur, Université du Maine Peter Sloep, Professeur associé, Open University of the Netherlands
Examinateurs :	Erik Duval, Professeur, Université Catholique de Leuven Florence Sèdes, Professeur, Université Paul Sabatier Philippe Vidal, Directeur de Recherche, Université Paul Sabatier
Invités :	Philippe Baqué, Ingénieur CICT Daniel Marquié, Responsable International E-Mi@ge, Université Paul Sabatier Jean-François Mazoin, Directeur IUT A, Université Paul Sabatier Maïa Wentland Forte, Professeur, Université de Lausanne

Institut de Recherche en Informatique de Toulouse – UMR UPS – INPT - CNRS 5505
Equipe Systèmes Informatique Et Répartition d'Applications

Résumé

Aujourd'hui, la Formation Ouverte et A Distance (FOAD) est de plus en plus répandue aussi bien dans les institutions publiques que privées, mais elle est loin de représenter une méthode d'apprentissage utilisée par tous. Un moyen de faciliter et d'encourager la conception de cursus pédagogiques par les éducateurs est de mettre à leur disposition une masse importante d'objets pédagogiques de natures variées tant au niveau des domaines d'apprentissage visés que du type de supports numériques utilisés, ainsi que de la langue dans laquelle ils sont produits.

Nos travaux exposent les capacités d'un Environnement Informatique pour l'Apprentissage Humain (EIAH) à fournir deux services complémentaires qui conduisent à la virtualisation des ressources pédagogiques : une vue unique d'un ensemble de ressources pédagogiques renfermées dans des viviers de connaissance distincts, et un accès facilité à celles-ci à travers les plates-formes d'apprentissage. Nous présentons une architecture ouverte basée sur des standards de l'e-formation établis ou en cours d'élaboration, et qui assure l'intégration des services énoncés ci-dessus au sein d'un EIAH. Nous présentons deux expérimentations déployées au sein de contextes différents qui valident notre approche et qui favorisent ainsi le partage et la réutilisation des objets pédagogiques.

La seconde partie de nos travaux porte sur la supervision des ressources, systèmes informatiques et utilisateurs impliqués dans le processus de virtualisation afin de faciliter aux éducateurs la recherche de matériel pédagogique pertinent pour leur contexte parmi le nombre considérable de ressources offertes par la fédération des viviers de connaissance. Nous présentons un modèle d'information décrivant les entités précitées et qui constitue une extension de CIM (Common Information Model) pour l'EIAH, ainsi qu'une infrastructure de supervision composée d'agents de gestion, d'un fournisseur de données, d'un gestionnaire centralisé et d'une application graphique de gestion. Nous appliquons ensuite cette infrastructure au sein de notre EIAH et de l'outil de recherche et d'indexation d'objets pédagogiques de la fondation ARIADNE. Enfin, nous exposons comment les informations statistiques recueillies peuvent être exploitées pour présenter aux concepteurs de cursus pédagogiques des ressources de qualité.

Mots-clés : environnement informatique pour l'apprentissage humain, objet pédagogique, vivier de connaissance, plate-forme d'apprentissage, intégration de services, modèle de l'information, supervision de systèmes.

Abstract

Today, e-learning is widely disseminated within both private and public institutions, but it is far from representing a learning process used by everybody. A means to facilitate and encourage conception of pedagogical distance trainings by e-trainers consists in offering them the opportunity to access a big pool of learning objects that take into account various learning contexts and numerical supports, together with a large diversity of languages.

Our works investigate the capabilities of a Web-based Learning Environment(WLE) to provide two complementary services that lead to learning objects' virtualization : a single view of a set of resources stored into various learning object repositories, and an easy access to these resources through the use of learning management systems. We present an open architecture based on ubiquitous e-learning standards which allows to integrate the above services within a WLE. Two experiments deployed within various e-learning contexts validate our approach and thus contribute to learning objects' share and reuse.

The second part of our works deals with the management of resources, computer systems and users implied in the virtualization process. Among the large amount of resources offered by the federation of several knowledge pools, we aim at helping educators to search for relevant pedagogical material according to their learning contexts. We present first an information model describing the above entities and that constitutes a CIM extension for WLE, and then a supervision infrastructure composed of two management agents, a data provider, a centralised manager and a graphical management application. Next we apply this infrastructure to our WLE and to the search and index learning objects tool of the ARIADNE foundation. Finally, we expose the way to benefit from statistical information that are gathered by our framework in order to offer to e-trainers some well-adapted resources.

Keywords : web-based learning environment, learning object, learning object repository, learning management system, service integration, information model, system management.

Remerciements

Cette thèse a été réalisée au sein de l'équipe Service IntEgration and netwoRk Administration (Administration de réseaux et intégration de services) du laboratoire IRIT (Institut de Recherche en Informatique de Toulouse) à l'Université Paul Sabatier. Je tiens à remercier toutes les personnes qui, par leur coopération, leurs discussions et leurs encouragements m'ont aidé tout au long de ces trois années.

Ce travail n'existerait pas sans l'aide du Professeur Philippe Vidal qui a assuré la direction de cette thèse avec compétence et parfaite disponibilité malgré un emploi du temps chargé. Je le remercie vivement pour sa confiance, pour avoir accepté d'être mon directeur de recherche, pour son expertise, pour les précieuses connaissances qu'il m'a transmises, mais aussi pour les nombreux et bons moments partagés au-delà du cadre scientifique. Je lui serai toujours reconnaissant.

Je souhaite aussi adresser mes remerciements les plus vifs à Messieurs les Professeurs Alain Derycke, Pascal Leroux et Peter Sloep pour avoir pris le temps de lire ce manuscrit, pour avoir accepté d'être rapporteurs et de participer au jury. Leurs remarques constructives et leurs suggestions ont permis d'aboutir à la présentation de ce travail. Aussi je remercie sincèrement Madame Florence Sèdes, Professeur à l'Université Paul Sabatier, pour la relecture du document, ses commentaires pertinents et sa participation au jury. Enfin, je remercie tout particulièrement Monsieur Erik Duval, Professeur à l'Université Catholique de Leuven, pour ses conseils, discussions et collaborations fructueuses menées au cours de la réalisation de ce travail, pour sa confiance et les moyens associés, et enfin pour avoir accepté de participer au jury malgré un quotidien très chargé.

Je souhaite aussi adresser mes sincères remerciements à Philippe Baqué pour sa disponibilité, son expertise et ses précieux conseils. Les expérimentations réalisées dans le cadre de ces travaux n'auraient pu voir le jour sans l'aide et les moyens qu'il m'a apportés.

L'aventure toulousaine n'aurait été possible sans la confiance que Daniel Marquié m'a accordée. Je le remercie vivement pour son accueil chaleureux, pour l'intérêt qu'il a porté à mon

travail, pour son soutien continuel, et pour sa bonne humeur et sa gentillesse de tous les jours. Merci aussi de m'avoir permis d'expérimenter ces travaux dans un projet tel que International E-Mi@ge.

Je tiens tout particulièrement à remercier Mme Michelle Sibilla pour m'avoir initié et donné goût à la recherche lors de mon stage de DEA, mais aussi pour l'aide précieuse qu'elle m'a apportée lors de la rédaction de ce document et autres publications.

Je tiens aussi à remercier les communautés Ariadne et Prolearn pour les discussions, réunions, comdir, congrès, débats et autres aspects techniques que j'ai partagés avec les membres de ces initiatives.

Je remercie également M. André Aoun, M. Abdelmalek Benzekri, M. François Barrère, M. Thierry Desprats et M. Michel Galindo pour leur sympathie de tous les jours. Merci aussi à tous les docteurs, doctorants et stagiaires de l'équipe pour leur bonne humeur et leur soutien, et plus particulièrement à Audrey, Pierre, Anhtu et Olivier. Je pense aussi à mon loulou londonien pour ses conseils scientifiques, mais aussi et surtout pour les autres moments passés ensemble, que ce soit autour d'une bière ou sur le monocycle. Vivement qu'il revienne !

Maryse Cailloux a toujours été présente pour organiser les (nombreuses) missions et faciliter les conditions de mon travail, je lui adresse mes remerciements les plus chaleureux. Aussi je remercie vivement Martine Labruyère pour ses encouragements permanents, sa disponibilité et sa gentillesse.

Enfin, je ne pourrais terminer cette page sans remercier ma famille et tout particulièrement ma mère qui m'a toujours accordé la plus grande confiance, soutenu et encouragé tout au long de ces nombreuses années d'étude et sans qui je ne serais jamais arrivé là.

Table des matières

Résumé **i**

Abstract **ii**

Remerciements **iii**

Table des matières **vi**

Table des figures **xiii**

Liste des tableaux **xvii**

Introduction **1**

Chapitre 1 Introduction générale **3**

1.1 Cadre scientifique 3

1.2 Contexte 4

1.3 Problématiques 5

1.4 Organisation du mémoire 6

1.4.1 Partie I : Etat de l'art 6

1.4.2 Partie II : Virtualisation et supervision de ressources pédagogiques 7

1.4.3 Partie III : Expérimentations 8

I Etat de l'art **11**

Chapitre 2 Evolution des architectures des EIAH **13**

2.1 Architecture centralisée 13

2.2 Architecture 3 tiers propriétaire 14

2.3 Architecture 3 tiers normalisée 15

2.4 Vers l'interopérabilité des composants de l'EIAH 16

Chapitre 3 Composants et standards associés **21**

3.1 Introduction 21

3.2 Les objets pédagogiques 22

3.2.1 Origine et métaphore 22

3.2.2 Définition 23

3.2.3 Granularité des objets pédagogiques 23

3.2.4 Les métadonnées 25

3.3 Les viviers de connaissance 27

3.3.1 Distinction entre viviers de documents et viviers de références 28

3.3.1.1 Les viviers de documents 28

3.3.1.2 Les viviers de références 28

3.3.2 La spécification IMS Digital Repositories Interoperability 30

3.3.3 La spécification Simple Query Interface 32

3.3.4 Déploiement des services de traitement 33

3.3.5 Synthèse 35

3.4 Les plates-formes de gestion d'apprentissage 36

3.4.1 Définitions et outils intégrés 36

3.4.1.1 Définitions 36

3.4.1.2 Outils intégrés 37
3.4.2 L'architecture LTSA 39
3.4.3 SCORM 41
3.4.3.1 Le modèle d'agrégation de contenu 42
3.4.3.2 L'environnement d'exécution 43
3.4.4 IMS Learning Design 44
3.4.4.1 La Spécification 44
3.4.4.2 L'unité d'apprentissage 45
3.4.4.3 La structure d'implémentation 45
3.4.5 Synthèse 48
3.5 Conclusion du chapitre 48

Chapitre 4 Les informations disponibles pour l'observation des ressources pédagogiques 51

4.1 Introduction 51
4.2 Informations communes aux systèmes d'apprentissage 52
4.3 Informations spécifiques aux viviers de connaissance 53
4.4 Informations spécifiques aux plates-formes d'apprentissage 54
4.5 Synthèse 54

Chapitre 5 Les verrous à lever 55

5.1 Intégrer les solutions hétérogènes existantes 55
5.2 Fournir un mécanisme d'authentification 56
5.3 Offrir des services d'exploitation 56
5.4 Superviser les interactions entre composants de l'EIAH 58

II Virtualisation de ressources pédagogiques et supervision de l'EIAH 61

Chapitre 6 Virtualisation de ressources pédagogiques 63

6.1 Architecture de virtualisation 63

6.2 Couche de Fédération 64

6.2.1 Service de recherche des objets pédagogiques 66

6.2.1.1 Obtention d'un objet pédagogique 66

6.2.1.2 Scénario de recherche d'un objet pédagogique 66

6.2.1.3 Spécifications des méthodes du service de recherche 68

6.2.2 Services de téléchargement et de consultation 71

6.2.3 Services d'importation 72

6.2.4 Service d'indexation 74

6.2.5 Service de génération automatique de métadonnées 75

6.2.6 Service d'authentification 78

6.3 Couche de Mutualisation 81

6.3.1 Lien entre les plates-formes de gestion d'apprentissage et la couche de Fédération 81

6.3.2 Fournisseur d'informations 82

6.4 Synthèse 82

Chapitre 7 Supervision de l'exploitation des ressources 85

7.1 Objectifs 86

7.2 Le modèle commun de l'information 87

7.2.1 Les principaux éléments de la modélisation CIM 87

7.2.2 Les niveaux d'abstraction 87

7.2.3 Le protocole de transport 89

7.2.4 Les raisons de notre choix 90

7.3 Modélisation de l'EIAH 91

7.3.1 Vue générale 91

7.3.2 Modélisation de l'environnement d'apprentissage 93

7.3.2.1 Les objets gérés 93

7.3.2.2 Les associations 97
7.3.3 Modélisation des utilisateurs 98
7.3.4 Relations entre l'EIAH et les utilisateurs 99
7.4 Architecture globale de notre EIAH 100
7.4.1 L'architecture de gestion WBEM 100
7.4.2 Un CIM *Object Provider* pour l'EIAH 102
7.4.2.1 L'interface dédiée à l'EIAH 102
7.4.2.2 L'interface de communication avec le CIM *Object Manager* . . . 104
7.4.3 Intégration du cadre de gestion WBEM au sein de notre architecture . . . 106
7.4.3.1 Architecture globale de notre EIAH 106
7.4.3.2 Un agent dédié à l'EIAH 107
7.4.3.3 Le CIM *Object Manager* et le référentiel CIM 109
7.4.3.4 L'application de gestion 109
7.5 Synthèse 110

III Expérimentations 113

Chapitre 8 Développement de la couche de Virtualisation des ressources pédagogiques 115

8.1 Fédération des viviers ARIADNE, MERLOT, EdNA, LRC 115
8.1.1 Hétérogénéité des métadonnées 116
8.1.2 Accès aux ressources pédagogiques 117
8.2 Téléchargement et importation de ressources externes 118
8.3 Indexation d'une nouvelle ressource 119
8.4 Extraction et génération automatique de métadonnées LOM 119
8.4.1 Indexation d'une ressource SCORM 119
8.4.2 Indexation d'un document arbitraire 121
8.4.2.1 Métadonnées générales 121

8.4.2.2 Métadonnées sémantiques 122
8.4.2.3 Métadonnées pédagogiques 122
8.4.2.4 Métadonnées techniques 124
8.4.2.5 Synthèse 124
8.5 Déploiement de la couche de Mutualisation 125
8.5.1 INES et International E-Mi@ge 125
8.5.2 Moodle et CGIAR / IUT1A 126
8.6 Synthèse 127

Chapitre 9 Développement d'une plate-forme de supervision des ressources pédagogiques 129

9.1 Les technologies retenues 130
9.2 Scénarios de gestion 131
9.3 Usage des entités gérées 132
9.3.1 Les objets pédagogiques 132
9.3.2 Les viviers de connaissance 133
9.3.3 Les plates-formes d'apprentissage 135
9.4 Recommandations personnalisées de ressources pédagogiques 136
9.4.1 Recommandation basée sur les concepts 136
9.4.2 Recommandation basée sur les disciplines 137
9.4.3 Recommandation basée sur les labels et commentaires 138
9.4.4 Evaluation des ressources pédagogiques 139
9.5 Synthèse 139

Conclusions et perspectives de recherche 140

Chapitre 10 Conclusions et perspectives de recherche 143

10.1 La fédération des viviers de connaissance 143

10.2 La mutualisation des ressources pédagogiques 144

10.3 La supervision des systèmes et utilisateurs impliqués dans un EIAH 145

10.4 Des métriques pour optimiser l'exploitation des ressources 147

Bibliographie **147**

Glossaire **161**

Annexes **167**

Chapitre A Standardisation des métadonnées **171**

A.1 Dublin Core (DC) 171

A.2 ARIADNE Metadata Set 172

A.3 Instructional Management Systems (IMS) 173

A.4 Le standard Learning Object Metadata (LOM) 176

A.5 Représentation des standards de métadonnées 177

A.5.1 Représentation XML du standard LOM 177

A.5.2 Représentation RDF du standard LOM 179

Chapitre B Scénarios de gestion **181**

B.1 Recherche 181

B.2 Consultation 183

B.3 Téléchargement 183

B.4 Importation 183

B.5 Indexation 184

Table des figures

2.1 Architecture centralisée 14

2.2 Architecture 3 tiers propriétaire 15

2.3 Architecture 3 tiers normalisée 16

3.1 Taxonomie des contenus d'objets 24

3.2 Synthèse des standards de métadonnées [Inc03] 27

3.3 Le Knowledge Pool System de la fondation ARIADNE 29

3.4 Architecture fonctionnelle de IMS DRI [Con03a] 31

3.5 Diagramme de classes de la spécification SQI [SMA+05] 32

3.6 Diagramme de séquence d'une requête en mode asynchrone [SMA+05] 33

3.7 Normalisation des services *web* 35

3.8 Les composants de l'architecture LTSA 40

3.9 La bibliothèque du SCORM [Lea04a] 42

3.10 Le modèle d'agrégation de contenus et les méta-données correspondantes [Per03] 43

3.11 IMS Content Packaging et IMS Learning Design [KT05] 45

3.12 Niveaux A, B et C de la spécification IMS-LD [Con03d] 46

6.1 Une architecture pour la virtualisation de ressources pédagogiques 64

6.2 Les services rendus par la couche de Fédération 65

6.3 Diagramme d'activités du processus d'obtention d'objets pédagogiques [Mas05] . 67

6.4 Scénario de recherche des ressources pédagogiques 67

6.5 Scénario d'authentification 79

7.1 Les éléments de base de la modélisation CIM 88
7.2 Extrait des différents niveaux d'abstraction du modèle CIM 88
7.3 Encapsulation des informations CIM dans les messages HTTP 89
7.4 Modélisation d'un EIAH 92
7.5 Représentation des systèmes d'apprentissage 93
7.6 Représentation des ressources d'apprentissage 94
7.7 Représentation des cursus pédagogiques 96
7.8 Représentation des labels et commentaires 96
7.9 Les associations entre classes gérées de l'environnement d'apprentissage 97
7.10 Représentation des utilisateurs dans le modèle CIM 98
7.11 Les associations entre l'environnement d'apprentissage et les utilisateurs 99
7.12 Les entités de gestion dans l'architecture WBEM 101
7.13 Structure d'un *Object Manager* 102
7.14 Architecture globale de notre EIAH 107

8.1 Déploiement de la couche de Virtualisation 116
8.2 Implémentation du service de recherche en PHP 117
8.3 Interface de la couche de Fédération 118
8.4 Implémentation du service de téléchargement en PHP 119
8.5 Implémentation du service d'indexation en PHP 120
8.6 Implémentation du service d'extraction des métadonnées d'un package SCORM en PHP 120
8.7 Résultat du service d'extraction de métadonnées à partir d'un fichier *imsmanifest.xml* 121
8.8 Extrait du fichier XML pour la génération automatique de métadonnées Sémantiques 123
8.9 Extrait de la bibliothèque exploitant l'extension d'un fichier 123
8.10 Déploiement de la couche d'Intégration au sein de la plate-forme INES 125
8.11 Déploiement de la couche d'Intégration au sein de la plate-forme Moodle 126

9.1 Un cadre de supervision des ressources pédagogiques 131

9.2 Construction et envoi d'une requête au CIM *Object Manager* 133

9.3 Usage des objets pédagogiques 134

9.4 Recherche par plate-forme d'apprentissage 135

9.5 Un outil personnalisé pour la recherche d'objets pédagogiques 137

A.1 Diagramme conceptuel de l'IMS Content Packaging [Con05] 175

A.2 Les éléments du Learning Object Metadata 177

A.3 Un exemple de représentation XML du LOM 178

B.1 Création d'une instance de la classe CIM_Person 182

Liste des tableaux

6.1 Spécification de la méthode createAnonymousSession [SMA+05] 68

6.2 Spécification de la méthode setQueryLanguage [SMA+05] 69

6.3 Spécification de la méthode setResultsFormat [SMA+05] 70

6.4 Spécification de la méthode synchronousQuery [SMA+05] 70

6.5 Spécification de la méthode previewUrl 71

6.6 Spécification de la méthode downloadDocument 72

6.7 Spécification de la méthode importUrl 73

6.8 Spécification de la méthode importDocument 73

6.9 Spécification de la méthode indexLearningObject 75

6.10 Spécification de la méthode getContextKeywords 76

6.11 Spécification de la méthode getFileInfo 76

6.12 Spécification de la méthode extractScormMetadata 77

6.13 Spécification de la méthode getAuthorInfo 78

6.14 Spécification de la méthode getUserAccount 80

6.15 Les services permettant d'exploiter les ressources pédagogiques 83

7.1 Spécification de la méthode receiveData 103

7.2 Spécification de la méthode getUserAccount 104

7.3 Spécification de la méthode searchEvent 105

7.4 Spécification de la méthode getAccountInstance 106

7.5 Informations capturées par l'agent en fonction des actions réalisées par l'utilisateur 108

8.1 Récapitulatif des conventions définies 124

A.1 Le Dublin Core Metadata Set 172

Introduction

Chapitre 1

Introduction générale

Sommaire

1.1 Cadre scientifique **3**
1.2 Contexte **4**
1.3 Problématiques **5**
1.4 Organisation du mémoire **6**
1.4.1 Partie I : Etat de l'art 6
1.4.2 Partie II : Virtualisation et supervision de ressources pédagogiques . . 7
1.4.3 Partie III : Expérimentations 8

1.1 Cadre scientifique

Ce manuscrit présente la synthèse de trois ans de travail effectués dans le cadre d'une thèse. Celle-ci s'est déroulée au sein de l'équipe SIERA[1] du laboratoire IRIT[2] à Toulouse. L'IRIT est une Unité Mixte de Recherche (UMR 5505) commune à plusieurs établissements qui sont l'Université Paul Sabatier[3], le CNRS[4], l'INPT[5], et l'Université Toulouse 1 Sciences Sociales[6]. Une des priorités du laboratoire est le développement de projets transversaux ouverts aux communautés nationale et internationale, d'actions de valorisation et de diffusion de la culture scientifique et technique, de transferts de connaissances par l'organisation régulière de rencontres Recherche/Monde socio-économique et d'opérations en direction du grand public et des scolaires et enfin, de transferts de savoir-faire et de technologies par la création de laboratoires communs avec l'industrie.

Les travaux de l'équipe SIERA, dirigée par Yves Raynaud, visent le contrôle et la maîtrise

[1]Service IntEgration and netwoRk Administration (Administration de Réseaux Et Intégration de Services)
[2]Institut de Recherche en Informatique de Toulouse
[3]http://www.ups-tlse.fr
[4]Centre National de la Recherche Scientifique - http://www.cnrs.fr
[5]Institut National Polytechnique de Toulouse - http://www.inpt-toulouse.fr
[6]http://www.univ-tlse1.fr

des infrastructures et services de communication de dernières générations, mais aussi des systèmes et applications complexes dynamiquement agrégés, distribués et trans-organisationnels. Ainsi, ces travaux s'attachent à la définition et à l'évaluation de nouveaux paradigmes de gestion et apportent à la communauté des architectures, des plates-formes, des outils ou encore des contributions à la normalisation. Dans ce contexte, plusieurs thématiques particulières font l'objet d'études et de propositions : (1) une démarche de conception d'applications de gestion dégageant une double indépendance, l'une vis-à-vis des domaines spécifiques à gérer et l'autre vis-à-vis des plates-formes de gestion, (2) l'évaluation du potentiel des SMA[7] à répondre aux exigences de la gestion coopérative et de la gestion autonome, (3) la gestion de la sécurité des flux d'information d'une organisation virtuelle à travers le raffinement de politiques pour concevoir et déployer des politiques inter-domaines, et la définition d'un environnement de gestion qui permet de contrôler les politiques de sécurité et de valider formellement les règles exécutables de sécurité implantées liées aux mécanismes et protocoles de sécurité appropriés, et (4) la distribution et la gestion des environnements de formation en ligne qui constituent de nos jours des systèmes complexes d'exploitation d'objets pédagogiques à grande échelle. Ce dernier thème constitue le sujet de thèse sur lequel j'ai travaillé. Je présente dans ce document les travaux effectués ainsi que les résultats obtenus.

1.2 Contexte

La dernière décennie a connu un véritable engouement en matière d'enseignement à distance. A l'origine dénommé Formation à Distance puis Formation Ouverte et A Distance, le terme e-formation (*e-learning* pour nos voisins anglo-saxons) est désormais employé pour désigner un dispositif de formation utilisant l'Internet ou les Intranets comme vecteurs de diffusion de connaissances et de formations. L'e-formation apporte de multiples avantages liés à ses modes de production, de présentation et de diffusion de contenu, mais il permet également la réduction des frais indirects de formation surtout dans les organisations privées, une individualisation de l'apprentissage en rendant disponible un catalogue de formation à l'apprenant, et un déploiement rapide de la formation sur une grande échelle.

Cependant, un environnement d'e-formation est bien plus complexe qu'un système de mise en ligne de documents pédagogiques par le biais des technologies *web*. En effet, celui-ci regroupe plusieurs fonctionnalités inhérentes à la gestion des utilisateurs, formations et cursus. L'accès à des bibliothèques de ressources électroniques, la production de contenus pédagogiques par composition ou agrégation de modules de natures différentes portant sur les connaissances à

[7]Service Monitoring Agent

diffuser, la mise à disposition d'un espace de travail et des outils de communication permettent à tout usager d'améliorer le processus d'acquisition ou de délivrance de compétences. Le terme environnement informatique pour l'apprentissage humain (EIAH) est alors apparu au début des années 2000 pour caractériser un "environnement informatique conçu dans le but de favoriser l'apprentissage humain, c'est-à-dire la construction de connaissances chez un apprenant" [Tch02]. Le terme EIAH sera employé dans le reste du manuscrit pour dénoter un environnement d'apprentissage (un "EIAH") mais également le domaine de recherche correspondant (l'"EIAH").

1.3 Problématiques

Même si l'e-formation est de plus en plus répandue aussi bien dans les institutions publiques que privées, elle est loin de représenter une méthode d'apprentissage utilisée par tous. Un moyen de contribuer à son développement est de mettre à disposition des utilisateurs une masse importante d'objets pédagogiques de natures variées tant au niveau des domaines d'apprentissage visés que du type de supports numériques utilisés, ainsi que de la langue dans laquelle ils sont produits. Dans cet objectif, l'émergence de standards laisse entrevoir la possibilité d'atteindre des solutions de fédération nous amenant à partager et à réutiliser les ressources pédagogiques.

Les travaux présentés ici visent la mise en oeuvre de la virtualisation des objets pédagogiques contribuant à l'utilisation de masse de l'e-formation. D'après [BGC02], "Virtualiser, c'est donner au système d'exploitation ou à l'application une vision logique de l'espace de stockage, complètement dissociée de la réalité physique des systèmes matériels sur lesquels sont effectivement stockées les données". Dans le cas d'un EIAH, elle doit mettre à disposition des enseignants et concepteurs de cursus un nombre considérable de ressources pédagogiques localisées sur des systèmes distincts tout en leur facilitant la recherche, l'acquisition et la mise à disposition de ce matériel pédagogique.

Toutefois, la virtualisation des ressources soulève une autre problématique liée à la supervision de leur exploitation par les multiples systèmes informatiques impliqués dans le processus de virtualisation. En effet, ce processus engendre une difficulté pour les concepteurs de cursus qui doivent retrouver du matériel pédagogique pertinent à leur contexte parmi un nombre considérable de ressources. Alors la mise en oeuvre d'un mécanisme capable d'observer et d'enregistrer l'ensemble des interactions intervenant entre les systèmes et les utilisateurs doit permettre de proposer des méthodes de recherche plus avancées, et de sélectionner le matériel proposé aux concepteurs de cursus en fonction de la situation d'apprentissage.

1.4 Organisation du mémoire

Afin de présenter les différents travaux que nous avons conduits dans le cadre de la virtualisation et de la supervision des ressources pédagogiques, nous avons organisé ce manuscrit en trois parties principales qui sont : (1) l'état de l'art des différents composants mis en jeu dans un EIAH et des normes qui leurs sont associées, afin de mettre en avant les barrières à franchir pour atteindre la virtualisation des ressources pédagogiques, (2) les propositions de solutions que nous avons faites pour faciliter d'une part le partage et la réutilisation des ressources d'apprentissage, et d'autre part la supervision de ces mêmes ressources, et (3) la validation de ces propositions à travers leur développement et déploiement au sein de systèmes d'e-formation existants. Enfin, une conclusion et un ensemble de perspectives clôturent ce document.

1.4.1 Partie I : Etat de l'art

Cette partie, constituée de quatre chapitres, est une synthèse des différents domaines de recherche qui ont constitué le cadre de notre travail.

Evolution des architectures des EIAH

Le chapitre 2 évoque les trois fonctionnalités charnières qui ont été mises en avant pour la mise en oeuvre d'un EIAH, et présente les différentes mutations des architectures des EIAH qui ont assuré la réalisation de ces fonctionnalités. Nous discernons trois principaux types d'architecture qui sont l'architecture centralisée, l'architecture trois tiers propriétaire, et l'architecture trois tiers normalisée. Pour chacune d'entre elles, nous dégageons les difficultés et obstacles qui contrarient le partage et la réutilisation des ressources pédagogiques. La dernière partie de ce chapitre exprime les besoins qui doivent être satisfaits pour atteindre la virtualisation des ressources pédagogiques.

Composants et standards associés

Le chapitre 3 expose les trois éléments d'un EIAH qui constituent le coeur de nos travaux : les ressources pédagogiques, les systèmes de stockage de ces ressources, et les plates-formes d'apprentissage destinées à leur exploitation. Pour le premier élément, nous examinons les principales définitions proposées par des organismes nationaux ou internationaux dans le but de déterminer le sens d'une ressource d'apprentissage dans le reste du manuscrit ; aussi nous évoquons les discussions liées à la granularité des ressources pédagogiques avant de dresser une synthèse des différents standards et normes ayant vu le jour depuis la fin des années 1990 pour décrire ces

ressources. Nous distinguons ensuite deux types de systèmes de stockage avant de présenter des spécifications visant l'interopérabilité de tels systèmes, ainsi que leurs services associés qui s'appuient sur la technologie des services *web* en plein essor depuis trois ans. Nous rapportons dans la dernière section de ce chapitre quelques définitions d'une plate-forme d'apprentissage, et énumérons les principaux outils intégrés dans la majorité de ces systèmes. Nous exposons aussi les normes et standards qui sont pris en compte par ces entités et qui sont destinés à favoriser et améliorer le processus d'apprentissage en ligne.

Les informations disponibles pour l'observation des ressources pédagogiques

Le chapitre 4 présente les informations mises à disposition par la majorité des systèmes actuels pour superviser les ressources renfermées dans ces systèmes. Nous détaillons en particulier les données relatives à l'utilisation des ressources pédagogiques tant au niveau des systèmes de stockage qu'à celui des plates-formes d'apprentissage. Nous dégageons également certaines limites qui empêchent la mise en oeuvre d'une supervision efficace de l'exploitation des ressources pédagogiques.

Les verrous à lever

Le chapitre 5, dernier chapitre de l'état de l'art, met en évidence les lacunes et barrières des différentes notions présentées dans cette partie afin de déterminer et de définir les tâches à accomplir pour atteindre nos objectifs de virtualisation et de supervision des ressources d'apprentissage. Nous exprimons dans un premier temps la nécessité de proposer une solution générique qui intègre l'hétérogénéité des éléments d'un EIAH afin de combler les faiblesses des solutions existantes. Dans un deuxième temps, nous identifions les services requis pour satisfaire nos objectifs de généricité et d'indépendance vis-à-vis des systèmes communiquant entre eux. Enfin, nous démontrons le besoin de mettre en oeuvre une alternative aux techniques d'observation existantes qui soit capable non seulement de superviser les interactions survenant au sein d'un ensemble de systèmes, c'est-à-dire dans un EIAH constitué de plusieurs composants, mais aussi d'exposer de manière unifiée et homogène les informations issues de ces systèmes hétérogènes.

1.4.2 Partie II : Virtualisation et supervision de ressources pédagogiques

Cette partie présente les travaux de conception d'une architecture de virtualisation des ressources pédagogiques, ainsi qu'une approche orientée modèle permettant la supervision de leur exploitation par les entités impliquées dans un environnement d'apprentissage.

Virtualisation de ressources pédagogiques

Le chapitre 6 définit une architecture de virtualisation structurée en couches pour le support de notre EIAH qui prend en compte les normes et standards en cours d'élaboration dans le domaine de l'e-formation et présentés dans la partie I. Nous nous attachons à détailler les deux niveaux qui constituent la couche additionnelle que nous avons introduite dans l'architecture trois tiers normalisée et qui met en œuvre les fonctionnalités et services nécessaires à la virtualisation. La couche de Fédération se base sur une architecture orientée service (SOA[8]) pour offrir aux utilisateurs un grand nombre de ressources pédagogiques localisées dans des systèmes distincts et hétérogènes, alors que la couche de Mutualisation vise à rendre l'accès à la couche de Fédération le plus facile et transparent possible aux utilisateurs et systèmes d'un environnement informatique pour l'apprentissage humain.

Supervision de l'exploitation des ressources : une approche conduite par les modèles

Le chapitre 7 s'intéresse à un modèle d'information pour la représentation des différents utilisateurs et entités mis en jeu dans un EIAH. Ce modèle, organisé autour d'un métamodèle existant issu du monde de la gestion de réseau, offre la possibilité de superviser les interactions intervenant dans notre EIAH, mais démontre également comment il peut être étendu pour satisfaire les caractéristiques d'autres outils et systèmes utiles à l'apprentissage en ligne. Cette approche permet de recueillir et d'exploiter des informations statistiques comme le nombre de consultations ou de téléchargements d'une ressource pédagogique particulière, ou des informations relatives aux utilisateurs ayant manipulé les ressources d'apprentissage disponibles à travers la couche de Virtualisation. La dernière section du chapitre enrichit l'architecture de virtualisation présentée dans le chapitre précédent pour intégrer les composants de supervision nécessaires à l'observation de l'exploitation des ressources par les différents composants qui constituent notre environnement d'apprentissage.

1.4.3 Partie III : Expérimentations

Cette partie est relative à nos travaux de déploiement et d'implémentation de nos propositions au sein de projets d'e-formation nationaux et internationaux existants.

Développement de la couche de Virtualisation des ressources pédagogiques

La chapitre 8 concerne une application s'inscrivant dans l'architecture de virtualisation

[8]Service Oriented Architecture

qui permet la fédération d'un ensemble de ressources pédagogiques renfermées dans des systèmes de stockage distincts. Toutes les ressources accessibles à travers la fédération sont contextualisées pour l'EIAH par l'usage de descripteurs et directement utilisables par des auteurs, étudiants, enseignants, mais aussi et surtout par les systèmes d'apprentissage. Notre solution est transparente aux utilisateurs : bien que les systèmes demeurent des entités indépendantes, ils constituent une entité unique capable d'être interrogée et utilisée simultanément. Ces travaux ont été déployés dans plusieurs universités françaises et étrangères, mais également dans une société privée internationale de recherche agricole visant la réduction de la pauvreté et promouvant le développement durable. Notre solution, développée sous la forme d'une API[9] et dans le cadre de la fondation ARIADNE[10] à laquelle nous sommes rattachés, est disponible en libre téléchargement sur le site de Source Forge[11].

Développement d'une plate-forme de supervision des ressources pédagogiques

Le chapitre 9 s'intéresse au développement d'une application de supervision basée sur les technologies *web* qui exploite notre modèle d'information et qui offre aux utilisateurs différents services facilitant la recherche de ressources pédagogiques pertinentes à leur contexte d'apprentissage. Outre des informations statistiques relatives aux différentes entités et composants de notre EIAH, nous proposons aux usagers un service personnalisé permettant de retrouver des ressources correspondant à leurs concepts et notions préférés, et qui ne leur impose pas de renseigner un formulaire électronique précisant le sujet auquel ils s'intéressent. Nous proposons également aux utilisateurs un moyen de classifier de façon personnalisée les ressources pédagogiques en leur associant des labels ou commentaires.

Pour terminer ce manuscrit, nous présentons les conclusions que nous avons tirées du travail effectué durant ces trois années de thèse et nous proposons quelques pistes pour la suite.

[9] Application Program Interface

[10] Alliance of Remote Instructional Authoring and Distribution Networks for Europe - http://www.ariadne-eu.org

[11] http://sourceforge.net/project/showfiles.php?group_id=111465

Première partie

Etat de l'art

Chapitre 2

Evolution des architectures des EIAH

Sommaire

2.1 Architecture centralisée **13**
2.2 Architecture 3 tiers propriétaire **14**
2.3 Architecture 3 tiers normalisée **15**
2.4 Vers l'interopérabilité des composants de l'EIAH **16**

Lors de l'apparition de la Formation Ouverte et A Distance, trois fonctionnalités vitales ont été mises en avant pour la mise en œuvre d'un EIAH [Paq00] :

- un moyen de stockage des ressources pédagogiques produites,
- un ensemble de services offrant des opérations de traitement sur les ressources pédagogiques renfermées dans ces systèmes de stockage,
- et une interface dédiée à l'apprentissage à travers des fonctionnalités spécialisées qui exploitent les ressources d'apprentissage d'une part, et qui proposent des fonctionnalités de gestion inter-acteurs d'autre part.

Nous exposons dans ce chapitre les trois principales évolutions des architectures des EIAH au cours de ces dernières années qui ont mis en oeuvre l'ensemble des fonctionnalités précitées, avant de mettre en avant les besoins à satisfaire pour atteindre l'interopérabilité des composants d'un EIAH.

2.1 Architecture centralisée

Les premiers systèmes proposaient une application centralisée réalisant l'ensemble de ces fonctionnalités[1] (cf. figure 2.1) [Ell00] [SSS+99]. Un composant du dispositif était dédié à la ges-

[1] http://gi.insa-lyon.fr/insav/public/village/Apropos/page1.html

tion des ressources d'apprentissage, un autre aux services permettant d'ajouter, de modifier ou de supprimer ces ressources, et un autre élément du système proposait un ensemble de services dédiés à l'apprentissage en ligne tels que la gestion des rôles des utilisateurs au sein du système, la création de formations/cursus, des outils de communication, des devoirs, etc. Ces dispositifs d'apprentissage peuvent être comparés aux plates-formes d'apprentissage d'aujourd'hui que nous présentons dans la section 3.4, même si elles sont désormais beaucoup plus évoluées. Ces environnements présentent l'avantage de proposer une interface unique qui facilite leur prise en main par les acteurs de la FOAD, mais ce type d'architecture cloisonnée n'autorise aucun partage ou réutilisation des outils et matériels pédagogiques disponibles.

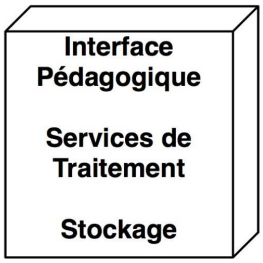


FIG. 2.1 – Architecture centralisée

2.2 Architecture 3 tiers propriétaire

Les acteurs de la FOAD se sont rapidement rendus compte de la nécessité de dissocier ces fonctionnalités, c'est-à-dire de décomposer un EIAH en un ensemble complémentaire de composants, chacun se focalisant sur une fonctionnalité spécifique. Ces ensembles de briques pourraient alors être réutilisées indépendamment les unes des autres (cf. figure 2.2).

Un des premiers EIAH à s'appuyer sur cette architecture fût le système ARIADNE [Duv04] [FME+02] qui reposait sur un ensemble d'outils permettant l'indexation, le stockage, la diffusion, et l'exploitation des différents documents pédagogiques. Des outils auteurs ont également été développés pour aider les ingénieurs pédagogiques dans la création de ces documents. Depuis 2003, la fondation ARIADNE issue des projets européens ARIADNE I [FWD97a] et ARIADNE II [FWD97b] se concentre sur sa base de données renfermant les multiples objets pédagogiques ainsi que sur l'outil de recherche et d'indexation de ces objets. Son dispositif offrant des services d'apprentissage a été complètement abandonné, tandis que des outils auteur comme GenDoc [Dav03] ou Phoenix [FMMW05] sont toujours utilisés et améliorés pour la production de nouvelles ressources.

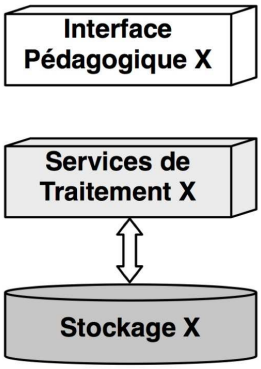


Fig. 2.2 – Architecture 3 tiers propriétaire

Les interfaces pédagogiques proposaient des formats de cursus qui leur étaient spécifiques ou ne pouvaient déployer que des ressources respectant une typologie bien définie ; les services de traitement ainsi que le stockage des ressources présentaient des spécifications particulières qui inhibaient toute communication avec d'autres environnements.
Les systèmes conçus au sein d'une même institution étaient destinés aux seuls membres de l'organisation, et mettaient en œuvre des solutions propriétaires qui n'avaient pas pour vocation de s'ouvrir vers les autres EIAH existants [FWFD96].

2.3 Architecture 3 tiers normalisée

Dans cet objectif d'ouverture et de collaboration, de nombreuses initiatives de normalisation sont apparues [VAB03] [VBD04], qui ont permis à des systèmes d'intégrer d'autres environnements existants, mais sous la condition qu'ils respectent les mêmes spécifications (cf. figure 2.3) [PCMG02].

Des recommandations ont vu le jour afin de permettre aux interfaces pédagogiques de déployer des ressources communes. Au niveau des services de traitement, des initiatives sont apparues (certaines sont toujours en cours de spécification) afin de promouvoir l'accessibilité des systèmes de stockage. Le projet OAI[2], par exemple, a pour objectif de fournir un ensemble de protocoles communs liés aux processus de recherche d'informations. Ce projet définit des spécifications qui sont nécessaires pour que les archives deviennent interopérables, c'est-à-dire pour qu'elles puissent être interrogées simultanément, quelque soit leur localisation. Open Archives Initiative permet de rechercher des contenus dans différentes bases d'informations telles

[2]Open Archives Initiative - http://www.openarchives.org

que Google[3] ou CiteSeer[4], mais ces systèmes ont dû implémenter la spécification OAI avant que cette interopérabilité soit fonctionnelle.

Dans le même ordre d'idée, le modèle pair à pair[5] (P2P) permet aux utilisateurs d'échanger des ressources, quelle que soit leur nature. Ce modèle connaît un très large succès depuis la fin des années 90, époque à laquelle Napster, une application P2P de partage de fichiers où chaque pair jouait le rôle de client et de serveur, permit à des millions d'usagers connectés à l'Internet de télécharger et partager librement des fichiers multimédias. Suite à l'arrêt du service Napster dû à un procès initié par les grands noms de l'industrie du disque, un autre groupe de développeurs se lança dans le projet Gnutella[6] et montra qu'il est possible de proposer aux usagers un service qui ne repose sur aucune infrastructure physique concrète mais sur un simple logiciel distribué au sein d'une communauté d'utilisateurs.
Dans le domaine de l'éducation, le projet Edutella [NWQ+02], fortement inspiré de l'initiative précitée, offre un échange P2P de ressources pédagogiques. Plus récemment, le projet LionShare [HSH+03] s'inspire également de ce modèle P2P et propose en libre téléchargement son logiciel spécifique.

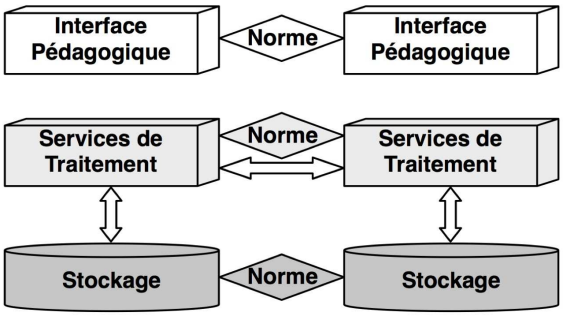


Fig. 2.3 – Architecture 3 tiers normalisée

2.4 Vers l'interopérabilité des composants de l'EIAH

L'architecture 3 tiers normalisée introduit une certaine forme de communication horizontale entre composants de même niveau d'un EIAH.
Même si les interfaces pédagogiques sont spécifiques à un produit particulier et qu'elles fournissent des services d'apprentissage qui leur sont propres, certaines normes leur permettent d'exploiter des ressources d'apprentissage communes de façon standardisée et homogène. Cependant,

[3]http://google.com
[4]http://citeseer.ist.psu.edu/
[5]ou Poste à Poste
[6]http://www.gnutella.com/

ces systèmes ne sont pas capables de communiquer entre eux pour échanger directement leur matériel pédagogique.
Certains services de traitement associés au stockage des objets pédagogiques et de leurs descripteurs ont opté pour des recommandations communes qui leur permettent de partager le matériel d'apprentissage qu'ils renferment, mais la plupart d'entre eux présentent encore des fonctionnalités et spécifications hétérogènes. En revanche, la description et la représentation des ressources présentent une plus forte maturité se traduisant par l'adoption de standards déjà bien établis par une majorité des systèmes de stockage.

En vue d'une exploitation à plus large échelle des objets pédagogiques, la communication verticale entre systèmes impliqués dans l'EIAH doit être mise en oeuvre. Le partage, la réutilisation, ainsi que l'exploitation des ressources impliquent que tous ces systèmes, qu'il s'agisse des systèmes de stockage ou des interfaces pédagogiques, soient en mesure de communiquer entre eux, c'est-à-dire d'être interopérables. L'ATICA[7] définit l'interopérabilité comme la "faculté que possèdent des ensembles informatiques hétérogènes de fonctionner conjointement et de donner accès à leurs ressources de façon réciproque" [OP01].

Alors pour permettre une communication totale (horizontale et verticale) entre composants de l'EIAH et tendre vers la quatrième génération des EIAH, nous sommes confrontés à deux problématiques complémentaires qui, d'après la définition donnée en section 1.3, peuvent être assimilées à la virtualisation des objets pédagogiques : d'une part assurer la fédération des systèmes de stockage hétérogènes existants et d'autre part, mettre en place une communication entre systèmes de stockage et interfaces pédagogiques dans le but de rendre facilement accessibles et exploitables les multiples ressources d'apprentissage fédérées.

La fédération des systèmes de stockage nécessite de répondre à des besoins variés qui sont :

- La description standardisée et adaptée à l'apprentissage en ligne des ressources pédagogiques qu'ils renferment dans le but de proposer aux concepteurs de cursus des modes de classification communs de qualité. Cela faciliterait également le travail des indexeurs de ressources pédagogiques qui n'auraient plus à s'approprier différentes méthodologies de catégorisation.
- L'utilisation de formats standardisés pour la représentation de ces descripteurs. Les outils informatiques recherchant des ressources au sein des systèmes de stockage doivent obtenir les résultats sous la forme d'une structure de données homogène et comprise par une large variété de systèmes.

[7] Agence pour les Technologies de l'Information et de la Communication dans l'Administration

- Prendre en compte les différentes spécifications des services de traitement existants et hétérogènes ; même si certains services ont adopté des recommandations communes, ils présentent pour la plupart des spécifications qui leur sont propres. Dans le cadre d'un EIAH déployé à grande échelle, ces services doivent également être facilement accessibles à travers une architecture Réseau fortement distribuée.

Des exigences supplémentaires doivent être satisfaites pour assurer la communication entre interfaces pédagogiques et systèmes de stockage :

- L'ouverture des interfaces pédagogiques vers les autres composants d'un EIAH, et en particulier vers les systèmes de stockage.
- La prise en compte et l'adoption de langages et de protocoles communs largement répandus dans l'EIAH aujourd'hui afin que l'interopérabilité ne soit pas dépendante d'une plate-forme ou technologie particulière.
- Etre capable d'authentifier les utilisateurs et de connaître leurs rôles et privilèges vis-à-vis des différents composants de l'EIAH.
- Mettre à disposition des utilisateurs un ensemble de fonctionnalités qui leur permet d'interagir de façon transparente avec les services de traitement associés aux systèmes de stockage.

Néanmoins, la virtualisation des ressources pédagogiques que nous souhaitons mettre en oeuvre ne doit pas devenir un inconvénient pour les concepteurs de cursus pédagogiques et enseignants désireux de retrouver du matériel d'apprentissage pertinent pour leur contexte. La fédération d'une multitude de systèmes de stockage met à disposition des utilisateurs un nombre considérable de ressources liées à de multiples disciplines et produites en différentes langues, mais un éducateur risque de se décourager devant l'étendue des ressources qui lui sont offertes. Nous devons donc proposer un moyen de raffiner la sélection des ressources présentées à ces utilisateurs afin de leur exposer les ressources les plus adaptées à leur contexte d'apprentissage.

L'affinement du matériel proposé aux éducateurs implique la mise en place de mécanismes de supervision dédiés à la collecte et aux traitements d'informations relatives à l'usage et à l'exploitation des ressources pédagogiques par les divers composants et utilisateurs de l'EIAH. Nous devons être en mesure de proposer aux éducateurs une recherche de ressources d'apprentissage basée sur des critères différents et plus évolués que ceux suggérés jusqu'à présent et qui se limitent, le plus souvent, à la correspondance entre des termes déterminés par l'utilisateur et les descripteurs des ressources. Les modes de recherche et de sélection que nous voulons mettre en oeuvre doivent être basés sur des données statistiques telles que :

- Le nombre de consultations des descripteurs d'une ressource.
- Le nombre de téléchargements d'une ressource.
- Le nombre d'intégrations d'une ressource au sein de cursus pédagogiques.

D'autres informations comme la discipline ou les concepts liés à un cursus pédagogique devront également être exploitées afin de proposer les ressources les plus pertinentes pour un contexte d'apprentissage.

Chapitre 3

Composants et standards associés

Sommaire

3.1 Introduction **21**
3.2 Les objets pédagogiques **22**
3.2.1 Origine et métaphore 22
3.2.2 Définition 23
3.2.3 Granularité des objets pédagogiques 23
3.2.4 Les métadonnées 25
3.3 Les viviers de connaissance **27**
3.3.1 Distinction entre viviers de documents et viviers de références 28
3.3.2 La spécification IMS Digital Repositories Interoperability 30
3.3.3 La spécification Simple Query Interface 32
3.3.4 Déploiement des services de traitement 33
3.3.5 Synthèse 35
3.4 Les plates-formes de gestion d'apprentissage **36**
3.4.1 Définitions et outils intégrés 36
3.4.2 L'architecture LTSA 39
3.4.3 SCORM 41
3.4.4 IMS Learning Design 44
3.4.5 Synthèse 48
3.5 Conclusion du chapitre **48**

3.1 Introduction

Nous avons introduit dans le chapitre précédent deux composants majeurs d'un environnement d'apprentissage qui sont les systèmes de stockage des ressources, aussi appelés viviers de connaissance, et les interfaces pédagogiques ou plates-formes d'apprentissage. Ces deux entités, qui sont basées sur la manipulation de ressources pédagogiques, proposent toutefois des services complémentaires : la première permet de mettre à disposition des concepteurs de cursus un grand nombre de ressources, alors que la seconde offre des services typiquement liés à l'apprentissage en ligne.

Dans ce chapitre nous détaillons ces deux composants en prenant soin de définir en premier lieu les ressources d'apprentissage sur lesquelles ces systèmes s'appuient pour proposer leurs services. Un des objectifs de la normalisation consistant à améliorer la qualité des flux de données entre les composants liés à un environnement d'apprentissage pour faciliter et favoriser les échanges et travaux collaboratifs, nous présentons les principales initiatives de standardisation qui ont vu le jour ces dernières années que ce soit au niveau des ressources, de leur stockage, ou des services qui permettent de les exploiter. Nous serons alors en mesure de déterminer les solutions apportées par ces standards pour permettre une communication verticale entre ces composants et atteindre la virtualisation des ressources pédagogiques.

3.2 Les objets pédagogiques

3.2.1 Origine et métaphore

Les premiers travaux théoriques sur le développement et l'assemblage de ressources numériques autonomes dans un contexte d'apprentissage ont été réalisés par David Merrill et ses collègues dans le cadre du projet TICCIT[1] dans les années 70. Leur théorie était fondée sur la définition et l'utilisation d'objets alors nommés « knowledge objects », soit objets de connaissance.

L'origine du terme *Learning Object* est, quant à elle, attribuée à Wayne Hodgins, directeur du Worldwide Learning Strategies Autodesk[2]. Il eut l'intuition d'un environnement d'apprentissage construit sur la base d'éléments autonomes en 1992, alors qu'il regardait des enfants s'amuser avec des blocs de Lego™. Suite à sa présentation dans différents forums, plusieurs groupes s'y intéressèrent et commencèrent à l'explorer, et ce aussi bien aux Etats-Unis qu'en Europe.

En 1995, la société Oracle™ initia un groupe dont le mandat était de développer le Oracle Learning Application (OLA), une application intégrée de développement et de gestion pour le domaine de l'e-formation. Le développement de cette application fondée sur une approche par objets pédagogiques fut abandonné par l'entreprise deux ans plus tard.

L'idée de créer des *Learning Objects* (LO), ou "Objets Pédagogiques", est donc un besoin récent. Ces LO sont actuellement au cœur de nombreux travaux tant dans les institutions d'enseignement ou les sociétés de formation que dans les organismes de standardisation. L'idée est de définir des composants ou briques pédagogiques afin de faciliter leur partage et leur réutilisation.

[1] Time-Shared, Interactive, Computer-Controlled Educational Television
[2] http://www.autodesk.com

3.2.2 Définition

Le groupe de travail IEEE[3] LOM[4] P1484.12 donne deux définitions des objets pédagogiques. La première définition se trouve dans la version finale décrivant le LOM : « un objet pédagogique est défini comme toute entité, numérique ou non, qui peut être utilisée pour l'enseignement ou l'apprentissage ». Si on se réferre à cette définition, tout est objet pédagogique. La seconde définition se situe dans la page de présentation du standard LOM, il est écrit : « un objet pédagogique est défini comme toute entité numérique ou non qui peut être utilisée, réutilisée ou référencée pendant des activités d'apprentissage assistées par ordinateur (enseignement intelligent assisté par ordinateur, environnements d'enseignement interactifs, système d'enseignement à distance, environnement d'apprentissage collaboratif) ». Cette dernière définition limite les objets pédagogiques aux entités utilisées dans le cadre des systèmes d'apprentissage assistés par ordinateur, c'est-à-dire aux documents numériques. Ainsi, des documents ou composants partagés entre enseignants et utilisés lors d'un enseignement traditionnel (ou présentiel) ne seraient pas des objets pédagogiques. Or, ces différents documents ou composants doivent pouvoir être réutilisés au même titre que les autres.

L'AUF[5] utilise, quant à elle, plutôt le terme d'objet d'apprentissage et le définit comme « un bloc de contenus qui focalise sur un objectif d'apprentissage spécifique. Ces objets d'apprentissage peuvent être constitués d'un ou de plusieurs composants ou blocs d'informations incluant du texte, des images, de la vidéo, des sons, des animations, etc. ».

Dans la pratique, un objet pédagogique peut correspondre à une page *web*, une image, une simulation, un test ou à tout autre type d'élément intervenant dans l'apprentissage. Les objets pédagogiques ne se limitent pas à des cours ou contenus de formation proprement dits. Un objet pédagogique peut aussi faire référence à une procédure ou des directives pour aider l'apprenant dans son cheminement. Dans la suite du document, nous utiliserons indifféremment les termes objets pédagogiques, objet d'apprentissage, ressource pédagogique ou ressource d'apprentissage pour désigner un document qui peut être utilisé ou réutilisé dans un EIAH.

3.2.3 Granularité des objets pédagogiques

Une problématique liée à l'exploitation des objets pédagogiques réside dans la nécessité de les indexer et de les réutiliser à différentes fins, dans différentes plates-formes ou pour différents publics. Par exemple, une séquence vidéo de deux minutes illustrant une loi en probabilité pourrait servir à la fois dans un cours de mathématiques et un cours sur le multimédia, ou dans le cadre

[3]Institute of Electronical and Electrnics Engineers - http://www.ieee.org
[4]Learning Object Metadata - http://ltsc.ieee.org/wg12/
[5]Agence Universitaire de la Francophonie

d'un test. Elle pourrait également être insérée dans différentes plates-formes. En résumé, de nombreux utilisateurs pourraient se servir de cette séquence vidéo de différentes façons en fonction de leurs besoins spécifiques, et l'assembler à d'autres objets pédagogiques selon le contexte dans lequel ils se trouvent (cf. figure 3.1). L'objectif de cette approche est d'aboutir à des composants réutilisables afin de rentabiliser leur production. Cette approche repose sur la notion de briques et d'agrégats. Un objet d'apprentissage est vu comme un assemblage de briques indépendantes que l'on peut utiliser et réutiliser pour construire d'autres objets pédagogiques.

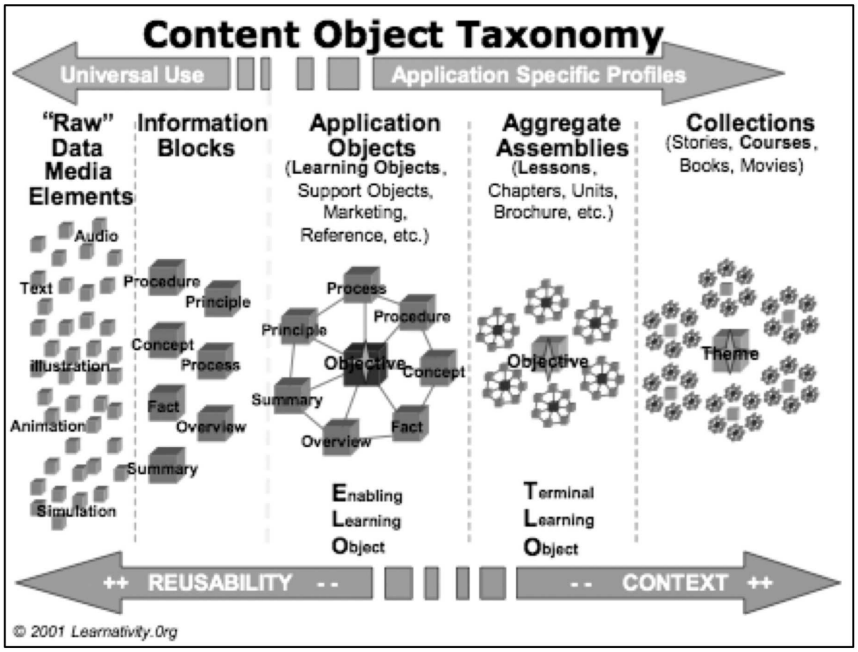


Fig. 3.1 – Taxonomie des contenus d'objets

Les avantages techniques et économiques liés à la programmation objet (réutilisabilité, fiabilité, interopérabilité, rapidité de développement) sont indiscutables mais le concept de réutilisation n'est pas aussi simple dans le cadre de l'apprentissage à distance que dans celui du jeu de Lego^{TM} ou de la réutilisation de morceaux de code existants. Certains auteurs [WKG04] ont montré comment la réutilisation peut mener à une contradiction pédagogique, et que l'approche centrée sur les ressources n'est pas celle à privilégier dans le cadre d'une formation à distance [Kop01] [McC03]. La pédagogie s'attache à atteindre de façon efficace des objectifs éducationnels en adaptant le contexte d'apprentissage en fonction du public visé, des conditions spatiales et/ou temporelles, etc. La mise en place du processus d'apprentissage n'est alors plus indépendante du contexte mais fortement liée aux connaissances de l'apprenant, aux habitudes d'apprentissage, au contexte social de l'environnement d'apprentissage, etc. Baumgartner a ap-

pelé cette contradiction *le paradoxe de la réutilisation des objets et de l'instruction*[6] [BBH06]. Il a alors proposé la division d'un objet pédagogique (au sens défini dans la section précédente) en deux types différents d'objets afin d'apporter une solution au paradoxe ROI : les objets d'information qui illustrent le contenu, et les objets éducationnels qui incluent les aspects pédagogiques et didactiques liés au contexte [BBH06]. Nous verrons dans la section 3.4 que certains standards tentent de regrouper ces deux types d'objet dans leur spécification.

Qu'il s'agisse d'objets d'information ou éducationnels, la réutilisation de ceux-ci peut de nouveau être appliquée ! Nous ne distinguons pas dans nos travaux ces deux types d'objets (le terme *objet pédagogique* désignera dans le reste du document les concepts définis dans la section précédente) mais nous nous efforçons de faciliter leur partage et leur réutilisation.
Une condition nécessaire pour développer le partage et la réutilisation d'objets pédagogiques consiste à les indexer à l'aide de descripteurs normalisés appelés *métadonnées*.

3.2.4 Les métadonnées

Même si le *web* et Internet fournissent de plus en plus de ressources pédagogiques, il est souvent difficile de les identifier comme telles car leur description est insuffisante dans l'objectif de les réutiliser dans un EIAH ; la qualité et la pertinence de la description sont des attributs importants pour des ressources utilisées dans un processus d'apprentissage. « Une métadonnée est une information ajoutée à un texte ou un logiciel afin de donner des informations sur son contenu » [Sim02]. Dans le cas d'un fichier informatique par exemple, le nom, la taille ou la date de création représentent les métadonnées de ce fichier.

Dans l'univers des contenus numérisés et offerts sur le *web*, les métadonnées représentent une fiche descriptive numérisée qui est associée à une ressource pédagogique et qui permet d'en spécifier divers aspects comme le type de ressource, sa date de création, l'auteur, etc. Mais la capacité des métadonnées à faciliter l'accès aux ressources en ligne dépend de l'existence d'un standard, doté de certaines propriétés [DHSW02] :

- **Durabilité** : les contenus pédagogiques doivent être capables d'affronter les changements technologiques sans la nécessité d'une ré-ingénierie ou d'un redéveloppement.
- **Modularité** : des éléments d'informations provenant aussi bien de différents schémas que de vocabulaires doivent pouvoir être combinés syntaxiquement et sémantiquement de façon interopérable.
- **Espace de nommage** : n'importe quel ensemble d'éléments de métadonnées doit constituer un espace de nommage lié par des règles et des conventions déterminées par son or-

[6] the Reusability Object and Instruction paradox - ROI

ganisme de maintenance. Ainsi, la déclaration de divers espaces de nommage au sein d'un bloc de métadonnées permet aux éléments de ces métadonnées d'être identifiés comme appartenant à l'un ou l'autre des ensembles d'éléments.

- **Extensibilité** : les systèmes de métadonnées doivent tenir compte des extensions de façon à satisfaire les besoins particuliers d'une application donnée. Certains éléments de métadonnées sont susceptibles d'être trouvés dans la plupart des schémas de métadonnées alors que d'autres seront spécifiques à une application ou un domaine particulier.
- **Multilinguisme** : il est essentiel d'adopter des architectures de métadonnées qui respectent les diversités linguistique et culturelle. Le multilinguisme est un des aspects du large problème que représente le multiculturalisme qui inclut, entre autres, la façon dont des dates sont représentées dans différents calendriers, la manière dont le texte est lu et affiché, les connotations culturelles de certains graphismes et pictogrammes, etc.

Plusieurs ensembles de métadonnées ont été ou sont en cours de spécifications actuellement et visent plus spécifiquement certaines fonctions comme par exemple la description du profil de l'apprenant, la structuration des contenus, ou les objets pédagogiques. Les différentes initiatives de standardisation de métadonnées présentées dans l'annexe A répondent au besoin de description des ressources pédagogiques et fournissent un moyen pour catégoriser et indexer des ressources d'apprentissage, facilitant autant que possible leur recherche et leur réutilisation [Moo01]. Nous assistons depuis 1997 à l'apparition de différentes solutions sur la scène de la normalisation mais la collaboration des différents acteurs entre eux, et plus particulièrement la convergence des diverses initiatives vers celle du LOM, permet une description commune des ressources d'apprentissage. Aussi, une partie des travaux du Comité ISO/IEC JTC1 SC36[7] qui travaille à la normalisation des technologies de l'information destinées à l'apprentissage, à l'enseignement et à la formation, porte sur les métadonnées pour les ressources d'apprentissage et devrait aboutir d'ici peu à la proposition de premiers volets d'une norme internationale (cf. figure 3.2).

D'autre part, les représentations de ces métadonnées par des formats de données standardisés comme XML[8] [BPSM+04b] ou RDF[9] laissent entrevoir des solutions d'interopérabilité entre systèmes renfermant des ressources pédagogiques. XML reste le standard le plus répandu aujourd'hui grâce à sa simplicité d'implémentation, mais les travaux en cours de réalisation au niveau du *binding* LOM/RDF devraient permettre une exploitation et un déploiement de celui-ci dans l'année en cours ou en 2007. Cette avancée permettra une interopérabilité à plus large échelle

[7] http://jtc1sc36.org/
[8] eXtensible Markup Language
[9] Resource Description Framework

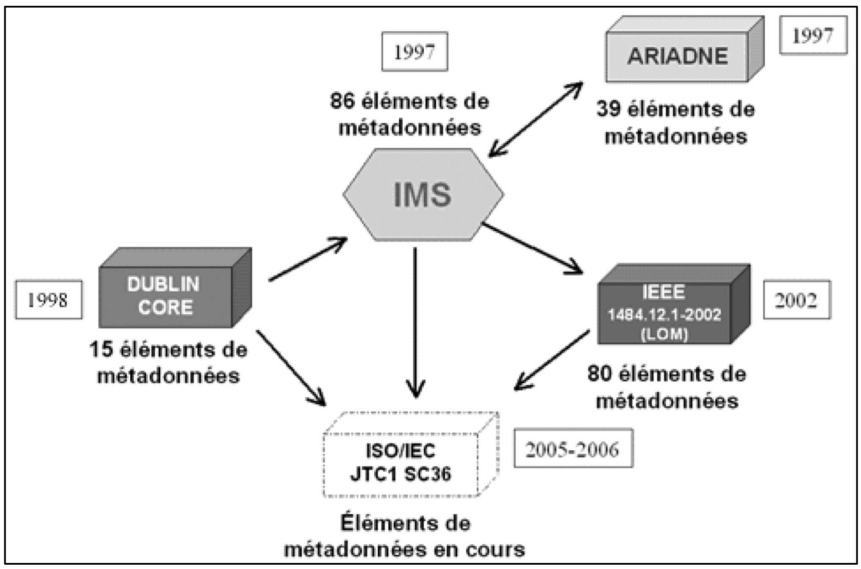


FIG. 3.2 – Synthèse des standards de métadonnées [Inc03]

avec les systèmes d'apprentissage basés sur les ontologies et le *web* sémantique qui utilisent, pour la plupart, le langage RDF pour décrire l'ensemble de leurs informations.

Dans la section qui suit, nous présentons les systèmes de stockage plus connus sous le nom de *vivier de connaissance* et qui offrent différents services de traitement sur les objets pédagogiques tels que la recherche, l'indexation ou la modification de ressources d'apprentissage. Nous décrivons les spécifications existantes favorisant leur interopérabilité avant de nous intéresser à la technologie des services *web* fortement déployée au sein des systèmes actuels et qui facilite leur accessibilité à grande échelle.

3.3 Les viviers de connaissance

Les objets pédagogiques sont stockés dans des viviers de connaissance, basés sur des SGBD[10]. Ils fournissent un moyen transparent pour partager et réutiliser les objets pédagogiques. Ces systèmes renferment de véritables objets pédagogiques, c'est-à-dire à la fois les documents eux-mêmes mais également leurs métadonnées. Cependant, une distinction doit être faite entre vivier de documents et vivier de références.

[10]Systèmes de Gestion de Bases de Données

3.3.1 Distinction entre viviers de documents et viviers de références

3.3.1.1 Les viviers de documents

Les viviers de documents, ou LOR[11] (aussi appelés LCMS[12]), contiennent des documents numériques tels que des fichiers texte, audio, image ou vidéo, ainsi que les métadonnées qui leur sont associées. Ces fichiers peuvent être stockés au sein même de la BD[13] ou dans un système de fichiers indépendant de la base ; dans ce cas, la BD ne contient que les métadonnées. Deux principaux viviers de connaissance existent à ce jour : ceux du réseau eduSource et et de la fondation ARIADNE.

Le projet eduSource Canada[14] est un réseau de banques d'objets d'apprentissage reliées d'un bout à l'autre du Canada et proposant aux éducateurs un ensemble d'outils et de protocoles. Les fournisseurs de contenu ont la possibilité de déposer des objets pédagogiques sur un serveur afin qu'ils soient facilement repérables par d'autres parties du réseau.

Le KPS[15] de la fondation ARIADNE [CHV+98] [DFC+01] consiste en une BD répartie sur divers sites en Europe et Amérique du Sud. Ses fonctionnalités sont de cataloguer et stocker des éléments pédagogiques électroniques. Actuellement environ 20 noeuds locaux (LKP[16]) échangent leur contenu grâce à un noeud central (CKP[17]) situé à Leuven en Belgique et à des noeuds régionaux (RKP[18]). En effet, un système de réplication permet de mettre en commun les documents situés dans les différents LKP et permet ainsi un partage et une réutilisation optimaux des ressources pédagogiques. La figure 3.3 illustre les liens entre les différents noeuds du KPS.

3.3.1.2 Les viviers de références

Contrairement aux viviers de documents, les viviers de références, ou Learning Object reFeratory (LOF), renferment uniquement les métadonnées associées aux objets pédagogiques, les documents consistant en une URL[19] représentée par un champ du modèle de métadonnées. Les ressources sont donc localisées n'importe où sur le *web* sur des systèmes et machines totalement indépendants de la base de données contenant les métadonnées. Cela peut représenter un inconvénient dans la mesure où, de nos jours, l'infrastructure de la Toile ne cesse d'évoluer,

[11] Learning Object Repository
[12] Learning Content Management System
[13] Base de Données
[14] http://www.edusource.ca/
[15] Knowledge Pool System
[16] Local Knowledge Pool
[17] Central Knowledge Pool
[18] Regional Knowledge Pool
[19] Uniform Resource Locators

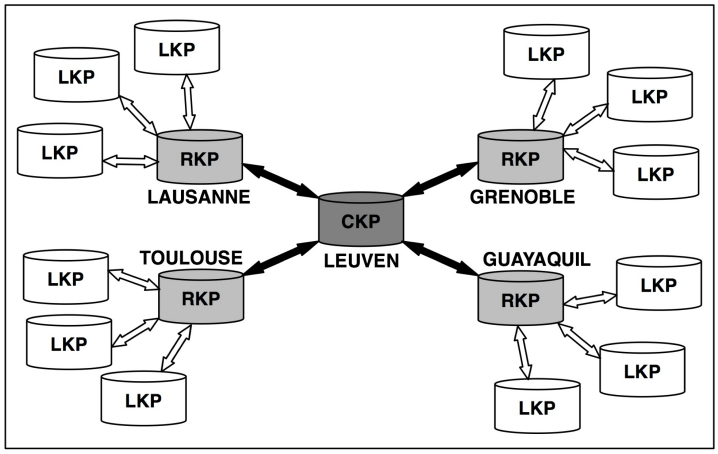


FIG. 3.3 – Le Knowledge Pool System de la fondation ARIADNE

favorisant ainsi le risque de trouver des liens obsolètes. Ces systèmes imposent donc de maintenir l'intégrité des métadonnées des objets pédagogiques, et plus précisément le descripteur renseignant la localisation du document. Néanmoins de nombreux viviers de connaissance sont aujourd'hui basés sur les LOF ; il n'est pas possible de présenter l'intégralité des LOF existants, nous évoquons ici les plus importants et qui émanent de différentes initiatives internationales :

- **MERLOT**[20] : ce vivier américain est un système libre dont le but est de donner accès à des documents pédagogiques aux étudiants de l'Enseignement Supérieur.
- **EdNA**[21] : EdNA Online, qui résulte d'un partenariat entre les différents gouvernements Australiens, a pour objectif de supporter et de promouvoir les avantages de l'Internet pour l'Aprrentissage et l'Education en Australie.
- **NIME**[22] : cette institution contribue à l'amélioration de l'Enseignement Supérieur au Japon par des actions de recherche et développement dans le domaine des contenus et méthodologies d'apprentissage en ligne.
- **EducaNext**[23] : engagée en 2000, l'initiative EducaNext est issue du projet IST[24] UNIVERSAL financé par le 5ème programme cadre de l'Union Européenne. L'objectif de ce projet était de créer un environnement innovant pour permettre l'échange de matériels pédagogiques à travers l'Internet.

La multitude de viviers de connaissance existants pose donc le problème de leur intercon-

[20]Multimedia and Educational Resource for Learning and Online Teaching - http://www.merlot.org
[21]Education Network Australia - http://www.edna.edu.au
[22]National Institute of Multimedia Education - http://www.nime.ac.jp/
[23]http://www.educanext.org
[24]Information Society Technologies

nexion. Dans cet objectif, l'IMS/GLC[25] a élaboré la spécification IMS-DRI[26] que nous présentons ci-après et qui vise à faciliter l'interopérabilité de ces différents viviers afin de permettre l'échange des métadonnées qu'ils renferment.

3.3.2 La spécification IMS Digital Repositories Interoperability

L'objectif de la spécification IMS Digital Repositories Interoperability [Con03a] est de fournir des recommandations pour l'interopérabilité des fonctions les plus communes associées aux viviers de connaissance. IMS-DRI n'emploie pas le terme *Learning Object Repository* (ou *reFeratory*) mais *Digital Repositories* que l'on pourrait traduire par *viviers numériques*. La spécification définit un vivier numérique comme « une collection de ressources accessible via un réseau sans connaissance préalable de la structure de cette collection. Un vivier peut contenir des documents ou les métadonnées qui décrivent ces documents. Toutefois, les documents et leurs métadonnées peuvent ne pas résider dans le même vivier »[27]. Nous pouvons alors assimiler ce concept à celui de vivier de connaissance défini précédemment.
IMS-DRI spécifie l'ensemble de fonctionnalités fonctionnelles suivant pour permettre la recherche, la récupération, le stockage et la délivrance de documents (cf. figure 3.4) :

- **Rechercher / Présenter** (*Search/Expose*).
- **Récolter / Présenter** (*Gather/Expose*).
- **Soumettre / Stocker** (*Submit/Store*).
- **Emettre une requête / Délivrer** (*Request/Deliver*).

La spécification introduit trois entités pour représenter les échanges entre viviers numériques et services d'interaction :

1. **Les rôles** (apprenant, enseignant, explorateur d'informations, agent).
2. **Les composants fonctionnels** pour les utilisateurs de ressources, les viviers, la gestion des accès et les services d'approvisionnement.
3. **Les services** tels que les annuaires.

IMS-DRI propose donc un modèle de référence pour faciliter l'accès aux contenus des viviers dans le contexte de la formation à distance par des applications telles que des portails, plates-formes d'apprentissage, etc. Toutefois, IMS-DRI se focalise sur deux types de viviers basés

[25] Instructional Management Systems Global Learning Consortium - http://www.imsglobal.org/
[26] IMS Digital Repositories Interoperability
[27] http://imsglobal.org/digitalrepositories/

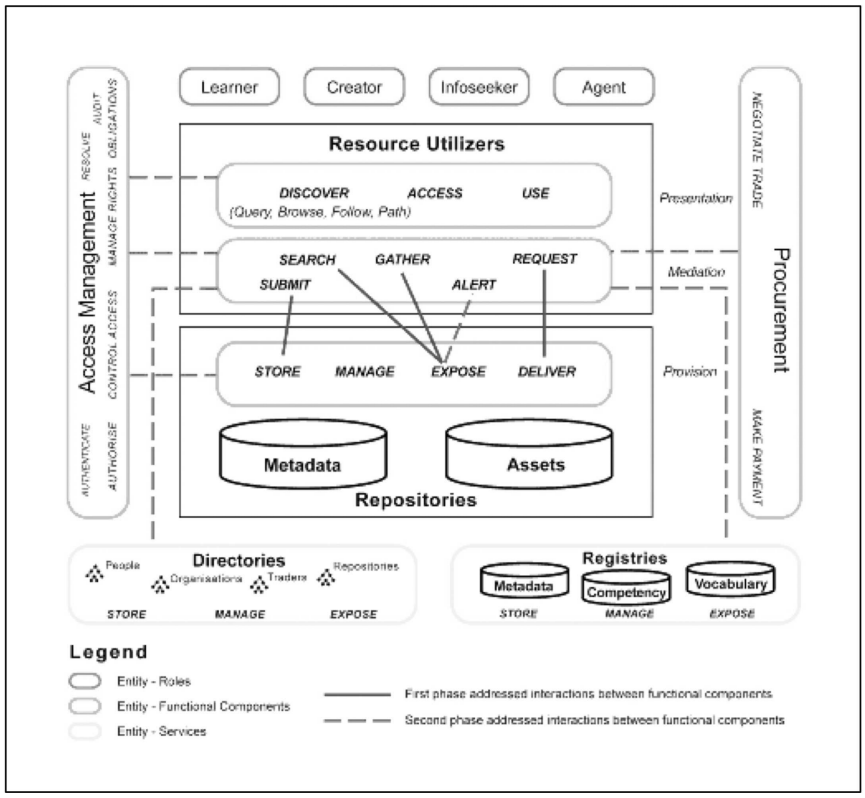


FIG. 3.4 – Architecture fonctionnelle de IMS DRI [Con03a]

sur le protocole Z39.50[28] ou sur le format XML. La spécification expose alors le concept de *chercheurs intermédiaires* qui agissent comme des traducteurs d'un format de recherche vers un autre, ou comme des agrégats capables de retrouver des métadonnées issues de différents viviers et qui les rendent disponibles pour la recherche.

La spécification IMS-DRI ne se soucie pas des problèmes liés à la gestion des authentifications ou des autorisations même si elle assure qu'ils seront pris en compte dans une version future. D'autre part, cette spécification étant complexe à déployer, seul le réseau eduSource implémente IMS-DRI [HEW04] à ce jour.

Alors dans un soucis de simplicité, une initiative de standardisation nommée Simple Query Interface (SQI) a pour objectif de spécifier un ensemble d'opérations qui permettent d'interroger des viviers de connaissance (LOR ou LOF) hétérogènes. Dans la section suivante nous exposons les principales caractéristiques et propriétés de cette initiative.

[28] http://www.niso.org/z39.50/z3950.html

3.3.3 La spécification Simple Query Interface

La spécification Simple Query Interface (SQI) [SMA+05] est un effort ouvert, collaboratif, ayant pour objectif d'atteindre un niveau élevé d'interopérabilité entre les viviers hétérogènes. Ses principales caractéristiques sont les suivantes :

- **Simplicité** et facilité d'implémentation.
- **Neutralité** en termes de langages de requête et de formats de résultat.
- Supporte les requêtes en mode **synchrone ou asynchrone**.

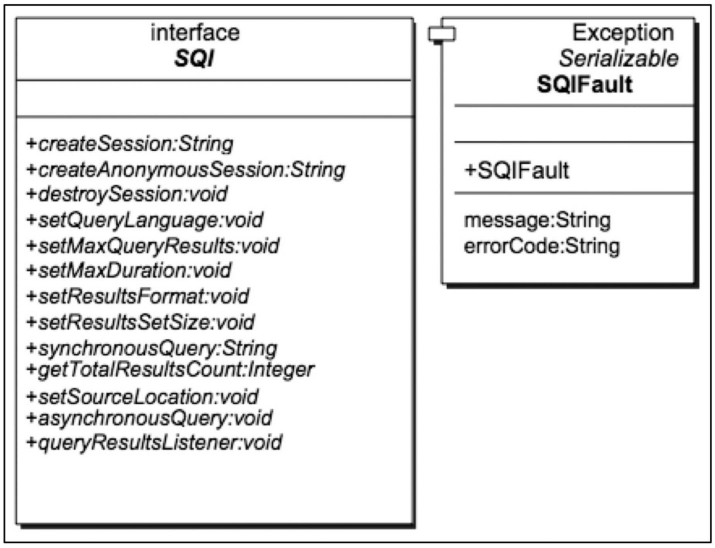


FIG. 3.5 – Diagramme de classes de la spécification SQI [SMA+05]

Les spécifications des méthodes proposées par SQI décrites par le diagramme de classes de la figure 3.5 sont regroupées en quatre catégories :

1. **Gestion des sessions :** une session doit être établie pour permettre le transfert d'informations entre une application cliente et un vivier implémentant la spécification SQI. Cependant, un login et un mot de passe ne sont pas toujours obligatoires puisqu'il existe la possibilité d'ouvrir une session anonyme. Une session sera active tant que celle-ci n'est pas détruite par le biais de la méthode destroySession, ou automatiquement au bout de 30 minutes d'inactivité.
2. **Gestion des requêtes :** outre les méthodes permettant de mettre en oeuvre les modes (a)synchrones décrits ci-après, un ensemble de méthodes permet de définir le langage de requête et le nombre maximal de résultats fourni par une requête. Une autre méthode permet d'obtenir le nombre total d'enregistrements retournés par une requête.

3. **Gestion du mode synchrone :** ce type de scénario est recommandé pour l'accès à un seul vivier de connaissance. Lorsqu'une application cliente utilise ce mode d'accès, les résultats de sa requête lui sont immédiatement retournés par la cible SQI.
4. **Gestion du mode asynchrone :** ce mode est utilisé dans le cas d'accès à plusieurs viviers simultanément. Ici la cible SQI récupère une requête puis la transfère vers les différents viviers mentionnés dans la requête. Le(s) vivier(s) traite(nt) la requête puis retourne(nt) les résultats à la cible SQI. Enfin, l'application pourra accéder aux résultats par le biais d'un *listener*. La figure 3.6 présente le diagramme de séquence UML[29] illustrant la réalisation d'une requête en mode asynchrone.

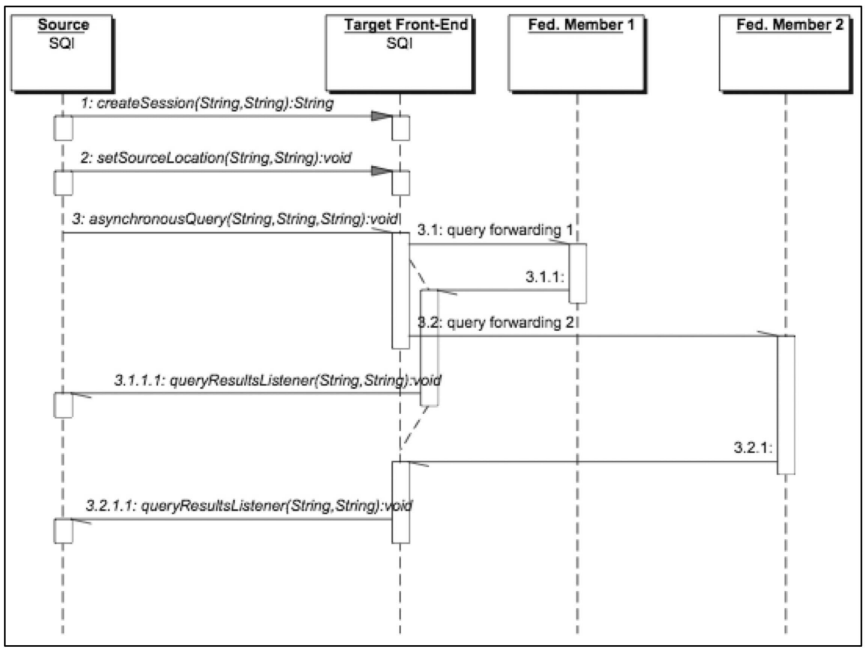


FIG. 3.6 – Diagramme de séquence d'une requête en mode asynchrone [SMA+05]

Le mécanisme d'exception fourni par SQI est volontairement basique. Il vise la simplicité dans le but d'offrir la plus grande opportunité d'implémentation par une variété d'applications. Lorsqu'une erreur se produit, chacune des méthodes de l'interface SQI est capable de lever une exception qui spécifie un code d'erreur prédéfini et un simple message texte.

3.3.4 Déploiement des services de traitement

La plupart des EIAH sont déployés au sein d'environnements distribués. Plusieurs *midd-*

[29] Unified Modeling Language

lewares comme CORBA[30] [Gro04], EJB[31] [DK05] ou DCOM[32] [HK97] fournissent un accès distant à leurs composants à travers des protocoles de communication propriétaires tels que IIOP[33] [Gro04], JRMP[34] ou ORPC[35] [EK01]. Paradoxalement, la force de leur approche orientée objet représente aussi un gros désavantage lorsqu'il s'agit de bénéficier des technologies *web* actuelles qui favorisent le passage à l'échelle [FM04].

La technologie des services *web*, initiée par de grandes entreprises issues des Technologies de l'Information, est aujourd'hui en cours de normalisation par le W3C[36] et s'articule autour des 3 composants suivants :

1. **SOAP**[37] [GHM+03] : ce protocole, basé sur le paradigme d'appel de procédures à distance, permet le transfert d'informations entre applications dans un environnement décentralisé et distribué. Un message SOAP est un flux ASCII enveloppé dans des balises XML et acheminé via un protocole de transport tel que HTTP[38] ou SMTP[39]. C'est le langage de description des échanges.
2. **WSDL**[40] [CCMW01] : ce langage permet de décrire en langage XML le contenu véhiculé par le protocole SOAP, c'est-à-dire le point d'accès, les méthodes fournies et la signature d'un service *web*. Ce langage de description définit, de manière abstraite et indépendante du langage d'implémentation du service *web*, l'ensemble des opérations et des messages qui peuvent être transmis vers/depuis un service *web* donné.
3. **UDDI**[41] [ftAoSIS04] : c'est une normalisation d'annuaire dynamique de services *web* permettant l'intégration et la recherche de tels services. UDDI s'appuie sur des standards du W3C et de l'IETF[42], à savoir XML, HTTP et les DNS[43].

Aujourd'hui, la normalisation des services *web* s'oriente d'une part autour de 3 axes horizontaux (cf. figure 3.7) :

1. **Couche de transport** : son objectif est de définir la structure des messages pour la communication. C'est la seule couche aujourd'hui normalisée ; elle utilise SOAP pour le

[30] Common Object Request Brocker Architecture
[31] Enterprise Java Bean™
[32] Distributed Component Object Model
[33] Internet Inter-Orb Protocol
[34] Java Remote Method Protocol
[35] Object Remote Procedure Call
[36] World Wide Web Consortium - http://www.w3.org/
[37] Simple Object Access Protocol
[38] HyperText Transfer Protocol
[39] Simple Mail Transfer Protocol
[40] Web Service Description Language
[41] Universal Description Discovery and Integration
[42] Internet Engineering Task Force - http://www.ietf.org
[43] Domain Name Server

transfert d'information et WSDL pour la description des données.

2. **Sémantique des messages** : cette couche traite de la normalisation des données participant aux échanges.
3. **Gestion des processus** : son but est de standardiser la gestion des processus qui s'étendent sur plusieurs applications à travers l'Internet.

Et d'autre part autour de 3 axes verticaux[44] (cf. figure 3.7) :

1. **Service d'annuaire** : s'intéresse aux moyens d'accès à un service ou à un fournisseur de services.
2. **Service de sécurité** : concerne la standardisation de l'authentification et des droits d'accès.
3. **Service de transaction** : vise la normalisation des moyens permettant l'intégrité des transactions impliquant l'utilisation de plusieurs services *web*.

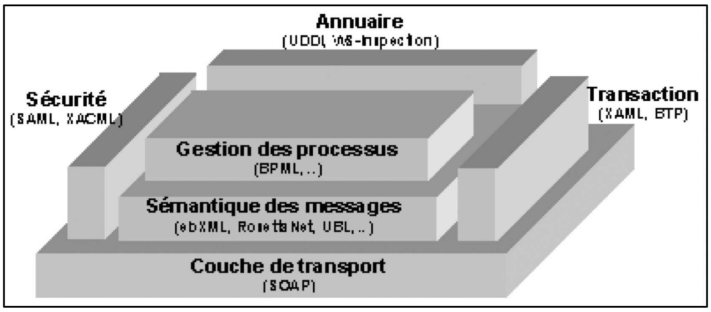


FIG. 3.7 – Normalisation des services *web*

3.3.5 Synthèse

La spécification SQI offre des opérations de base qui visent l'homogénéisation des services de traitement des objets pédagogiques stockés dans les viviers de connaissance. Cette récente proposition toujours en cours de spécification est déjà implémentée au sein de certains LOR, en particulier dans ceux impliqués dans le consortium international GLOBE[45] dont l'objectif est de fournir un réseau d'objets pédagogiques de qualité qui prennent en compte les standards de la formation ouverte et à distance. SQI offre donc une perspective intéressante pour l'interconnexion de viviers indépendants et facilite le travail de fédération de différents viviers. Toutefois,

[44] http://www.labo-sun.com/resource-fr-essentiels-832-15-java-j2ee-les-web-services-publication-de-services.htm
[45] http://globe.edna.edu.au/globe/go

tous les LOR ne sont pas (encore) compatibles avec ce standard et nous devrons être capables d'intégrer d'autres spécifications dans notre proposition afin de démontrer le caractère générique et indépendant de la fédération mise en oeuvre.

Aussi, la plupart des viviers de connaissance ont opté pour le déploiement des services de traitement sous la forme de services *web* dans le but de favoriser leur interopérabilité et l'échange du matériel pédagogique qu'ils renferment : cette technologie, indépendante d'un langage ou d'une plate-forme spécifique, bénéficie de son ouverture et de son étendue basées sur des standards communs tels que XML et HTTP, de sa capacité à "traverser" les pares-feus grâce aux protocoles de transport utilisés, et de sa facilité d'implémentation grâce aux multiples outils existants [BR01]. D'autre part, le protocole SOAP véhicule des informations XML et facilite donc le transport des descripteurs d'objets pédagogiques qui respectent également ce standard du W3C.

3.4 Les plates-formes de gestion d'apprentissage

3.4.1 Définitions et outils intégrés

3.4.1.1 Définitions

L'IGPDE[46] définit une plate-forme d'apprentissage, ou plate-forme de téléformation, comme un « système informatique conçu pour optimiser, sur un réseau Internet ou Intranet, la gestion de l'ensemble des activités de formation, depuis l'information sur l'offre, l'inscription des participants, la distribution des ressources, l'organisation de parcours individualisés, le suivi par le tuteur et du tuteurat (gestion intégrée des intéractions apprenants-formateur), l'animation de communautés d'apprentissage ».

Dans la communauté anglo-saxone, plusieurs termes sont utilisés pour désigner un tel système informatique. Paulsen [Pau02] utilise le terme *Learning Management System* (LMS) pour désigner « un grand nombre de systèmes qui organisent et fournissent un accès à des services d'apprentissage en ligne à des étudiants, enseignants, et administrateurs. Ces services incluent généralement un contrôle d'accès, un approvisionnement de contenus pédagogiques, des outils de communication, et des moyens pour organiser des groupes d'utilisateurs. Un synonyme de ce terme est *plate-forme d'apprentissage* »[47].
Kaplan-Leiserson[48] fournit une autre définition du terme LMS [Pau02] : « logiciel qui automatise l'administration d'événements d'apprentissage. Un LMS enregistre des utilisateurs, mémorise

[46]Institut de la Gestion et du Développement Economique - http://www.institut.minefi.gouv.fr
[47]Traduction de l'anglais
[48]http://www.learningcircuits.org/glossary.html

des cours dans un catalogue, et recueille les données des apprenants ; il fournit également des rapports à des fins de gestion. Un LMS est typiquement dédié à la manipulation de cours par de multiples éditeurs et fournisseurs. Il n'inclut généralement pas de fonctionnalité auteur, il se focalise sur la gestion des cours créés par une variété d'outils externes ».
Enfin, Hall[49] reprend et complète la définition de Kaplan-Leiserson en décrivant certaines fonctionnalités qu'un LMS pourrait (ne pas) fournir : outils auteurs, gestion de classes virtuelles, gestion des compétences, gestion des connaissances, apprentissage certifié, personnalisation, tutorat, *chat*, tableau de discussion.

Le terme *Virtual Learning Environment* (VLE) est également utilisé dans la littérature anglaise pour désigner une plate-forme d'apprentissage, mais certains diront qu'un VLE se focalise moins sur les fonctionnalités liées à la gestion de l'apprentissage. Hall[50] définit un *Virtual Learning Environment* comme suit : « logiciel tout-en-un qui peut faciliter l'apprentissage en ligne pour une organisation. Il inclut les fonctions d'un LMS relatives aux cours délivrés dans l'environnement d'apprentissage, mais il peut ne pas être capable d'observer les cours qui n'ont pas été créés au sein de cet environnement d'apprentissage particulier ».

Nous constatons ainsi qu'une grande variété de termes est utilisée pour désigner des systèmes informatiques d'apprentissage en ligne. Dans la suite du document, nous emploierons les termes plate-forme d'apprentissage, plate-forme de téléformation ou LMS pour désigner les systèmes présentés dans cette section. L'objet de nos travaux n'est pas de discuter des fonctionnalités nécessaires à l'apprentissage en ligne, mais nous nous attachons à intégrer ce type de systèmes au sein de notre environnement informatique pour l'apprentissage humain. Cependant, la partie suivante expose les fonctionnalités de base d'une plate-forme d'apprentissage afin de donner au lecteur un aperçu des services fournis et des différentes approches mis en oeuvre au sein de ces systèmes.

3.4.1.2 Outils intégrés

Afin d'assurer la gestion de l'ensemble des activités de formation, une plate-forme d'apprentissage en ligne doit être agrémentée d'un certain nombre d'outils visant à faciliter l'échange entre apprenants et enseignants. Aujourd'hui une multitude de LMS sont disponibles sur le marché de la FOAD, et les fonctionnalités offertes varient d'un système à un autre selon l'approche qu'il privilégie. Toutefois, certains outils sont indispensables à l'ensemble des acteurs participant à une activité pédagogique ; nous tâchons de décrire ici ceux qui sont essentiels au processus d'apprentissage et qui sont implantés dans la majorité des plates-formes de téléformation.

[49] http://www.brandon-hall.com/free_resources/glossary.shtml
[50] http://www.brandon-hall.com/free_resources/glossary.shtml

Il est clair que la liste des fonctionnalités exposées n'est pas exhaustive et que tous ces outils ne sont pas présents dans la totalité des systèmes existants.

En premier lieu, un LMS doit être capable de gérer l'identification des différents acteurs, ainsi que de distinguer les rôles de ces acteurs au sein du système. Quatre types d'acteur sont généralement reconnus : l'administrateur global de la plate-forme, les apprenants, les enseignants/tuteurs et les responsables pédagogiques des formations. Pour chacun de ces acteurs, différents outils sont suggérés.

Côté administrateur

L'administrateur de la plate-forme dispose de statistiques concernant l'usage du serveur telles que la répartition des utilisateurs par cours ou le détail des accès aux cours/outils ; cette personne est généralement responsable de la gestion des cours (création, modification, suppression, sauvegarde, restauration) et de leur paramétrage.
C'est également lui qui va assurer la gestion des utilisateurs, c'est-à-dire la création des comptes, l'affectation des droits d'accès correspondants et leur paramétrage, ou la création de groupes. Enfin, la possibilité de modifier l'interface graphique et la convivialité du système peut lui être offerte.

Côté apprenant

Du côté de l'apprenant, des outils généraux tels que l'inscription en ligne, la navigation entre les différentes matières enseignées ou la visualisation rapide des nouveautés sont souvent suggérés.
L'accès rapide aux cours ou aux exercices, la visualisation des résultats ou l'état d'avancement des tests réalisés constituent les supports de formation. Des outils synchrones (*chat*, visio-conférence) ou asynchrones (messagerie, forum, espace commun de groupe) représentent les modes de communication usuels, auxquels peuvent s'ajouter d'autres outils tels que les emplois du temps, les glossaires, les moteurs de recherche, etc.

Côté enseignant

Au niveau des supports de cours, le nombre de fonctionnalités offertes à l'enseignant/tuteur sont nombreuses : création d'un cours accessible à des utilisateurs prédéfinis, modification du parcours pédagogique, insertion de liens vers des outils de communication, connaissance de l'assiduité (statistiques d'accès) des apprenants, visualisation des travaux rendus, notation des travaux, suppression d'un travail rendu pour autoriser une nouvelle remise, création de quizzs,

de rapports d'évaluation, etc. C'est l'enseignant qui modère les forums en supprimant, déplaçant ou masquant les messages ; il peut également sauvegarder ou accéder aux archives des *chats*. Afin d'obtenir un aperçu du parcours pédagogique élaboré, un enseignant peut émuler le mode apprenant et modifier l'affichage ou la disposition des outils et activités inclus dans le cours. Des outils complémentaires comme la prise de notes, le tableau blanc ou le maintien d'un journal de bord des modifications peuvent aussi être présents.

Côté responsable pédagogique

Enfin, le responsable pédagogique d'une formation a la possibilité de rendre un cours public ou privé, d'accepter, ou non, l'inscription des apprenants à un cours sans autorisation préalable, de décrire brièvement les objectifs/pré-requis, de choisir la langue d'enseignement d'un cursus, ou de créer des annonces particulières à un cours.

Nous pouvons constater que le nombre d'outils et services intégrés au sein d'une plate-forme d'apprentissage est très important, ce qui peut provoquer une hétérogénéité de ces systèmes. L'IEEE a donc spécifié, à partir de composants et de rôles attribués à ces composants, l'architecture LTSA[51] [FT98] afin de modéliser l'ensemble des services d'une plate-forme de téléformation.

3.4.2 L'architecture LTSA

LTSA est un modèle fonctionnel composé de diagrammes de flots de données (DFD[52]) qui fournit un cadre de référence pour favoriser l'interopérabilité et la portabilité des différents LMS. Un diagramme de flots de données se compose de :

- **Processus** : c'est une fonction à laquelle on attribue des données en entrée et qui calcule des valeurs en sortie.
- **Flot de données** : c'est le passage d'une donnée d'un point vers un autre.
- **Réservoir de données** : c'est un objet passif qui permet à un traitement d'accéder à des valeurs de données qui lui sont nécessaires.

La figure 3.8 illustre les différents composants de l'architecture LTSA. Les quatre processus représentés par des cercles sont les suivants :

- **Learner Entity** : ce processus est une abstraction d'un étudiant réel. Il peut représenter un simple étudiant ou un groupe d'étudiants.

[51] Learning Technology System Architecture
[52] Data Flow Diagram

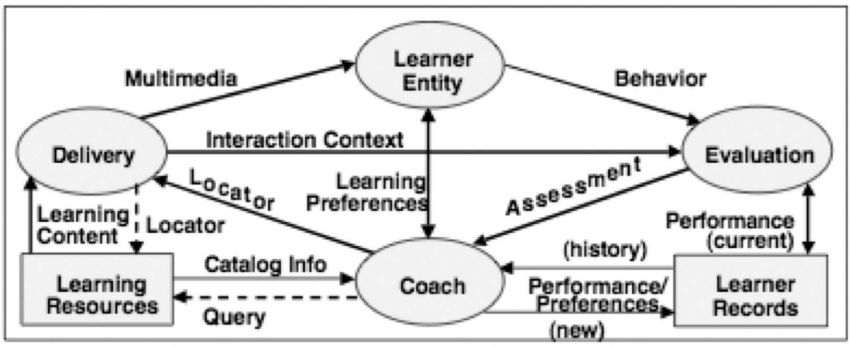


FIG. 3.8 – Les composants de l'architecture LTSA

- **Coach** : ce composant représente un tuteur, un responsable ou une administration.
- **Evaluation** : ce composant capture les données de l'étudiant pour les enregistrer.
- **Delivery** : ce processus délivre les ressources multimédias à l'étudiant.

Les réservoirs de données, représentés par des rectangles sur la figure 3.8, sont les suivants :

- **Learning Resources** : ce sont des ressources pédagogiques.
- **Learning Records** : toutes les informations d'un élève sont gérées par ce réservoir de données.

Les flots de données représentent les interactions entre les processus et les réservoirs de données. Ils sont représentés par des flèches sur la figure 3.8 :

- **Learning Preferences** : ce flot de données représente les échanges entre l'étudiant et le tuteur, il négocie les préférences.
- **Behavior** : ce sont des informations relatives aux activités de l'étudiant et utilisées pour évaluer les résultats de la leçon.
- **Multimedia** : ce sont des flots multimédias pouvant être de la vidéo, du son ou des images.
- **Interaction Context** : ces informations sont collectées en même temps que les activités de l'étudiant (flot Behavior) par le processus *Evaluation* pour enregistrer les performances de l'élève.
- **Assessment** : ce message fournit des informations au sujet de l'état de l'étudiant qui peuvent être utilisées par le tuteur.

- **Locator** : il s'agit d'un identifiant de la ressource multimédia.
- **Performance (current)** : les performances de l'élève sont enregistrées dans le réservoir de données *Learning Records*.
- **Performance / Preferences (new)** : l'administration met à jour les données de l'élève.
- **Query and Catalog Info** : le flot *Query* est une interrogation sur des ressources disponibles alors que le message *Catalog Info* est le résultat de cette recherche.
- **Learning Content** : c'est le contenu de la ressource.

Les plates-formes d'apprentissage disponibles aujourd'hui présentent l'ensemble ou une partie des composants définis dans l'architecture LTSA. Cependant, chacun de ces systèmes proposent des formats de cursus d'apprentissage qui leur sont propres, empêchant ainsi le partage et la réutilisation des objets/cursus pédagogiques (renfermés dans le réservoir de données *Learning Resources*) par d'autres acteurs utilisant un autre système d'apprentissage. Ainsi, les acteurs de la standardisation évoqués dans l'annexe A ont initié, depuis 2001, plusieurs spécifications visant à rendre les contenus d'apprentissage et cursus pédagogiques interopérables et réutilisables d'une plate-forme à une autre afin de simplifier et d'optimiser le travail des concepteurs de cours. Nous évoquons dans les deux sections qui suivent les standards de fait les plus utilisés aujourd'hui, à savoir SCORM[53] et IMS-LD[54].

3.4.3 SCORM

Le projet ADL[55] visait à moderniser l'éducation et la formation à l'aide des technologies de l'information et de la communication [Cho02]. Un de ses mandats était de promouvoir la collaboration entre les gouvernements, les milieux académiques et les milieux d'entreprise en vue de développer des normes pour la formation en ligne. Dans cette foulée, en 1997, ADL s'associait avec plusieurs partenaires afin d'identifier les éléments critiques pour une technologie éducative fondée sur le *web*. Dans ce contexte, SCORM [Lea04a] définit un modèle d'agrégation de contenus et d'environnement d'exécution pour les objets d'apprentissage sur le *web*. Ces séries de spécifications sont basées sur les travaux de l'AICC, de l'IEEE et de l'IMS. Le modèle SCORM permet la production d'objets pédagogiques réutilisables et indépendants qui peuvent circuler librement à travers divers environnements d'apprentissage. Il définit des formats de données ainsi que des moyens d'y accéder afin de faciliter l'interopérabilité des objets d'apprentissage. Les critères du modèle de référence sont la durabilité, l'interopérabilité, l'accessibilité, et la possibilité de réutilisation par de nombreux outils d'apprentissage. Ces critères sont exclusivement

[53] Sharable Content Object Reference Model
[54] IMS Learning Design
[55] Advanced Distributed Learning - http://www.adlnet.org

techniques et ne concernent donc que la bonne disposition des contenus vis à vis des standards technologiques. L'organisation du modèle de référence SCORM (cf. figure 3.9) est divisée en trois parties :

- **Le document 1** présente un aperçu du projet ADL [Lea04b], une justification du modèle de référence, un résumé des lignes directrices ainsi que des spécifications techniques.
- **Le document 2** [Lea05a] présente le modèle d'agencement et de structuration de contenus. Il contient des directives permettant d'identifier et d'agréger les ressources en un contenu d'apprentissage structuré. Le dictionnaire des méta-données basées sur le LOM permet d'identifier les ressources pédagogiques alors que le conditionnement de contenu, basé sur l'IMS Content Packaging, permet de transporter les ressources d'un système de gestion à un autre, ou entre viviers de connaissances.
- **Le document 3** [Lea05b] traite de l'environnement d'exécution et de communication (RTE[56]). Ce dernier contient les directives pour le lancement et le suivi du contenu dans un environnement *web* ainsi que pour la communication avec ce contenu.

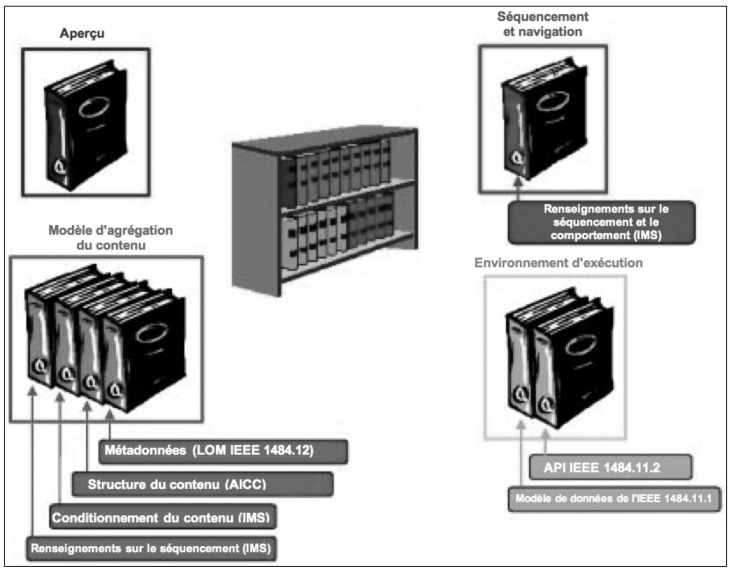


FIG. 3.9 – La bibliothèque du SCORM [Lea04a]

3.4.3.1 Le modèle d'agrégation de contenu

Le document 2 du modèle SCORM définit les spécifications pour le modèle de contenu, les métadonnées, et le conditionnement de contenus ; il décrit les composantes spécialisées de

[56]Run Time Environment

SCORM.

Ainsi SCORM définit une structure arborescente de représentation, avec au plus haut niveau le cours (agrégation de contenus), composé de blocs ou SCO[57], eux mêmes composés de blocs plus petits (Asset) ou de SCO (cf. figure 3.10).

SCORM fait référence au standard IEEE LOM pour décrire les agrégations de contenus, les SCO et les Assets (cf. figure 3.10). SCORM permet d'avoir une cartographie hiérarchisée des valeurs, des unités de contenus et des blocs qui peuvent être utilisés par le dispositif de formation. Enfin, le modèle de conditionnement de contenu SCORM est basé sur l'IMS Content Packaging [Con05] présenté dans l'annexe A.3.

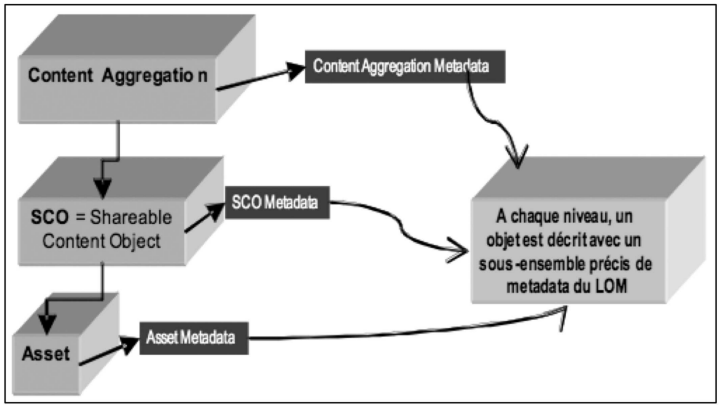


FIG. 3.10 – Le modèle d'agrégation de contenus et les méta-données correspondantes [Per03]

Ainsi, la spécification de conditionnement de contenu permet de définir un ensemble de structures utilisées pour échanger le contenu et permet l'interopérabilité entre les systèmes, les outils de développement et les dispositifs de FOAD désirant importer, exporter ou assembler des ressources d'apprentissage.

3.4.3.2 L'environnement d'exécution

L'environnement d'exécution [AHL+02] des objets d'apprentissage partageables SCO inclut des recommandations pour le lancement, la communication et le suivi du contenu dans un environnement *web*.
Les procédures et responsabilités pour l'établissement de la communication entre les ressources d'apprentissage délivrées (SCO, Asset) et le LMS varient selon le type des ressources d'apprentissage SCORM qui sont lancées. Dans le cas des Assets, SCORM exige que le LMS les lancent

[57] Sharable Content Object

en utilisant le protocole HTTP alors que dans le cas du SCO, le modèle de lancement SCORM exige que le LMS lance un SCO et que seul un SCO soit actif à chaque instant. De plus, le modèle exige que les SCO ne puissent pas lancer d'autres SCO.

Si de plus en plus de plates-formes d'apprentissage sont compatibles avec les spécifications SCORM[58] qui permettent de mettre en oeuvre des contenus pédagogiques interopérables et réutilisables sur différents LMS, ces systèmes de formation à distance intègrent un ensemble d'outils d'un point de vue technique mais ne prennent pas en compte un cadre pédagogique explicite [Fae01] [Kop00]. Dans cet objectif, la spécification IMS-LD cherche à incorporer la flexibilité pédagogique et complète certains aspects traités par les spécifications de l'annexe A.

3.4.4 IMS Learning Design

3.4.4.1 La Spécification

La version 1.0 de la spécification IMS Learning Design a été publiée en février 2003 par le IMS/GLC [Con03d][Con03b][Con03c]. A. Lejeune [Lej04] indique que « cette proposition, très largement inspirée du langage EML[59] développé par R. Koper [KM04] fournit un cadre conceptuel de modélisation d'une Unité d'Apprentissage et prétend proposer un bon compromis entre d'une part la généricité permettant de mettre en oeuvre des approches pédagogiques variées et d'autre part, la puissance d'expression permettant une description précise de chaque unité d'apprentissage ».

Cette spécification flexible, souvent appelée *méta-modèle pédagogique*, permet de représenter et d'encoder des structures d'apprentissage pour des apprenants à la fois seul et en groupe, rassemblés par rôles tels que "Apprenants" et "Equipe" [BBG+05]. La modélisation d'un plan de cours en IMS-LD est réalisée en définissant des rôles, des activités d'apprentissage, des services et plusieurs autres éléments, ainsi qu'en construisant des unités d'apprentissage. Finalement, la structure de l'unité, les rôles et les activités construisent le scénario d'apprentissage qui doit être exécuté dans un système compatible avec IMS-LD.

IMS-LD utilise la métaphore théâtrale [BANK05], ce qui implique l'existence de rôles, de ressources et de scénario d'apprentissage : une pièce est divisée en un ou plusieurs actes et conduite par plusieurs acteurs qui peuvent endosser différents rôles à différents moments. Chaque rôle doit réaliser un certain nombre d'activités pour achever le processus d'apprentissage. De plus, tous les rôles doivent être synchronisés à la fin de chaque acte avant de traiter l'acte suivant.

[58] http://www.conform2scorm.com/
[59] Educational Modelling Language - langage de modélisation pédagogique

3.4.4.2 L'unité d'apprentissage

L'objectif premier de IMS-LD est de modéliser des unités d'apprentissage. [Con03d] définit une unité d'apprentissage comme tout élément pouvant être utilisé dans l'apprentissage tel qu'un cours, un module, ou une leçon, et qui inclut une variété d'activités prédéfinies comme des devoirs, des activités de discussion ou des activités de résolution de problèmes fournies par les enseignants, formateurs, tuteurs, etc. Une activité d'apprentissage représente donc une description d'une méthode permettant aux apprenants d'atteindre certains objectifs en réalisant certaines activités dans un ordre défini au sein d'un contexte pédagogique. Une unité d'apprentissage peut alors être modélisée par un paquet de contenu IMS (cf. figure A.1) auquel sont ajoutées des activités pédagogiques (cf. figure 3.11). Un paquet de contenu IMS est appelé *unité d'apprentissage* si et seulement s'il inclut des activités d'apprentissage valides dans la partie organisation du fichier *manifeste* évoqué dans la section 3.4.3.1.

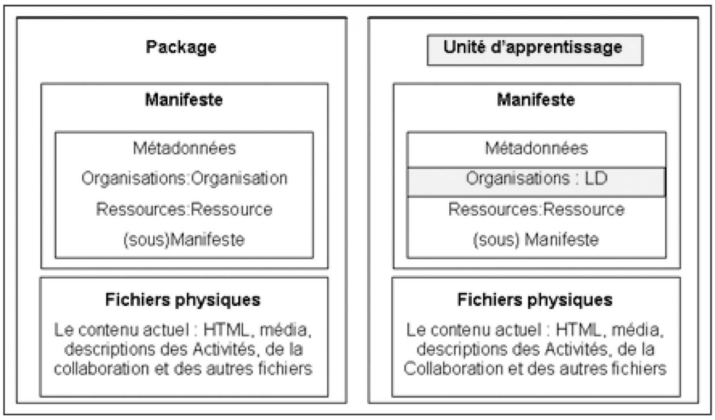


Fig. 3.11 – IMS Content Packaging et IMS Learning Design [KT05]

La différence entre IMS Content Packaging et IMS-LD réside d'une part dans le fait que la première spécification n'inclut aucune méthode ou pédagogie dans le fichier *manifeste*, et d'autre part que la seconde ajoute une déclaration complète sous le label *Organisations.*

3.4.4.3 La structure d'implémentation

La spécification de IMS-LD définit trois niveaux d'implémentation et de compatibilité (cf. figure 3.12) :

- **Le niveau A** contient le noyau de la conception pédagogique d'IMS (rôles, activités élémentaires et ressources) ainsi que leur coordination. Les activités d'apprentissage sont simplement ordonnées dans le temps pour être exécutées par des apprenants en utilisant

les objets et/ou les services d'apprentissage ; le déroulement des activités dans le cadre du processus d'apprentissage est constant.

- **Le niveau B** ajoute des propriétés et des conditions au niveau A, offrant ainsi la possibilité de personnaliser le séquencement et l'évolution d'un scénario pédagogique en fonction de la valeur d'une ou plusieurs propriétés ou du résultat de l'évaluation d'une expression. Le déroulement des activités dans le cadre du processus d'apprentissage est dynamique. La séparation des propriétés et conditions dans un niveau à part entière permet également de l'utiliser indépendamment du reste de la spécification.
- **Le niveau C** ajoute des notifications au niveau B qui permettent de déclencher une ou plusieurs activités en fonction de l'apparition d'un événement ; l'environnement d'exécution doit donc supporter un mécanisme de notification. Suite au dépôt d'un devoir par un apprenant, un courrier électronique pourra par exemple être envoyé à l'enseignant afin que celui-ci puisse le corriger.

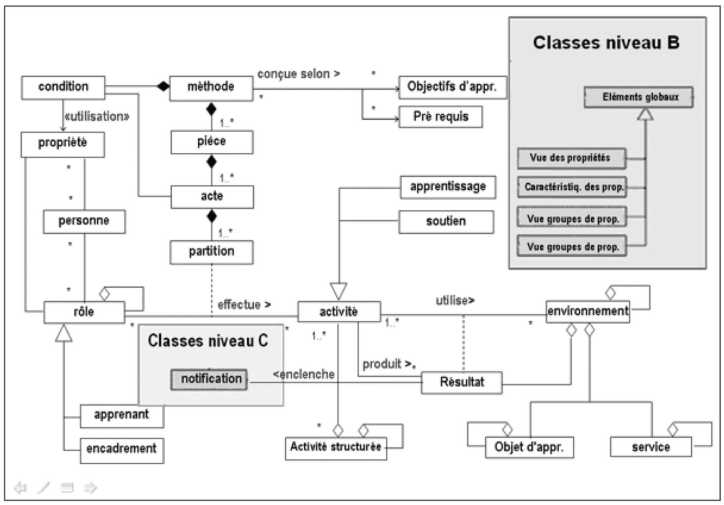


FIG. 3.12 – Niveaux A, B et C de la spécification IMS-LD [Con03d]

La combinaison des trois niveaux de représentation permet de spécifier et d'implémenter une grande diversité de contenus pédagogiques dans les environnements de Formation Ouverte et A Distance. Cette spécification, actuellement en cours de standardisation, suscite déjà un fort intérêt dans les communautés francophone et internationale [BANK05]. Au sein du LICEF[60] et dans le cadre du projet LORNET[61], l'équipe de G. Paquette a développé un éditeur graphique nommé MOT+ IMS-LD[62] [PlTL05] ainsi qu'un outil [PMlT+05] permettant à l'utilisateur de

[60] Laboratoire en Informatique Cognitive et Environnements de Formation - http://www.licef.teluq.uquebec.ca/
[61] Learning Object Repositories NETwork - http://www.lornet.org/
[62] http://www.unfold-project.net/about_folder/events/cops/bcna2/general_resources_folder/tools/mot/

représenter de façon hiérarchique la représentation d'une activité pédagogique IMS-LD. L'Open University of the United Kingdom propose un autre éditeur [GTZ05] basé sur le paradigme Client/Serveur qui permet aux enseignants de créer leurs cours en ligne, de les visualiser, de les gérer et enfin d'y participer.

Cependant, la (récente) proposition de la spécification IMS-LD ne satisfait pas tous les acteurs de la formation en ligne, en particulier au niveau des solutions proposées pour les activités d'apprentissage collaboratives.
Partant du besoin des enseignants d'exprimer des scénarios d'apprentissage collaboratifs afin de favoriser la réutilisation et le partage des pratiques pédagogiques et ce, quel que soit leur modèle pédagogique de référence, l'ENT[63] "Cartable électronique" propose un "Modèle de Participation" [MFC+04] conçu pour la description d'activités conjointes. A l'inverse des scénarios IMS-LD qui imposent un nombre fini de niveaux de structuration, le Modèle de Participation introduit la notion de rôles locaux à travers les rôles qui sont liés localement à chacune des activités. L'assemblage se fait en explicitant les liens existants entre rôles d'une activité à une autre.
Enfin, [GLD05] propose une extension de la spécification IMS-LD afin d'être en mesure de décrire le modèle d'organisation du tutorat suggéré dans [GD04] et [GDL04] qui s'articule autour des trois composantes suivantes de l'activité d'accompagnement des apprenants : le tutorant, le tutoré et la nature du tutorat. Jusqu'à présent, la spécification IMS-LD ne permet pas de décrire précisément la nature de la tâche du tuteur et la description plus claire des rôles. Ainsi, cette extension a pour objectifs « (1) d'organiser les tâches entre les acteurs tuteurs et apprenants, (2) d'assurer auprès des apprenants un accompagnement adapté à la situation d'apprentissage et (3) de spécifier les outils support à l'activité des acteurs de la formation ».

A l'heure actuelle, aucune plate-forme d'apprentissage en ligne n'intègre la spécification IMS-LD. Cependant, le nombre d'outils auteur permettant de créer des unités d'apprentissage IMD-LD est en constante augmentation, et des outils comme LAMS[64] [Dal03] ou Reload[65] permettent d'implémenter cette spécification intuitivement. Le système LAMS offre la possibilité d'être intégré au sein d'un LMS, mais des forums de discussion actifs tels que celui de la plate-forme d'apprentissage MOODLE[66] portent à croire que la prise en compte de cette spécification sera effective d'ici peu.

[63] Espace Numérique de Travail
[64] Learning Activity Management System - http://www.lamsinternational.com
[65] http://www.reload.ac.uk/
[66] Modular Object-Oriented Dynamic Learning Environment - http://moodle.org/mod/forum

3.4.5 Synthèse

Les plates-formes d'apprentissage jouent un rôle primordial dans un EIAH puisqu'elles représentent les systèmes par lesquels les apprenants apprennent les cursus pédagogiques élaborés par les enseignants ou formateurs. Une multitude de plates-formes ont été développées dans les milieux publics ou privés pour répondre à des besoins spécifiques de formation, ce qui a provoqué une diversité des services et fonctionnalités mis à disposition des acteurs de la formation en ligne.

Deux normes principales ont émergé depuis le début des années 2000 afin de réduire l'hétérogénéité des multiples LMS existants. La première, SCORM, vise l'interopérabilité des contenus de formation d'une plate-forme à une autre et connaît un large succès dans la mesure où elle est prise en compte dans de nombreux systèmes. Cependant, elle ne permet pas d'associer un scénario pédagogique à ces contenus. La récente initiative IMS-LD du consortium IMS, qui vise à combler ce manque en offrant un moyen pour incorporer un cadre pédagogique dans les contenus eux-mêmes, est très intéressante mais reste aujourd'hui au stade expérimental.

Même si ces deux normes favorisent l'exploitation de ressources pédagogiques communes auxquelles sont associés des scénarios d'apprentissage, les plates-formes de téléformation demeurent des systèmes fermés qui ne sont pas, ou très peu, ouverts sur d'autres composants d'un EIAH. Ces systèmes ne communiquent pas directement entre eux, ils se contentent d'intégrer les normes existantes afin de présenter une certaine compatibilité avec les objets pédagogiques produits. Au niveau du stockage du matériel pédagogique, elles proposent un espace dédié au sein du serveur sur lequel elles sont déployées, mais ne présentent aucun mécanisme pour y associer des métadonnées conformes aux standards exposés dans l'annexe A. D'autre part, cet espace dédié empêche tout partage et réutilisation des ressources qu'il contient puisque son accès est réservé et spécifique à la plate-forme elle-même.

3.5 Conclusion du chapitre

Ce chapitre nous a permis de percevoir l'effervescence qui se vit en matière de normalisation de la formation en ligne depuis la fin des années 1990, et la multitude des acteurs qui interviennent dans le développement de normes et de standards d'interopérabilité pour le matériel pédagogique électronique :

- **Au niveau des métadonnées** permettant de décrire les objets pédagogiques, l'initiative LOM est devenue un standard de fait largement répandu dans les viviers de connaissance actuels.
- **Au niveau des services de traitement** associés à ces systèmes de stockage, la

spécification SQI définit certaines opérations de base permettant de facilement accéder aux contenus pédagogiques renfermés dans les viviers. Sa simplicité contraste avec la complexité de l'initiative IMS-DRI que peu de systèmes ont implémentée.

- **Au niveau des plates-formes d'apprentissage**, la norme SCORM permet le déploiement de ressources sur différents systèmes, alors que la prometteuse initiative IMS-LD permet d'intégrer de véritables scénarios pédagogiques à ces contenus d'apprentissage.

L'ensemble de ces normes permet de satisfaire certains besoins exprimés dans la section 2.4, nous nous attacherons donc à les prendre en compte dans la solution que nous proposons pour atteindre la virtualisation des objets pédagogiques.

Chapitre 4

Les informations disponibles pour l'observation des ressources pédagogiques

Sommaire

4.1 Introduction **51**
4.2 Informations communes aux systèmes d'apprentissage **52**
4.3 Informations spécifiques aux viviers de connaissance **53**
4.4 Informations spécifiques aux plates-formes d'apprentissage **54**
4.5 Synthèse **54**

4.1 Introduction

Dès 1929, H.G. Wells [LP05] prédisait que « pour former un citoyen efficace du monde complexe qui est en train de se développer, il est autant nécessaire de savoir calculer, penser en moyenne, maximum et minimum, qu'il est impératif aujourd'hui de savoir lire et écrire. »[1]

L'intérêt pour l'observation, l'instrumentation et l'évaluation des systèmes éducatifs en ligne est de plus en plus fort au sein des communautés EIAH. Connaître les actions effectuées par les utilisateurs (apprenants et/ou enseignants), leurs préférences ou habitudes dans le but de leur offrir des environnements de travail adaptés à leurs besoins constituent des enjeux majeurs aujourd'hui, comme le montre le nombre d'appel à soumission ou de projets relatifs à ce sujet.

Nos travaux s'intéressent plus spécifiquement à l'observation de l'usage des objets pédagogiques et aux informations statistiques associées qui permettraient de fournir des méthodes de recherche avancées. Dans ce chapitre, nous décrivons les informations qui sont rendues disponibles par les systèmes d'apprentissage en ligne actuels, en nous intéressant particulièrement

[1] Traduit de l'anglais

aux viviers de connaissance et plates-formes de téléformation. Nous exposons dans un premier temps les informations communes à ces deux types de système avant de présenter celles qui sont plus spécifiques à chacun d'entre eux.

4.2 Informations communes aux systèmes d'apprentissage

L'apprentissage en ligne est très souvent basé sur le paradigme Client/Serveur : le système d'apprentissage est hébergé sur un logiciel serveur d'une machine distante, et les enseignants/apprenants utilisent un logiciel client tel qu'un navigateur Internet pour effectuer leurs formations. Aujourd'hui, la plupart des serveurs *web* tels que le logiciel Apache[2] propose un moyen de recueillir des informations concernant les pages, images ou fichiers qui ont fait l'objet d'une requête HTTP, les utilisateurs qui ont effectué ces requêtes, ou encore le nombre d'octets qui ont été transférés.

Les instructeurs peuvent ainsi déterminer le navigateur et le système d'exploitation utilisés par les apprenants, l'endroit géographique où ils se trouvent, le moment (date et heure) auquel ils ont effectués une certaine activité, la durée qu'ils ont passé sur une certaine page, le nombre de fois qu'une page spécifique a été visitée, ou encore l'adresse IP[3] correspondant à chacun des événements. Ces informations présentent un certain potentiel pour évaluer l'efficacité d'un cursus en ligne ; elles permettent de quantifier les interactions entre les utilisateurs et les pages du cours/cursus et ainsi de répondre à certaines questions comme : les pages *web* du cours sont-elles adaptées au navigateur le plus utilisé par les apprenants ? est-ce que les apprenants accèdent facilement aux pages essentielles du cours ? une page spécifique est-elle réellement nécessaire à ce cours ? le laps de temps passé par les étudiants sur une page particulière est-il adapté pour atteindre l'objectif visé ? etc.

Toutefois, les informations disponibles ne sont pas très précises. Les données produites par les applications s'exécutant sur le poste client (telles que les applets Java™ ou les fichiers Macromedia Flash™ qui génèrent des contenus multimédias plus riches que des applications "ordinaires") ou celles issues d'applications produisant du contenu dynamique (comme PHP[4] ou Perl[5] qui est basé sur la technologie CGI[6]) ne sont pas recueillies, empêchant ainsi de connaître les informations réellement consultées par les utilisateurs [FR04]. Pourtant, de plus en plus de contenus pédagogiques s'appuient sur ces technologies, ce qui limite les informations pertinentes, utiles et nécessaires à l'observation de l'usage des ressources d'apprentissage.

[2]http://www.apache.org
[3]Internet Protocol
[4]PHP : Hypertext Preprocessor
[5]Pratical Extraction and Report Language
[6]Common Gateway Interface

4.3 Informations spécifiques aux viviers de connaissance

Au niveau des viviers de connaissance, les travaux entrepris concernent surtout l'utilisation des métadonnées, c'est-à-dire l'étude des différents descripteurs employés par les indexeurs d'objets pédagogiques ou les valeurs assignées à ces différents éléments [Fri04] afin d'améliorer les standards de métadonnées existants. Des analyses empiriques [NTD04] permettent de répondre à certaines questions comme :

- Quels éléments de métadonnées sont utilisés au sein de viviers de connaissance ?
- Quelles valeurs sont prédominantes pour ces descripteurs ?
- Quels types ou formats de fichier sont les plus courants ?
- etc.

Des mécanismes spécifiques à un système particulier peuvent être déployés afin de récolter des informations statistiques relatives à l'utilisation des ressources. Dans le cas du KPS[7] de la fondation ARIADNE par exemple, chaque interaction d'un utilisateur avec le système est enregistrée dans un fichier au format XML. Chaque enregistrement donne des informations intéressantes telles que le titre des ressources ayant été indexées, téléchargées ou supprimées, l'utilisateur à l'origine de l'enregistrement, ainsi que la date de création de l'enregistrement.
Cependant, des opérations complexes comme le "nettoyage" du fichier, sa décomposition et la mise en relation des différentes informations doivent être effectuées pour calculer les statistiques liées à un objet pédagogique particulier. De plus, ce fichier est propre à chaque LKP[8] constituant le KPS global (cf. section 3.3.1.1) ; des efforts supplémentaires sont donc nécessaires pour obtenir une vision générale de l'utilisation des objets pédagogiques au sein du vivier.

Enfin, d'autres travaux [LG04] suggèrent des extensions aux standards de métadonnées actuels afin de pouvoir sélectionner des ressources appropriées à un apprenant et à son contexte d'apprentissage. Ces extensions sont basées sur des informations relatives à l'objet pédagogique lui-même (objectif pédagogique, prérequis), à l'apprenant (vitesse de lecture, expérience dans le domaine visé, type de présentation préféré) et à l'historique d'utilisation de l'objet pédagogique (opinion des apprenants ayant utilisé cette ressource, combien de fois cette ressource a été utilisée pour tout type d'apprenant, combien de fois cette ressource a été utilisée pour une certaine catégorie d'apprenant). Même si ces descripteurs additionnels offrent des informations pertinentes pour la sélection d'objets pédagogiques, ils ne sont pas (encore) pris en compte par les standards prédominants de l'e-formation et ne peuvent donc pas être exploités dans le cadre d'un EIAH faisant intervenir un grand nombre de viviers de connaissance.

[7] Knowledge Pool System
[8] Local Knowledge Pool

4.4 Informations spécifiques aux plates-formes d'apprentissage

L'apprentissage en ligne nécessite l'obtention d'informations qui sont spécifiques aux activités pédagogiques comme les résultats des étudiants pour un devoir donné ou le nombre de bonnes réponses des apprenants à un questionnaire à choix multiples afin de leur proposer des scénarios correspondant à leurs résultats. Si l'on considère la vision de Hall mentionnée section 3.4.1 quant à une plate-forme d'apprentissage, un LMS, par définition, « recueille les données des apprenants ».

L'observation de l'utilisation des plates-formes d'apprentissage se focalise donc sur la surveillance des activités des apprenants afin d'améliorer les cours et cursus qui leurs sont proposés ; les modèles existants comme SCORM ou IMS-LD contribuent également à l'intégration de ces activités au sein d'un scénario pédagogique, mais les fonctionnalités mises en oeuvre n'ont pas pour objectif de mesurer l'usage des ressources pédagogiques par les concepteurs de cursus. Les informations relatives aux ressources pédagogiques permettent seulement de connaître celles qui sont déployées dans un cursus d'apprentissage particulier.

4.5 Synthèse

Ce chapitre a permis de donner un aperçu des différentes informations disponibles au sein des systèmes d'apprentissage. Nous constatons que les logiciels serveur offrent aujourd'hui un ensemble de données qui peuvent être utilisées pour améliorer le caractère pédagogique d'un cursus, mais qui ne sont pas suffisantes pour produire des statistiques propres à l'utilisation des objets pédagogiques. Dans le cas des viviers de connaissance, les centres d'intérêt portent plus particulièrement sur les métadonnées décrivant les ressources pédagogiques que sur leur exploitation par les utilisateurs, même si certains systèmes proposent d'enregistrer les interactions utilisateurs/système dans un fichier spécifique. Les informations exploitées dans le cadre d'une plate-forme d'apprentissage se focalisent sur les activités des apprenants afin d'améliorer les scénarios pédagogiques, mais aucune donnée statistique concernant les ressources déployées au sein des cursus n'est disponible.

Chapitre 5
Les verrous à lever

Sommaire

5.1 Intégrer les solutions hétérogènes existantes **55**
5.2 Fournir un mécanisme d'authentification **56**
5.3 Offrir des services d'exploitation **56**
5.4 Superviser les interactions entre composants de l'EIAH **58**

Le chapitre 3 nous a permis de mettre en avant les différents besoins satisfaits par les divers efforts de normalisation pour atteindre la virtualisation des ressources pédagogiques, c'est-à-dire la fédération de ressources pédagogiques réparties dans différents viviers et leur accessibilité à travers différentes plates-formes d'apprentissage. Dans le chapitre précédent, nous avons exposé les différentes informations rendues disponibles par les viviers de connaissance et les plates-formes d'apprentissage. Ce chapitre a pour objectif de mettre en avant les barrières qu'il reste à franchir pour atteindre nos objectifs de virtualisation et de collecte d'informations statistiques.

5.1 Intégrer les solutions hétérogènes existantes

Nous nous intéressons donc à deux types de systèmes : d'une part, les viviers de connaissance permettant le stockage de ressources numériques ainsi que des métadonnées qui leur sont associées, et d'autre part les plates-formes d'apprentissage capables d'exploiter de tels documents. Notre solution devra prendre en compte le caractère distribué et hétérogène de ces composants, et ne devra pas être liée à un vivier ou plate-forme spécifique afin d'être adaptable à divers systèmes. En effet, des solutions existantes telles que celles proposées dans ces travaux [VRD+03] ou [Ghi04] proposent l'interconnexion entre une plate-forme de téléformation et un vivier de connaissance (respectivement Blackboard[1] avec le Knowledge Pool System à partir des *Building Block* propres à ce LMS, et .LRN[2] avec le Learning Object Repository System),

[1] http://www.blackboard.com/us/index.aspx
[2] http://dotlrn.org/

mais ces solutions ne sont valables que pour un type de LMS et LOR particuliers, et ne peuvent pas être déployées sur d'autres dispositifs d'apprentissage. Un des points essentiels de nos études sur la virtualisation des objets pédagogiques est le caractère générique de notre proposition qui encapsule l'hétérogénéité cohabitante et persistante des solutions existantes.

5.2 Fournir un mécanisme d'authentification

Nous avons vu dans le chapitre 3 que les composants d'un EIAH sont souvent munis d'un service permettant de gérer l'authentification des utilisateurs. Si la recherche d'objets pédagogiques dans les viviers de connaissance ne nécessite généralement pas d'authentification (dans le cas contraire, un compte générique est souvent disponible afin de permettre à n'importe quel internaute de parcourir le contenu du vivier), la fonctionnalité d'insertion de nouveaux objets au sein de ces viviers requiert toujours une authentification préalable. Au niveau des plates-formes d'apprentissage, ce mécanisme est indispensable pour permettre aux systèmes d'identifier le rôle de l'utilisateur au sein d'une formation et de lui offrir les services d'apprentissage correspondant à ses fonctions.

Notre objectif étant de faire communiquer les viviers de connaissances et les plates-formes d'apprentissage, nous devons fournir un mécanisme d'authentification global capable d'identifier non seulement les différents systèmes coopérant au sein de notre EIAH, mais également les multiples utilisateurs interagissant avec ces composants.
Ce service doit être transparent à l'utilisateur final afin que celui-ci ait le sentiment de n'utiliser qu'un système unique. En considérant que les utilisateurs exploitent les plates-formes d'apprentissage pour atteindre des ressources externes stockées dans différents viviers de connaissance, alors notre architecture ne devra pas leur imposer une nouvelle authentification relative à leur rôle vis-à-vis de ces systèmes de stockage. Ce service d'authentification est appelé *single sign-on* en anglais.

5.3 Offrir des services d'exploitation

Un EIAH doit permettre aux utilisateurs finaux d'exploiter les objets pédagogiques localisés dans les différents composants intégrés dans l'environnement d'apprentissage. Ainsi, des services tels que la recherche de ressources indexées dans les viviers de connaissance, le téléchargement sur le poste local (dans le cas des viviers de documents) ou la consultation (dans le cas des viviers de références) d'objets existants, l'importation de ces mêmes documents ou références (URL), ou l'indexation de nouvelles ressources permettraient aux utilisateurs de n'ex-

ploiter qu'un système unique, à savoir les plates-formes d'apprentissage.

De nombreux viviers de connaissance existent aujourd'hui (cf. section 3.3), donc le service de recherche d'objets pédagogiques existants doit d'une part, offrir l'opportunité d'exploiter le plus grand nombre de ces systèmes afin de satisfaire les préférences linguistiques et culturelles de chacun et d'autre part, permettre la consultation des métadonnées associées à chacune de ces ressources d'apprentissage. Atteindre une masse critique d'objets pédagogiques favorisera alors l'utilisation des environnements d'apprentissage par tous [Gla02].

Une fois que les ressources satisfaisant le cursus visé sont localisées, le service de téléchargement/consultation permettra aux concepteurs de cours de visualiser le document cible afin qu'ils puissent s'assurer de la qualité et de la viabilité de son contenu avant de l'importer au sein de leurs cursus. Même si les métadonnées renseignent la nature et le contenu du document (descripteurs "discipline", "concepts principaux", "type de science", etc.), elles ne suffisent généralement pas pour se faire une idée précise de la valeur et de la particularité de l'objet d'apprentissage. Ce service offre donc un moyen de vérifier le caractère pédagogique de la ressource d'apprentissage. Eventuellement, l'élaborateur du cours pourra modifier le document à sa guise afin d'être en adéquation avec les apprenants et la formation cibles ; dans ce cas, le concepteur aura produit un nouvel objet pédagogique qui devra être indexé dans un vivier de connaissance afin d'en faire bénéficier les autres utilisateurs de l'environnement d'apprentissage.

Lorsqu'un concepteur a repéré un objet pédagogique répondant aux objectifs de son cours, un service devra lui permettre d'importer l'objet cible au sein de ce cursus. Les LMS disposent d'un espace de stockage permettant d'accueillir les différentes ressources des multiples utilisateurs (cf. section 3.4.2), et proposent un moyen pour transférer un fichier numérique d'un poste client vers l'espace dédié du serveur. Deux solutions sont alors envisageables pour réaliser l'importation de ressources stockées dans des viviers externes :

1. La première se déroule en deux étapes et consiste d'abord à utiliser le service de téléchargement présenté ci-dessus afin de récupérer le document sur le poste local, puis à exploiter le service de transfert de fichiers fourni par la plate-forme cible.
2. La deuxième ne comporte qu'une étape unique et consiste à transférer directement le document du vivier de connaissance vers l'espace de stockage de la plate-forme.

Bien évidemment nous nous attacherons à réaliser la seconde méthode dans la mesure où celle-ci ne présente qu'une seule étape, évitant ainsi des manipulations supplémentaires et ennuyeuses pour les utilisateurs finaux et réduisant au maximum la charge du réseau supportant l'infrastructure globale de l'EIAH.

Enfin, le dernier service que nous souhaitons mettre en oeuvre a pour objectif d'optimiser

le partage et la réutilisation de nouvelles ressources pédagogiques. Lorsqu'un auteur a créé un nouvel objet dans le but de l'intégrer au sein de sa formation, nous voulons offrir la possibilité d'insérer cette ressource non seulement au sein de la plate-forme d'apprentissage (fonctionnalité déjà disponible dans la plupart des LMS), mais aussi et surtout au sein d'un vivier de connaissance afin que les utilisateurs enregistrés dans d'autres plates-formes puissent eux aussi bénéficier de cette nouvelle ressource. Ce service implique donc la création de métadonnées, puisque les viviers de connaissance stockent à la fois les documents et des descripteurs associés au contenu de ce document (cf. section 3.3). La tâche qui consiste à renseigner les métadonnées est souvent fastidieuse, aussi nous proposerons un service complémentaire de génération automatique de métadonnées afin de faciliter et d'encourager l'indexation de nouvelles ressources.

5.4 Superviser les interactions entre composants de l'EIAH

Dans le chapitre 4, nous avons exposé les différentes informations qui sont disponibles au sein des logiciels serveurs hébergeant les systèmes d'apprentissage et qui peuvent être exploitées dans le cadre de nos objectifs de supervision. Outre le problème lié aux contenus dynamiques générés par le logiciel client, des barrières subsistent quant à l'accès et à la signification de ces informations :

- **Elles sont renfermées dans des fichiers** qui ne sont généralement pas disponibles à n'importe quel utilisateur ; les administrateurs Réseau et Système sont de plus en plus méfiants et n'autorisent que très rarement l'accès à ce type d'information. De plus, l'acquisition de ces données doit être effectuée selon un intervalle de temps déterminé par le concepteur de l'outil de supervision, elle n'est pas réalisée en temps réel.
- **Elles ne transcrivent pas nécessairement l'exploitation réelle** du système par les utilisateurs : un grand nombre de moteurs de recherche tels que Google[3] ou Altavista[4] sont basés sur des robots[5] [Eic94] [Fie94] qui analysent le contenu d'un logiciel serveur afin d'en extraire les informations pertinentes. Même si certains systèmes différencient les visites effectuées par les robots de celles effectuées par les utilisateurs, la majorité d'entre eux accumulent le nombre de visites, ce qui ne rend pas compte de l'usage véritable du système d'apprentissage.
- Enfin, **elles ne sont liées qu'à l'utilisation d'un outil unique**. Pourtant il n'est pas rare aujourd'hui qu'un même utilisateur, enseignant ou apprenant, interagisse avec différents systèmes d'apprentissage.

[3] http://www.google.com
[4] http://www.altavista.com
[5] *spider* en anglais

D'autre part, les informations spécifiques à l'exploitation des viviers de connaissance ou des plates-formes d'apprentissage ne correspondent pas aux données que nous souhaitons obtenir. Dans le cas des viviers, les indications fournies concernent les descripteurs de métadonnées, alors que le suivi des activités ou performances des apprenants représente le principal centre d'intérêt dans le cas des plates-formes d'apprentissage.

Dans l'objectif de proposer aux concepteurs de cursus d'apprentissage une recherche d'objets pédagogiques basée sur des données statistiques traduisant l'exploitation des ressources pédagogiques par les différents utilisateurs, nous devons proposer un outil d'observation capable de recueillir et d'enregistrer les informations issues des diverses entités impliquées dans l'EIAH. Il est difficile d'envisager une mise en relation des différentes informations collectées par les systèmes actuels : d'une part les informations recueillies par les viviers de connaissance et les plates-formes d'apprentissage ne sont pas de même nature, visent des objectifs différents et peuvent poser des problèmes d'unification et d'homogénéité, et d'autre part ces données ne sont pas suffisantes pour atteindre nos objectifs de supervision. Nous proposons donc dans le chapitre 7 une alternative à ces deux techniques basée sur une approche de modélisation de l'ensemble des systèmes et utilisateurs mis en jeu au sein d'un EIAH.

Deuxième partie

Virtualisation de ressources pédagogiques et supervision de l'EIAH

Chapitre 6

Virtualisation de ressources pédagogiques

Sommaire

6.1 Architecture de virtualisation **63**

6.2 Couche de Fédération **64**

6.2.1 Service de recherche des objets pédagogiques 66

6.2.2 Services de téléchargement et de consultation 71

6.2.3 Services d'importation 72

6.2.4 Service d'indexation 74

6.2.5 Service de génération automatique de métadonnées 75

6.2.6 Service d'authentification 78

6.3 Couche de Mutualisation **81**

6.3.1 Lien entre les plates-formes de gestion d'apprentissage et la couche de Fédération 81

6.3.2 Fournisseur d'informations 82

6.4 Synthèse **82**

Nous proposons dans ce chapitre une architecture structurée en couches, technique de décomposition largement utilisée dans le développement logiciel pour augmenter la réutilisation et l'interopérabilité des composants. Dans la mesure où les couches Stockage des ressources et Interfaces Pédagogiques ont respectivement été détaillées dans les parties 3.3 et 3.4, nous détaillons dans la suite du chapitre les couches de Fédération et de Mutualisation permettant la mise en œuvre de la virtualisation des objets pédagogiques.

6.1 Architecture de virtualisation

Nous avons introduit un niveau supplémentaire à l'architecture 3 tiers normalisée illustrée par la figure 2.3, ce qui résulte en une architecture globale divisée en quatre niveaux illustrés sur la figure 6.1 [BV05] [BVBD05].

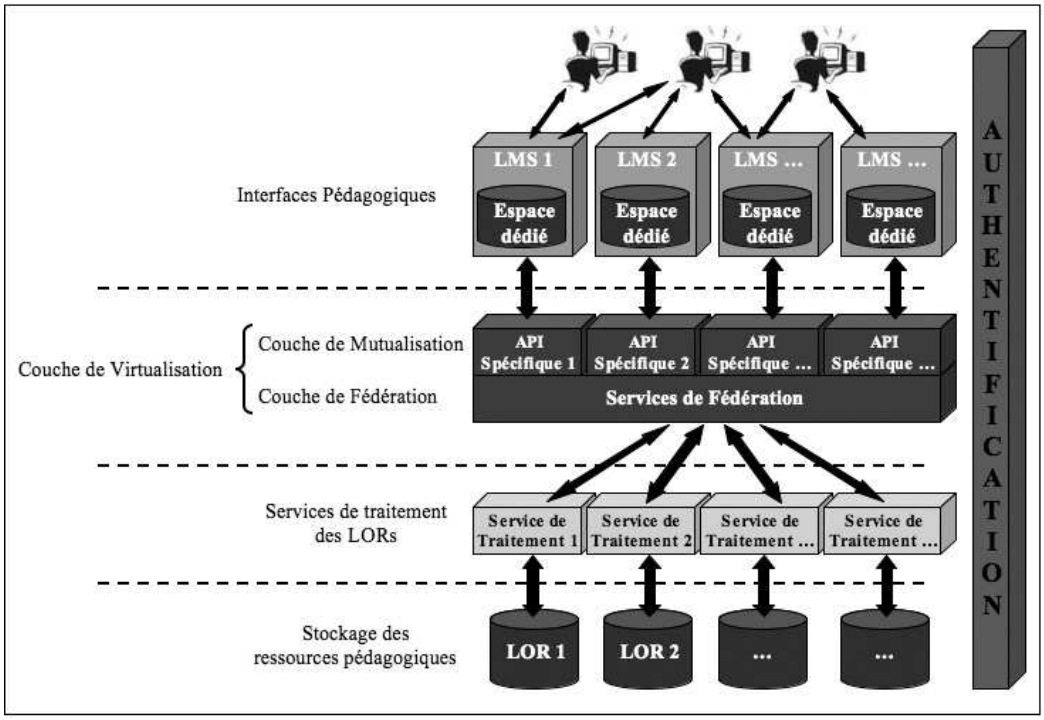


FIG. 6.1 – Une architecture pour la virtualisation de ressources pédagogiques

La couche de Virtualisation, divisée en deux sous-couches, offre une virtualisation des objets pédagogiques : la couche de Fédération permet de retrouver et d'exploiter les objets pédagogiques stockés dans des viviers de connaissances hétérogènes en invoquant les services rendus par la couche Services de Traitement, alors que la couche de Mutualisation a pour objectif de rendre facilement disponibles les services de la couche de Fédération aux systèmes inclus dans la couche Interfaces Pédagogiques. L'accès à ces services doit être transparent pour l'utilisateur final, tandis que l'interface de fédération des ressources doit être claire et simple d'utilisation afin de masquer la complexité du système global.

Nous avons élaboré la couche de Virtualisation de manière à ce qu'elle soit le plus générique possible, c'est-à-dire indépendante de la nature des plates-formes d'apprentissage et des systèmes de stockage participant à la construction de l'EIAH.

6.2 Couche de Fédération

Comme nous l'avons évoqué dans la section 3.4, la majorité des LMS stocke les ressources pédagogiques au sein d'un espace disque dédié, la plupart du temps sans métadonnées associées, ou du moins sans métadonnées permettant à l'utilisateur final d'identifier du matériel

pertinent à son contexte. En section 3.3, nous avons présenté l'intérêt des viviers de connaissance qui permettent le stockage d'objets pédagogiques et de leurs métadonnées tout en fournissant un environnement transparent pour partager et réutiliser ces ressources. Toutefois, ces objets pédagogiques stockés dans des systèmes isolés ne sont pas utiles à l'apprentissage s'ils ne peuvent pas être déployés sur des plates-formes d'enseignement. Aujourd'hui, la plupart des institutions publiques et privées possèdent un système similaire à des plates-formes d'apprentissage [Pau03], alors certaines fonctionnalités ou services doivent leur permettre d'interagir avec les différents viviers de connaissance existants de façon transparente (cf. section 5.3) :

- **Un service de recherche** doit permettre de retrouver des objets pédagogiques stockés dans différents viviers.
- **Un service de téléchargement** (dans le cas des LOR) ou de **consultation** (dans le cas des LOF) des ressources retrouvées par le service de recherche.
- **Un service d'importation** des objets pédagogiques renfermés dans les viviers au sein des cursus pédagogiques des LMS.
- **Un service d'indexation** de nouveaux objets pédagogiques au sein d'un LOR particulier.
- **Un service de génération automatique de métadonnées** visant à faciliter le processus d'indexation de nouvelles ressources.
- **Un service d'authentification** capable d'identifier les composants de l'EIAH ainsi que les utilisateurs.

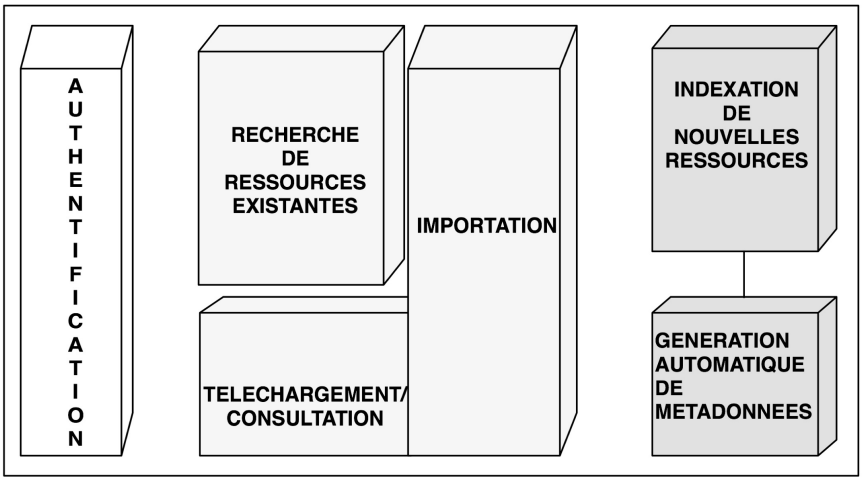


Fig. 6.2 – Les services rendus par la couche de Fédération

La couche de Fédération, basée sur une architecture orientée services, a pour objectif de fournir cet ensemble de services représenté par la figure 6.2. Ce type d'architecture est en plein essor actuellement dans le domaine de l'apprentissage en ligne [DCR05], comme le montrent les projets Sakai[1] ou JISC[2] e-Learning Framework[3]. Les architectures SOA permettent une ouverture des plates-formes qui restaient jusqu'ici cloisonnées (cf. section 3.4), mais offrent également une modularité des fonctionnalités et composants offerts qui renforce ainsi la généricité et la flexibilité de notre architecture globale.

6.2.1 Service de recherche des objets pédagogiques

Le service de recherche des objets pédagogiques a pour but d'offrir une vue unique de l'ensemble des ressources stockées dans différents LOR afin de favoriser l'échange et la réutilisation par tous de ces ressources pédagogiques. C'est ce service qui permet de réaliser la première étape de la virtualisation des ressources pédagogiques [VB05].

6.2.1.1 Obtention d'un objet pédagogique

Nous décrivons dans la figure 6.3 le processus d'obtention d'un objet pédagogique qui consiste à :

1. **Chercher et évaluer les métadonnées** : sélectionner un objet pédagogique qui satisfait la requête de l'utilisateur. Cette étape peut être répétée dans le but de raffiner la recherche et de trouver les objets pédagogiques appropriés.
2. **Localiser l'objet pédagogique** : selon la nature du vivier de connaissance (LOR ou LOF), la localisation de l'objet est respectivement indiquée par un identifiant utilisable seulement dans le contexte du vivier, ou par une URL.
3. **Consommer l'objet pédagogique** : obtenir l'objet pédagogique à partir de sa localisation.

6.2.1.2 Scénario de recherche d'un objet pédagogique

Pour réaliser les première et deuxième étapes du processus d'obtention d'un objet pédagogique, le service de recherche exploite les services de traitement associés à chacun des viviers de connaissance fédérés. L'ensemble des méthodes implémentées permet d'obtenir les métadonnées ainsi que la localisation des objets pédagogiques sous la forme d'un champ de

[1] http://www.sakaiproject.org/
[2] Joint Information Systems Committee
[3] http://www.jisc.ac.uk/index.cfm ?name=elearning_framework

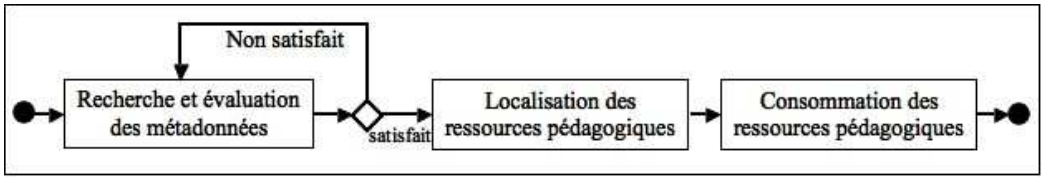


FIG. 6.3 – Diagramme d'activités du processus d'obtention d'objets pédagogiques [Mas05]

métadonnées (identifiant ou URL). Nous ne pouvons décrire ici les méthodes de chaque service de traitement spécifique à un vivier particulier, mais nous détaillons les différentes étapes ainsi que les méthodes associées qui permettent d'interagir avec un service de traitement conforme à la spécification SQI (cf. section 3.3.3).

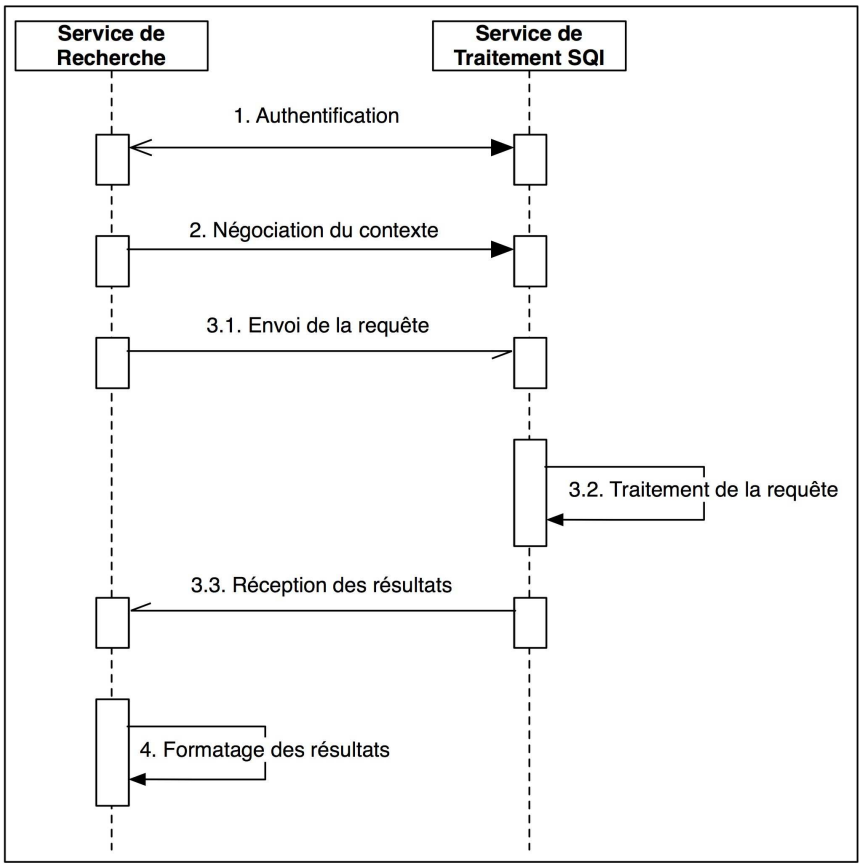


FIG. 6.4 – Scénario de recherche des ressources pédagogiques

La figure 6.4 illustre le diagramme de séquence UML correspondant aux différentes phases nécessaires à la recherche et à la localisation des objets pédagogiques à travers un service de traitement SQI :

1. **Authentification** : le service de recherche invoque la méthode createAnonymousSession du service de traitement SQI associé au vivier cible qui retourne un identifiant de session sous forme d'une chaîne de caractères. Cet identifiant est un élément obligatoire pour identifier les systèmes dans toutes les communications sous-jacentes.
2. **Négociation du contexte** : le service de recherche, en invoquant les méthodes setQueryLanguage et setResultsFormat, définit respectivement le langage de requête qui sera utilisé lors des communications sous-jacentes ainsi que le format de résultat correspondant au modèle de métadonnées décrivant les ressources pédagogiques.
3. **Envoi de la requête et réception des résultats** : dans un premier temps, le service de recherche construit la requête correspondant aux critères de recherche puis invoque la méthode synchronousQuery du service de traitement SQI cible qui effectuera la recherche des métadonnées au sein du LOR fédéré. Les métadonnées des objets pédagogiques satisfaisant les critères de la requête sont retournées au service de recherche sous la forme d'une chaîne de caractères.
4. **Formatage des résultats** : lorsque le service de recherche obtient les résultats de la requête, un mécanisme de mise en forme des métadonnées est effectué.

6.2.1.3 Spécifications des méthodes du service de recherche

Méthode createAnonymousSession

L'interface SQI repose sur un concept de gestion de session : une session doit être établie entre le service demandeur et le service de traitement SQI cible avant que toute interaction puisse prendre place. La méthode createAnonymousSession crée une session ne nécessitant aucun compte sur le système cible et retourne l'identifiant de cette session sous la forme d'une chaîne de caractères. Sa spécification est donnée par le tableau 6.1.

Nom de la méthode	**createAnonymousSession**
Type de retour	String
Erreur	METHOD_FAILURE

TAB. 6.1 – Spécification de la méthode createAnonymousSession [SMA+05]

La méthode createAnonymousSession peut générer l'erreur METHOD_FAILURE si la création de la session a échoué.

Méthode setQueryLanguage

La méthode setQueryLanguage dont la spécification apparaît dans le tableau 6.2 permet

au service demandeur de contrôler la syntaxe des requêtes sous-jacentes en définissant le langage de celles-ci à l'aide du paramètre queryLanguageID. Un exemple de valeur pour ce paramètre est "VSQL"[4] qui, comme son nom l'indique, spécifie un langage simple de requête. Le paramètre targetSessionID permet d'identifier la session en cours sur le service SQI cible.

Nom de la méthode	**setQueryLanguage**
Type de retour	Void
Paramètres	*Name* / *Type* targetSessionID / String queryLanguageID / String
Erreurs	NO_SUCH_SESSION QUERY_LANGUAGE_NOT_SUPPORTED METHOD_FAILURE

TAB. 6.2 – Spécification de la méthode setQueryLanguage [SMA+05]

La méthode setQueryLanguage peut générer les erreurs suivantes :

- **NO_SUCH_SESSION** si l'identifiant de session n'est pas valide.
- **QUERY_LANGUAGE_NOT_SUPPORTED** dans le cas où le langage de requête spécifié est inconnu du service de traitement SQI cible.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

Méthode setResultsFormat

La méthode setResultsFormat du tableau 6.3 permet au service demandeur de définir le format de résultats retourné par le service de traitement SQI cible lors de l'invocation des méthodes synchronousQuery ou asynchronousQuery. Ce format de résultats, spécifié par le paramètre resultsFormat, peut par exemple prendre "lom" pour valeur.

La méthode setResultsFormat peut générer les erreurs suivantes :

- **NO_SUCH_SESSION** si l'identifiant de session n'est pas valide.
- **RESULTS_FORMAT_NOT_SUPPORTED** lorsque le format de résultats spécifié n'est pas supporté par le service de traitement SQI cible.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

[4] Very Simple Query Language

Nom de la méthode	**setResultsFormat**
Type de retour	Void
Paramètres	<table><tr><th>*Name*</th><th>*Type*</th></tr><tr><td>targetSessionID</td><td>String</td></tr><tr><td>resultsFormat</td><td>String</td></tr></table>
Erreurs	NO_SUCH_SESSION RESULTS_FORMAT_NOT_SUPPORTED METHOD_FAILURE

TAB. 6.3 – Spécification de la méthode setResultsFormat [SMA+05]

Méthode synchronousQuery

La méthode synchronousQuery permet au service demandeur d'envoyer une requête définie par le paramètre queryStatement au service de traitement SQI cible. Le paramètre startResult permet de déterminer le numéro du premier résultat attendu dans l'ensemble des résultats retournés, et qui correspondent aux critères de recherche.

Nom de la méthode	**synchronousQuery**
Type de retour	String
Paramètres	<table><tr><th>*Name*</th><th>*Type*</th></tr><tr><td>targetSessionID</td><td>String</td></tr><tr><td>queryStatement</td><td>String</td></tr><tr><td>startResult</td><td>Integer</td></tr></table>
Erreurs	NO_SUCH_SESSION INVALID_QUERY_STATEMENT QUERY_MODE_NOT_SUPPORTED INVALID_START_RESULT NO_MORE_RESULT METHOD_FAILURE

TAB. 6.4 – Spécification de la méthode synchronousQuery [SMA+05]

La méthode synchronousQuery peut générer les erreurs suivantes :

- **NO_SUCH_SESSION** si l'identifiant de session n'est pas valide.
- **INVALID_QUERY_STATEMENT** si le format de la requête n'est pas conforme au langage de requête défini lors de l'appel de la méthode setQueryLanguage.
- **QUERY_MODE_NOT_SUPPORTED** dans le cas où le service de traitement SQI

cible ne supporte pas le mode de requête synchrone.

- **INVALID_START_RESULT** lorsque le nombre entier startResult est invalide.
- **NO_MORE_RESULT** si startResult a pour valeur 0 et qu'aucun résultat n'est disponible.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

Lorsque l'utilisateur dispose d'une vue unique des ressources stockées dans différents viviers, des services doivent lui être offerts afin qu'il puisse consommer le document correspondant, c'est-à-dire le télécharger ou l'intégrer au sein de la plate-forme d'apprentissage.

6.2.2 Services de téléchargement et de consultation

Le service de recherche nous permet d'obtenir les métadonnées associées à un document pédagogique particulier. Dans la mesure où l'une d'entre elles correspond à la localisation des ressources pédagogiques, un service de téléchargement a été mis en œuvre afin que l'utilisateur final puisse visualiser le contenu de l'objet avant de l'intégrer dans un cursus ou de l'adapter à son propre contexte d'apprentissage.

Dans le cas d'un vivier de références, le service de téléchargement permet de visualiser l'URL associée à l'objet pédagogique et définie par le paramètre url. La spécification de la méthode associée previewUrl figure dans le tableau 6.5.

Nom de la méthode	**previewUrl**
Type de retour	Void
Paramètres	*Name* : url ; *Type* : String
Erreurs	NO_SUCH_URL METHOD_FAILURE

TAB. 6.5 – Spécification de la méthode previewUrl

La méthode previewUrl peut générer les erreurs suivantes :

- **NO_SUCH_URL** dans le cas où l'URL associée au document n'existe pas.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

Dans le cas d'un vivier de documents, la ressource numérique identifiée par la paramètre docIdentifier est véritablement importée sur le poste local de l'utilisateur grâce au service de

traitement spécifique au vivier cible. Nous avons donc suggéré la méthode downloadDocument dont la spécification est présentée dans le tableau 6.6 pour assurer cette fonctionnalité. Dans la mesure où ce service nécessite une authentification, les paramètres login et password, obtenus à l'aide du service d'authentification exposé dans la section 6.2.6, permettent de spécifier les caractéristiques du compte de l'utilisateur courant vis-à-vis du LOR cible.

Nom de la méthode	**downloadDocument**
Type de retour	String
Paramètres	*Name* / *Type* docIdentifier / String login / String password / String
Erreurs	NO_SUCH_DOC_IDENTIFIER INVALID_ACCOUNT METHOD_FAILURE

TAB. 6.6 – Spécification de la méthode downloadDocument

La méthode downloadDocument peut générer les erreurs suivantes :

- **NO_SUCH_DOC_IDENTIFIER** dans le cas où l'identifiant du document n'existe pas dans le LOR.
- **INVALID_ACCOUNT** si le login et/ou le password sont inconnus du service de traitement.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

6.2.3 Services d'importation

Le service d'importation permet d'intégrer un objet pédagogique au sein de la plate-forme d'apprentissage cible suite à l'obtention des métadonnées par le service de recherche. Ce service est fortement lié à la plate-forme d'apprentissage puisque la couche de Fédération doit connaître le répertoire destiné au stockage de la ressource pédagogique ; l'identification de cette propriété est déterminée à partir des informations fournies par la couche de Mutualisation qui sont détaillées dans la section 6.3.2.

Dans le cas d'un vivier de références, l'URL associée à l'objet pédagogique est attribuée à la valeur de la localisation de la ressource au sein du LMS à travers l'exécution de la méthode importUrl présentée dans le tableau 6.7. Le paramètre fieldToFill détermine le champ du formulaire au sein de la plate-forme qui contiendra la valeur du paramètre url.

Nom de la méthode	**importUrl**
Type de retour	Void
Paramètres	
	Name / *Type* url / String fieldToFill / String
Erreurs	NO_SUCH_URL INVALID_FIELD_NAME METHOD_FAILURE

TAB. 6.7 – Spécification de la méthode importUrl

La méthode importUrl peut générer les erreurs suivantes :

- **NO_SUCH_URL** dans le cas où l'URL associée au document n'existe pas.
- **INVALID_FIELD_NAME** si le champ mentionné n'existe pas dans le formulaire de la plate-forme d'apprentissage.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

Dans le cas d'un vivier de documents, la ressource numérique est recopiée au sein de l'espace dédié de la plate-forme d'enseignement cible. L'importation d'un document imite le comportement du service de téléchargement, mis à part que la ressource n'est pas présentée à l'utilisateur final, mais recopiée au sein de la plate-forme d'apprentissage à l'aide de la méthode importDocument du tableau 6.8. Le chemin du répertoire destiné à recevoir le document est déterminé par le paramètre folderLocation.

Nom de la méthode	**importDocument**
Type de retour	Void
Paramètres	
	Name / *Type* docIdentifier / String folderLocation / String login / String password / String
Erreurs	NO_SUCH_DOC_IDENTIFIER INVALID_FOLDER_LOCATION INVALID_ACCOUNT METHOD_FAILURE

TAB. 6.8 – Spécification de la méthode importDocument

La méthode importDocument peut générer les erreurs suivantes :

- **NO_SUCH_DOC_IDENTIFIER** dans le cas où l'identifiant du document n'existe pas dans le LOR.
- **INVALID_FOLDER_LOCATION** dans le cas où le chemin d'accès au répertoire cible de la plate-forme est invalide.
- **INVALID_ACCOUNT** si le login et/ou le password sont inconnus du service de traitement.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

6.2.4 Service d'indexation

Un service supplémentaire de la couche de Fédération permet d'insérer de nouveaux objets pédagogiques à la fois dans l'espace dédié du LMS, mais également dans un vivier de connaissance afin de rendre les ressources accessibles à d'autres plates-formes. Après avoir sélectionné un document résidant sur son disque local, l'utilisateur doit renseigner l'ensemble des métadonnées requises pour l'insertion d'un nouvel objet pédagogique dans le vivier cible à travers un vaste formulaire électronique décrivant le standard de métadonnées implémenté dans ce vivier. Lorsque l'ensemble des champs obligatoires du formulaire sont remplis, la méthode indexLearningObject spécifiée dans le tableau 6.9 invoque le service de traitement responsable de la gestion du vivier cible afin d'insérer les métadonnées (paramètre metadata) et le document (paramètre document) au sein du système. Ce service est également réservé à certains utilisateurs, donc le service d'authentification exposé dans la section 6.2.6 est préalablement invoqué afin d'obtenir les paramètres login et password qui caractérisent le compte de l'utilisateur sur le vivier cible. En cas de succès, cette méthode retourne l'identifiant du nouvel objet pédagogique attribué par le vivier cible sous la forme d'une chaîne de caractères.

La méthode indexLearningObject est susceptible de générer les erreurs suivantes :

- **INVALID_DOCUMENT** dans le cas où la conversion du document en chaînes de caractères a échoué.
- **INVALID_METADATA** si le tableau de chaînes de caractères spécifiant les métadonnées ne contient pas au minimum les descripteurs requis pour l'indexation d'un nouvel objet pédagogique.
- **INVALID_FOLDER_LOCATION** dans le cas où le chemin d'accès au répertoire cible de la plate-forme est invalide.

Nom de la méthode	**indexLearningObject**
Type de retour	String
Paramètres	

Name	*Type*
document	String
metadata	String[]
folderLocation	String
login	String
password	String

Erreurs	INVALID_DOCUMENT INVALID_METADATA INVALID_FOLDER_LOCATION INVALID_ACCOUNT METHOD_FAILURE

TAB. 6.9 – Spécification de la méthode indexLearningObject

- **INVALID_ACCOUNT** si le login et/ou le password sont inconnus du service de traitement.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

Ce processus d'indexation est fastidieux et la majorité des acteurs de l'e-formation ne prennent pas le temps de générer des métadonnées ; pour les autres, des problèmes de consistance et de fiabilité peuvent apparaître [JJBW99]. Pourtant, les objets pédagogiques jouent un rôle prédominant au sein d'un EIAH [MP03]. Alors pour lever ce verrou et favoriser la mise à disposition de nouvelles ressources d'apprentissage, la couche de Fédération offre un service complémentaire au service d'indexation qui permet de générer automatiquement la majorité des métadonnées nécessaires à l'indexation d'un objet pédagogique dans un vivier de connaissance.

6.2.5 Service de génération automatique de métadonnées

Dans l'optique de générer automatiquement les métadonnées les plus importantes lors de l'indexation d'un objet pédagogique, un service de la couche de Fédération permet d'extraire des informations liées au contexte de la plate-forme d'apprentissage et de les faire correspondre à des attributs du modèle de métadonnées implémenté au sein du LOR, réduisant ainsi le nombre de champs à remplir manuellement [BVMD05]. En effet, la plupart des LMS demandent aux créateurs de formations de décrire brièvement le sujet relatif au cursus qu'ils définissent, c'est-à-dire de déterminer un ensemble de mots clés et de termes en relation avec le sujet du cursus. Ces termes représentent des éléments de métadonnées qui peuvent être utilisés pour alléger

le processus d'indexation. La collecte de ces informations est spécifique à chaque plate-forme d'apprentissage, la méthode getContextKeywords (cf. tableau 6.10) présente donc un paramètre unique targetLms qui détermine le LMS cible. Cette méthode permet d'extraire un ensemble de mots clés pertinents comme le nom du cours ou de la formation définis par le concepteur du cursus pédagogique, qui sont ensuite retournés dans un tableau de chaînes de caractères.

Nom de la méthode	**getContextKeywords**
Type de retour	String[]
Paramètres	*Name* / *Type*: targetLms / String
Erreurs	UNKNOWN_LMS METHOD_FAILURE

TAB. 6.10 – Spécification de la méthode getContextKeywords

La méthode getContextKeywords peut générer les erreurs suivantes :

- **UNKNOWN_LMS** dans le cas où la plate-forme d'apprentissage cible est inconnue.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

L'extension d'un fichier numérique permet de déduire de nombreuses informations relatives à son type MIME[5], au type/format du document ou aux systèmes d'exploitation capable d'interpréter ce fichier. Alors nous avons d'une part élaboré une librairie associant aux extensions de fichier les plus courantes ces métadonnées, et d'autre part une méthode getFileInfo exploitant cette librairie et permettant de retrouver les informations précitées à partir d'une extension particulière précisée par le paramètre extension. Cette méthode est spécifiée dans le tableau 6.11.

Nom de la méthode	**getFileInfo**
Type de retour	String[]
Paramètres	*Name* / *Type*: extension / String
Erreurs	UNKNOWN_EXTENSION METHOD_FAILURE

TAB. 6.11 – Spécification de la méthode getFileInfo

La méthode getFileInfo peut générer les erreurs suivantes :

[5]Multipurpose Internet Mail Extensions

- **UNKNOWN_EXTENSION** si l'extension mentionnée n'est pas définie au sein de la librairie.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

La qualité du service de génération automatique de métadonnées dépend fortement de l'adoption d'un standard commun pour déployer les ressources pédagogiques, permettant ainsi aux auteurs d'objets pédagogiques de développer du matériel indépendant des plates-formes. La section 3.4.3 a exposé la norme SCORM qui intègre la spécification IMS Content Packaging (cf. annexe A.3) pour décrire le contenu d'un package SCORM. Nous avons donc introduit la méthode extractScormMetadata dont la spécification apparaît dans le tableau 6.12 au sein du service de génération automatique afin d'extraire l'ensemble des métadonnées conformes au standard LOM (cf. annexe A.4) et renfermées dans le fichier imsmanifest.xml de tout package SCORM.

Nom de la méthode	**extractScormMetadata**
Type de retour	String[]
Paramètres	*Name* | *Type* manifestFile | String
Erreurs	INVALID_MANIFEST_FILE METHOD_FAILURE

TAB. 6.12 – Spécification de la méthode extractScormMetadata

La méthode extractScormMetadata peut générer les erreurs suivantes :

- **INVALID_MANIFEST_FILE** si le fichier imsmanifest.xml n'est pas un fichier XML valide ou s'il ne contient pas de section metadata.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

Les caractéristiques de l'auteur de la ressource SCORM figurent parmi les métadonnées contenues dans le paramètre de retour de la méthode extractScormMetadata. Les propriétés de l'auteur respectent le format *vcard* [HSD98] [DH98], alors la méthode getAuthorInfo définie dans le tableau 6.13 permet d'identifier les différents champs spécifiés dans une *vcard* (paramètre vcard) tels que les nom, prénom, adresse de courrier électronique, téléphone, etc. Ces informations sont contenues dans le paramètre de retour de la méthode.

La méthode getAuthorInfo peut générer les erreurs suivantes :

- **INVALID_VCARD** si la chaîne de caractères ne respecte pas le format standard d'une *vcard*.

Nom de la méthode	**getAuthorInfo**
Type de retour	String[]
Paramètres	*Name* | *Type* vcard | String
Erreurs	INVALID_VCARD METHOD_FAILURE

TAB. 6.13 – Spécification de la méthode getAuthorInfo

- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

6.2.6 Service d'authentification

Les services de téléchargement/consultation, d'importation et d'indexation nécessitent la mise en place d'un service supplémentaire permettant d'authentifier les utilisateurs vis-à-vis du vivier de connaissance cible. En effet, certains LOR n'offrent pas la possibilité d'obtenir l'intégralité des documents pédagogiques librement, comme c'est le cas du Knowledge Pool System (KPS) [DFC+01] de la fondation ARIADNE qui définit des rôles afin de limiter l'accès aux ressources qu'il renferme. Nous devons dans ce cas être capables de connaître les droits de l'utilisateur connecté à la plate-forme d'enseignement vis-à-vis du vivier de connaissance auquel il souhaite accéder.
Une solution intuitive serait de demander à l'usager les caractéristiques du compte qu'il possède sur le vivier, mais cette technique soulève deux inconvénients majeurs :

1. Bien que l'utilisateur soit déjà authentifié sur la plate-forme d'enseignement, la couche de Fédération lui imposerait une seconde authentification et ne respecterait pas la philosophie *single sign on* qui présente un intérêt majeur de simplification.
2. La transparence de l'accès aux viviers de connaissance ne serait plus assurée.

Nous proposons donc un autre mécanisme basé sur l'exploitation des informations de supervision [BV06b] exposées dans le chapitre 7. A partir de l'identité d'une personne, ces données permettent de connaître l'ensemble des comptes de l'utilisateur vis-à-vis des systèmes d'apprentissage avec lesquels il travaille. Le diagramme de séquence UML illustré par la figure 6.5 montre les différentes étapes réalisées par notre système entre le moment où un utilisateur invoque le service de téléchargement/consultation, d'importation ou d'indexation (entité *Service Demandeur* sur la figure 6.5) et celui où il obtient les résultats de l'invocation.

1. **Invocation du service demandeur** (téléchargement/consultation, importation ou indexation) par l'utilisateur de la plate-forme d'apprentissage.
2. Le service demandeur **invoque le service d'authentification** afin d'obtenir les propriétés du compte de l'utilisateur vis-à-vis du vivier de connaissance cible.
3. Le service d'authentification **transmet une requête** au composant de supervision qui est détaillé dans la section 7.4.2 par le biais de la méthode getUserAccount (cf. tableau 6.14).
4. Le composant de supervision **recherche le compte de l'utilisateur** en parcourant les différentes informations de supervision (cf. chapitre 7).
5. Lorsque la recherche est terminée, le composant de supervision **retourne au service d'authentification le *login* et le mot de passe** correspondant.
6. Ce dernier **délivre au service demandeur les informations obtenues**.
7. **Exécution du service demandeur** avec les caractéristiques propres à l'utilisateur.
8. Le **résultat du traitement** du service demandeur est retourné à l'utilisateur.

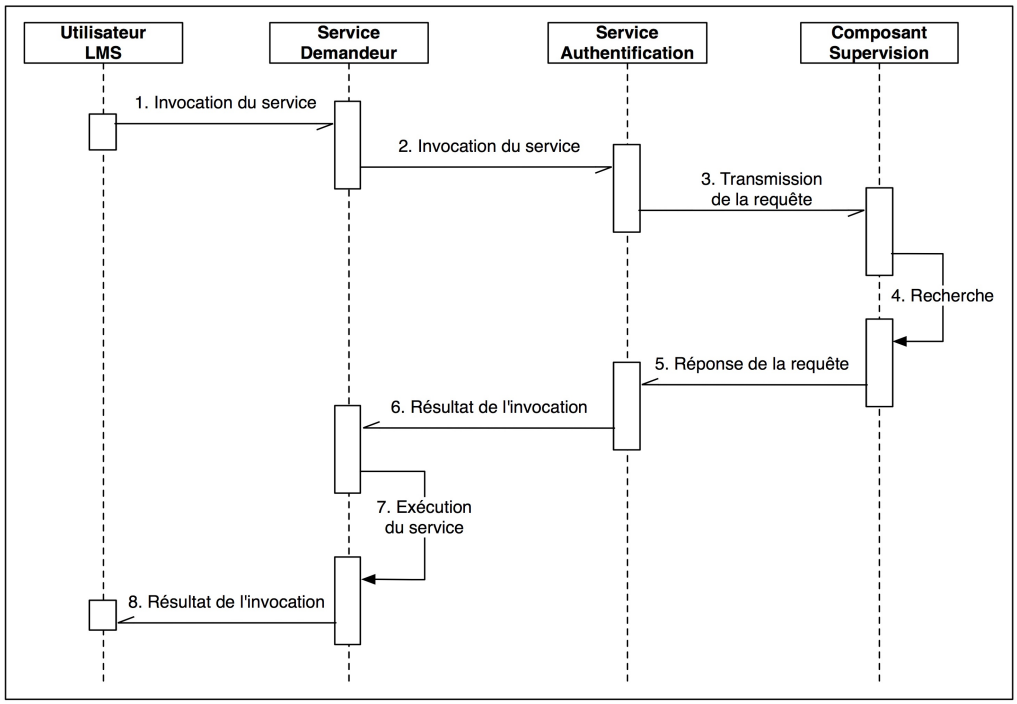


Fig. 6.5 – Scénario d'authentification

La méthode getUserAccount du service d'authentification spécifiée par le tableau 6.14 permet l'obtention des caractéristiques d'une personne par rapport à un système. Bien que cette méthode soit définie au sein du composant de supervision exposé dans la section 7.4.2,

nous présentons ici sa spécification afin de fournir au lecteur une vue globale de notre système d'authentification. Le service d'authentification intégré dans la couche de Fédération correspond en fait à l'appel de cette procédure distante.
Les paramètres userFirstName et userLastName permettent de définir l'identité de l'utilisateur, alors que le paramètre system correspond au nom du système cible (dans notre cas, un vivier de connaissance).

Nom de la méthode	**getUserAccount**
Type de retour	String[]
Paramètres	*Name* / *Type* userFirstName / String userLastName / String system / String
Erreurs	INVALID_USER_FIRST_NAME INVALID_USER_LAST_NAME INVALID_SYSTEM METHOD_FAILURE

TAB. 6.14 – Spécification de la méthode getUserAccount

La méthode getUserAccount peut générer les erreurs suivantes :

- **INVALID_USER_FIRST_NAME** si le prénom de l'usager n'est pas une chaîne de caractères, ou une chaîne de caractères vide.
- **INVALID_USER_LAST_NAME** si le nom de l'usager n'est pas une chaîne de caractères, ou une chaîne de caractères vide.
- **INVALID_SYSTEM** si le système cible n'est pas une chaîne de caractères, ou une chaîne de caractères vide.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

Le service d'authentification, c'est-à-dire l'envoi d'une requête au composant de supervision, pourrait être inclus dans l'ensemble des services de la couche de Fédération nécessitant une authentification. Cependant, notre approche facilite sa réutilisation par d'autres services qui pourraient être proposés ultérieurement aux utilisateurs.

La couche de Fédération permet de réaliser la première étape de la virtualisation des objets pédagogiques puisqu'elle fournit une vue unique des ressources stockées au sein de différents LOR. Elle fournit également un ensemble de services qui permettent d'exploiter ces ressources.

Nous exposons maintenant la couche de Mutualisation qui offre un accès facilité à ces ressources à partir d'un LMS et qui permet d'aboutir à la virtualisation des objets pédagogiques.

6.3 Couche de Mutualisation

La couche de Mutualisation joue deux rôles bien distincts : d'une part elle permet à l'utilisateur final d'atteindre la couche de Fédération ainsi que les services qui lui sont associés de manière transparente et homogène, et d'autre part elle fournit à cette même couche de Fédération les informations nécessaires à la réalisation du service d'importation des ressources externes au sein du LMS.

6.3.1 Lien entre les plates-formes de gestion d'apprentissage et la couche de Fédération

La couche de Mutualisation permet à un utilisateur connecté à la plate-forme d'apprentissage d'accéder facilement aux ressources retrouvées par les services de la couche de Fédération. Les services proposés par les LMS sont plus ou moins avancés mais l'opportunité d'insérer un nouveau document au sein de l'espace dédié est toujours offerte. Nous avons donc ajouté un hyperlien vers la couche de Fédération dans l'espace de travail dédié à l'insertion de nouveaux documents (cf. figures 8.10 et 8.11). Ainsi, le déploiement de la couche de Mutualisation au sein d'un LMS est très aisée car (a) peu de pages HTML[6] ont besoin d'être modifiées au sein du LMS, et (b) repérer l'espace de travail correspondant est très rapide. D'autre part, plusieurs contraintes sont satisfaites :

- Pour l'administrateur, le déploiement de la couche de Virtualisation au sein de la plate-forme cible est simple à réaliser : un fichier doit être mis à jour afin d'ajouter l'hyperlien vers la couche de Fédération, et un nouveau répertoire correspondant aux services de Fédération doit être ajouté dans l'arborescence de la plate-forme cible.
- Pour les créateurs de cursus, le déploiement de ressources stockées dans des LOR distants est facile à réaliser puisque notre approche présente une méthodologie similaire à celle mise en œuvre pour le déploiement de ressources internes au LMS.
- La couche de Virtualisation est facilement adaptable à différents LMS cibles puisque notre approche n'est pas fortement liée à une plate-forme particulière, comme le montre la section suivante.

[6]HyperText Mark-up Language

6.3.2 Fournisseur d'informations

Afin de rendre la couche de Fédération la plus indépendante et générique possible, le nombre de variables spécifiques au LMS et nécessaires à la bonne conduite des services de Fédération doit être aussi faible que possible. L'intégration ne se rapporte qu'à la gestion du matériel pédagogique et nécessite l'identification de seulement quatre caractéristiques :

- Le nom de l'utilisateur connecté au LMS afin de contrôler ses droits d'usage relatifs aux LOR.
- Le chemin du répertoire racine correspondant à l'espace dédié au stockage du matériel pédagogique au sein du LMS.
- L'identifiant du répertoire de la formation en cours afin de savoir où stocker les ressources au sein du répertoire racine.
- Le champ du formulaire de la plate-forme nécessaire au service d'importation pour indiquer la localisation d'un objet pédagogique stocké dans un vivier de références.

Ces quatre variables spécifiques au LMS peuvent être obtenues facilement : le nom de l'usager connecté (son *login*) est très souvent stocké dans une variable Session, le répertoire racine est soit identique, soit spécifique pour chaque instance du LMS (dans le dernier cas, il figure dans un fichier de configuration), l'identifiant du répertoire de la formation en cours correspond à une variable visible dans la barre d'adresse du navigateur, et le champ du formulaire est mentionné dans le même espace de travail que celui contenant l'hyperlien permettant d'atteindre la couche de Fédération. Lorsqu'un utilisateur cherche à accéder aux services de la couche de Fédération, la couche de Mutualisation prend connaissance de ces quatre caractéristiques et les transmet à la couche de Fédération qui est alors en mesure d'assurer l'ensemble des services présentés dans la section 6.2.

6.4 Synthèse

Nous avons proposé dans ce chapitre une architecture structurée en couches qui offre une vue unique de ressources pédagogiques réparties dans des viviers de connaissance distincts ainsi qu'un accès facilité à celles-ci à travers les plates-formes d'apprentissage : elle permet donc de mettre en œuvre la virtualisation des objets pédagogiques qui favorise le partage et la réutilisation du matériel pédagogique. Nous nous sommes appuyés sur la spécification SQI pour retrouver les métadonnées renfermées dans les viviers de connaissance hétérogènes, et nous avons spécifié un ensemble de services qui permettent d'exploiter les documents pédagogiques associés à ces métadonnées et qui sont résumés dans le tableau 6.15.

Services	*Méthodes*
Téléchargement Consultation	downloadDocument(docIdentifier, login, password) : string previewUrl(url) : void
Importation d'une URL Importation d'un document	importUrl(url, fieldToFill) : void importDocument(docIdentifier, folderLocation, login, password) : void
Indexation d'une ressource	indexLearningObject(document, metadata, folderLocation, login, password) : string
Génération automatique de métadonnées	getContextKeywords(targetLms) : string[] getFileInfo(extension) : string[] extractScormMetadata(manifestFile) : string[] getAuthorInfo(vcard) : string[]
Authentification des utilisateurs	getUserAccount(userFirstName, userLastName, system) : string[]

TAB. 6.15 – Les services permettant d'exploiter les ressources pédagogiques

La couche supplémentaire que nous avons introduite présente à la fois un caractère générique qui est illustré dans la dernière partie du manuscrit, mais également une transparence qui masque à l'utilisateur final la complexité de l'architecture globale.

Ainsi, une masse importante d'objets pédagogiques stockés dans des viviers de connaissance distincts est disponible aux concepteurs de cursus d'apprentissage. Toutefois, le service de recherche présenté dans ce chapitre, basé sur un ou plusieurs mots-clés spécifiés par l'utilisateur, ne permet pas de proposer à l'éducateur les ressources les plus adaptées et pertinentes à son contexte d'apprentissage : un concepteur de cursus doit parcourir un grand nombre d'objets pédagogiques parmi ceux qui lui sont proposés par le service de recherche avant de trouver une ressource correspondant à ses besoins. Alors pour lever ce verrou, le chapitre suivant expose notre approche dédiée à la supervision de l'usage des différentes entités impliquées dans l'architecture de Virtualisation. Cette approche permet de collecter et d'exploiter des données statistiques sur l'usage des différents composants de l'architecture, ainsi que sur les utilisateurs de l'environnement d'apprentissage.

Chapitre 7

Supervision de l'exploitation des ressources

Sommaire

7.1 Objectifs **86**

7.2 Le modèle commun de l'information **87**

7.2.1 Les principaux éléments de la modélisation CIM 87

7.2.2 Les niveaux d'abstraction 87

7.2.3 Le protocole de transport 89

7.2.4 Les raisons de notre choix 90

7.3 Modélisation de l'EIAH **91**

7.3.1 Vue générale 91

7.3.2 Modélisation de l'environnement d'apprentissage 93

7.3.3 Modélisation des utilisateurs 98

7.3.4 Relations entre l'EIAH et les utilisateurs 99

7.4 Architecture globale de notre EIAH **100**

7.4.1 L'architecture de gestion WBEM 100

7.4.2 Un CIM *Object Provider* pour l'EIAH 102

7.4.3 Intégration du cadre de gestion WBEM au sein de notre architecture . 106

7.5 Synthèse **110**

La couche de Virtualisation présentée dans le chapitre précédent introduit le problème de la supervision de l'exploitation des ressources d'apprentissage. En effet, un objet pédagogique peut être utilisé par un grand nombre de systèmes d'apprentissage et de multiples utilisateurs ; ces informations représentent des données importantes pour la sélection des objets pédagogiques proposés par le service de recherche de la couche de Fédération, puisque celui-ci pourrait alors choisir les ressources en fonction de leur taux d'utilisation, du contexte d'apprentissage dans lesquelles elles ont déjà été intégrées, des utilisateurs les ayant exploitées, etc.

Nous avons également présenté dans le chapitre 4 les fonctionnalités d'observation existantes associées aux systèmes d'apprentissage, et montré leurs inconvénients qui ne permettent pas de les mettre en oeuvre dans le cadre de nos travaux. Alors nous proposons dans ce chapitre

une approche de supervision qui permet l'observation de l'usage des différentes entités impliquées dans l'architecture de Virtualisation, c'est-à-dire la collecte des informations statistiques relatives à l'usage de ces entités.

7.1 Objectifs

Au niveau des objets pédagogiques, nous souhaitons être capables de collecter et d'analyser des informations statistiques liées à leur utilisation comme le nombre de consultations de leurs métadonnées, le nombre de téléchargements ou d'importations au sein d'un cursus d'apprentissage, mais aussi les plates-formes d'apprentissage déployant ces cursus pédagogiques ou les systèmes de stockage renfermant les ressources. Aussi, nous souhaitons identifier les utilisateurs interagissant avec ces différentes entités afin d'être en mesure de proposer un outil d'observation adéquate à la supervision de l'ensemble de l'architecture de Virtualisation.

Pour atteindre ces objectifs tout en prenant en compte des caractéristiques dynamiques et de mise à l'échelle, il faut tout d'abord répondre à certaines questions [Ste05] : (a) Quelles sont les informations utiles à la supervision des interactions entre les entités de l'architecture de Virtualisation et comment les représenter ? (b) Comment collecter ces informations ? et (c) Où stocker ces informations et comment les exploiter ?

Dans ce chapitre, nous présentons la manière dont nous avons répondu à ces trois questions. Notre approche repose sur la modélisation de l'ensemble des entités impliquées dans un environnement informatique pour l'apprentissage humain. Notre premier travail a donc consisté à identifier les informations requises pour atteindre nos objectifs ; ensuite nous nous sommes attachés à formaliser ces informations à l'aide d'une approche standard de Gestion. En effet, pour des raisons d'intégration et de coopération mais aussi pour satisfaire des critères de réutilisation et de gain de temps, nous avons adopté un métamodèle commun : l'approche de modélisation CIM[1] spécifie un nouveau métamodèle sur la base des concepts objets qui répond à des besoins d'unification et d'extensibilité.

La suite de ce chapitre est organisée en quatre parties. La première présente les points importants du métamodèle CIM qui permettent de répondre à nos critères d'observation. Dans la deuxième, nous répondons à la première question posée ci-dessus et détaillons le modèle d'information qui caractérise un EIAH d'un point de vue Gestion. La troisième partie relève des aspects architecturaux de notre proposition qui couvrent les problématiques (b) et (c) précitées et qui permettent la mise en oeuvre et le déploiement de l'approche suggérée. Enfin, une synthèse du chapitre exposant les apports de notre approche est présentée dans la dernière partie.

[1] Common Information Model

7.2 Le modèle commun de l'information

7.2.1 Les principaux éléments de la modélisation CIM

CIM est un modèle de données pour la gestion de systèmes, de réseaux et d'applications. L'approche de modélisation CIM spécifie un nouveau métamodèle sur la base d'un modèle commun d'abstraction décrivant une connaissance sémantique du monde géré. Les éléments de base suivants illustrés par la figure 7.1 sont utilisés pour modéliser les entités à gérer [SJM04] :

- **Objet géré / Classe**. Les objets gérés sont tous dérivés de la classe CIM_ManagedElement. Toute classe d'objet est spécifiée par un ensemble de **propriétés** et de **méthodes**.
- **Opération / Action**. Une action correspond à une opération à invoquer sur un élément du modèle commun d'abstraction. Elle correspond à une méthode spécifique définie dans la description de la classe qui pourra être invoquée par un gestionnaire CIM.
- **Relation**. Deux types de relation sont considérés dans CIM :
  - (i) *Une relation d'héritage* : les classes peuvent être dérivées d'au plus une classe (héritage simple).
  - (ii) Une *association* mettant en relation au moins deux classes d'objets. Parmi les associations, deux associations ont été définies à part, à savoir la composition (CIM_Component) qui exprime une relation de composition, et la dépendance (CIM_Dependency) qui traduit une sémantique existentielle ou fonctionnelle.

7.2.2 Les niveaux d'abstraction

CIM offre une uniformisation des informations de gestion en proposant un ensemble de schémas divisés en trois niveaux distincts illustrés par la figure 7.2 :

1. **Le modèle CIM Core** [For00] : il définit un ensemble de classes et d'associations génériques à tout domaine de gestion, c'est-à-dire une structure de base pour tous les schémas futurs qui concernent des domaines particuliers. Ainsi, le modèle CIM Core est stable et comporte des classes, au concept abstrait, dont le nombre est assez limité.
2. **Le modèle CIM Common** : il définit des classes et des associations indépendantes de toute implantation, relatives à un domaine particulier de la gestion. Il se compose ainsi de différents sous-modèles qui concernent les applications [For03b], les systèmes [For03d], les réseaux [For03a], les politiques de gestion [For03c] ou la sécurité [For03e].

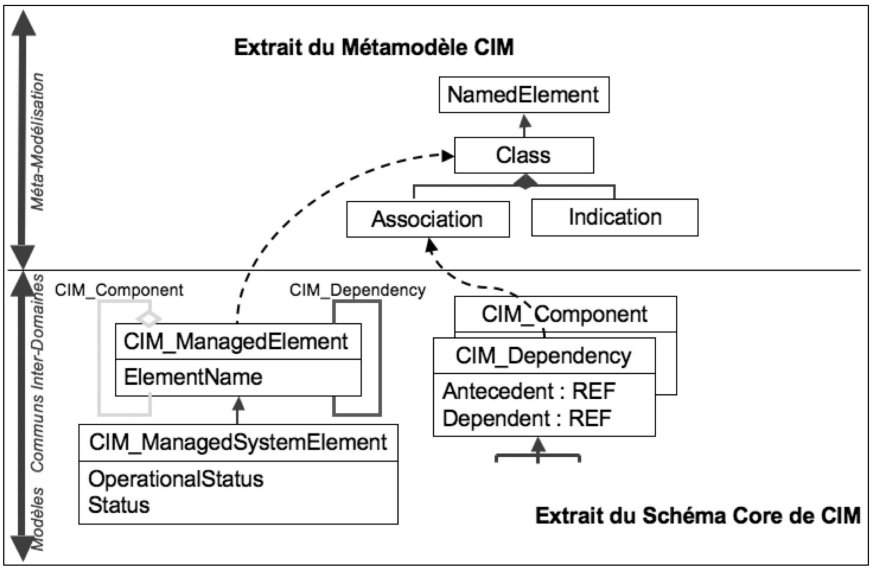


FIG. 7.1 – Les éléments de base de la modélisation CIM

3. **Les modèles CIM Extensions** : les schémas CIM Extension permettent d'étendre les classes existantes. Ces derniers sont très utiles pour spécialiser un schéma existant afin de prendre en compte les spécificités d'éléments particuliers ou pour couvrir un nouveau domaine qui ne soit pas pris en compte dans le modèle CIM Common.

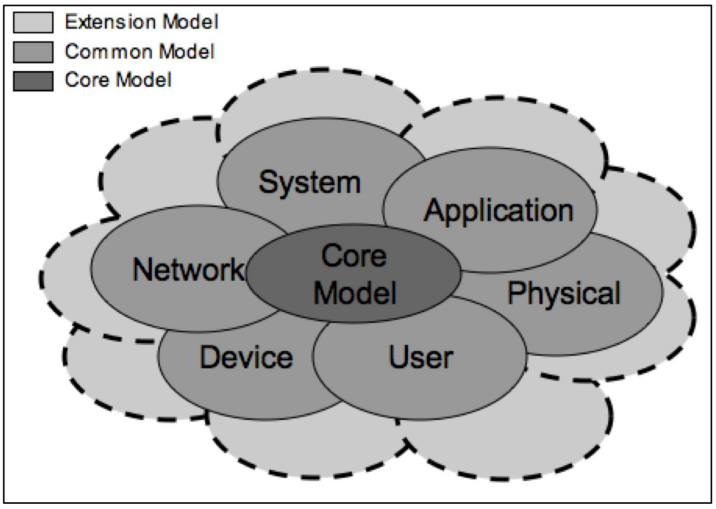


FIG. 7.2 – Extrait des différents niveaux d'abstraction du modèle CIM

C'est à partir des classes définies dans ces modèles CIM Core et CIM Common que vont être définies/modélisées par dérivation de nouvelles classes pour satisfaire des besoins spécifiques.

Dans nos travaux présentés plus loin, nous allons dériver certaines classes abstraites de ces modèles afin de représenter les éléments spécifiques d'un EIAH.

7.2.3 Le protocole de transport

Le protocole de communication proposé par le DMTF[2] s'appuie sur deux standards du *web* : XML pour l'encodage et HTTP [FGM+99] pour le transport. La spécification du DMTF repose sur une représentation du métamodèle CIM par des éléments XML et permet ainsi de générer une représentation uniforme des modélisations CIM aussi bien pour des classes que pour des objets gérés (instances).

Afin de réaliser le transport des informations CIM, le DMTF a opté pour l'utilisation du protocole HTTP étendu [NLL00]. Le standard proposé consiste à encapsuler la description XML des informations CIM dans le corps d'une unité de données HTTP. Cette encapsulation des informations CIM est schématisée par la figure 7.3.

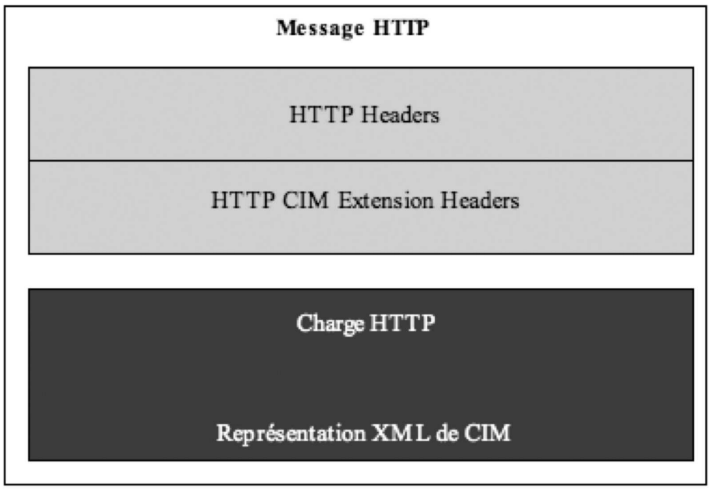


Fig. 7.3 – Encapsulation des informations CIM dans les messages HTTP

L'entête du message HTTP comporte deux parties : les entêtes d'un message POST définis par le protocole HTTP, et les entêtes CIM définies dans le cadre de l'extension de HTTP pour le transport de l'information CIM.
Ces spécifications CIM/HTTP ont été publiées en 2002 ; plusieurs plates-formes OpenSource[3] ont vu le jour et adopté ces spécifications. D'autres travaux plus récents ont permis d'établir une correspondance entre les informations CIM et le protocole SOAP, mais ce caractère récent fait

[2]Distributed Management Task Force
[3]http://wbemservices.sourceforge.net/#Other)

que, à ce jour, peu d'outils proposent une interface qui implémente ces nouvelles spécifications.

7.2.4 Les raisons de notre choix

Le métamodèle CIM et son ensemble de diagrammes représentent un choix approprié pour atteindre nos objectifs de supervision car son approche par niveaux d'abstraction offre plusieurs avantages incontestables :

- **Un modèle de gestion unificateur** : toutes les classes identifiant des ressources à gérer héritent de la classe CIM_ManagedElement du modèle CIM Core.
- **Une réutilisation optimale par métier/domaine** : CIM offre la possibilité de définir au sein des modèles CIM Common les modèles génériques d'un domaine (Réseau, Système, etc.) ou métier particulier (spatial, finance, e-formation).
- **Le métamodèle inclut la modélisation des utilisateurs** à travers le modèle CIM User [For03e] qui représente les personnes et leurs caractéristiques associées.
- **Une approche extensible** : le dernier niveau d'abstraction - modèle CIM Extensions - permet de spécialiser les modèles génériques d'un domaine particulier pour un projet ou environnement particulier.
- **Des choix conceptuels non spécifiques au monde de la gestion** : concepts Objet, utilisation des diagrammes de classes UML avec quelques particularités propres, syntaxe MOF[4] décrite par une DTD[5] XML [DMH+00] [HF99], etc. Des exemples de fichiers MOF sont donnés dans la section 7.3.2.
- **Le protocole de transport assure la mise à l'échelle** : les informations de gestion CIM sont encapsulées dans les entêtes HTTP et permettent de facilement traverser les pare-feux ou proxy présents au sein d'une architecture réseau.

La section suivante présente notre proposition de modélisation d'un EIAH conforme aux spécifications CIM et qui représente les différents systèmes et entités introduits dans l'architecture de Virtualisation illustrée par la figure 6.1. Nos objectifs étant de superviser l'exploitation des objets pédagogiques ainsi que les utilisateurs manipulant ces ressources et non pas l'ensemble des couches de l'architecture, notre modèle d'information se focalise sur l'infrastructure permettant de stocker et de consommer les ressources d'apprentissage ; ce modèle ne prend pas en compte les services intermédiaires et transparents (couches de Virtualisation et Services de Traitement) qui permettent aux utilisateurs d'interagir avec les objets pédagogiques.

[4]Managed Object Format
[5]Document Type Definition

7.3 Modélisation de l'EIAH

7.3.1 Vue générale

Un des objectifs que nous nous sommes fixés durant la conception de notre modèle concerne la généricité. Même si notre première contribution est relative à la communication entre plates-formes d'enseignement et viviers de connaissance pour permettre l'échange d'objets pédagogiques entre ces deux types de système, il nous semble important que le modèle de l'information que nous proposons puisse être appliqué et étendu à d'autres systèmes et ressources d'apprentissage. Ainsi, une application de supervision peut disposer d'une vue uniforme et cohérente de différentes infrastructures de formation en ligne dont elle assure l'observation. Notre travail a donc consisté à identifier les concepts communs à tout système d'e-formation d'un point de vue observation. Nous en avons recensé deux, qui sont :

1. **L'environnement d'apprentissage** qui regroupe d'une part les différents systèmes tels que les LMS ou les LOR, et d'autre part les ressources d'apprentissage elles-mêmes [BVD05].
2. **Les utilisateurs** qui interagissent avec l'environnement d'apprentissage afin d'exploiter les différents services qui leur sont offerts [BVS06].

Ces deux concepts ont conduit à l'élaboration de deux sous-modèles définis dans le cadre du schéma CIM : l'un est destiné à la supervision de l'environnement d'apprentissage, l'autre à la gestion des utilisateurs, des relations d'association permettant de faire le lien entre ces deux sous-modèles. La figure 7.4 montre la modélisation des concepts précités. Notons que le préfixe de nommage des classes que nous avons définies est EIAH, alors que les classes définies dans le modèle CIM portent naturellement le préfixe CIM. Les propriétés ou méthodes qui apparaissent en gras dans les figures qui suivent sont celles que nous avons spécifiées, les autres étant définies au sein des classes CIM existantes. Dans la suite de cette section, nous décrivons en détail le modèle de l'environnement d'apprentissage et les relations de celui-ci avec le modèle des utilisateurs, alors que ce dernier est brièvement exposé puisqu'il est défini au sein du modèle existant CIM User [For03e].

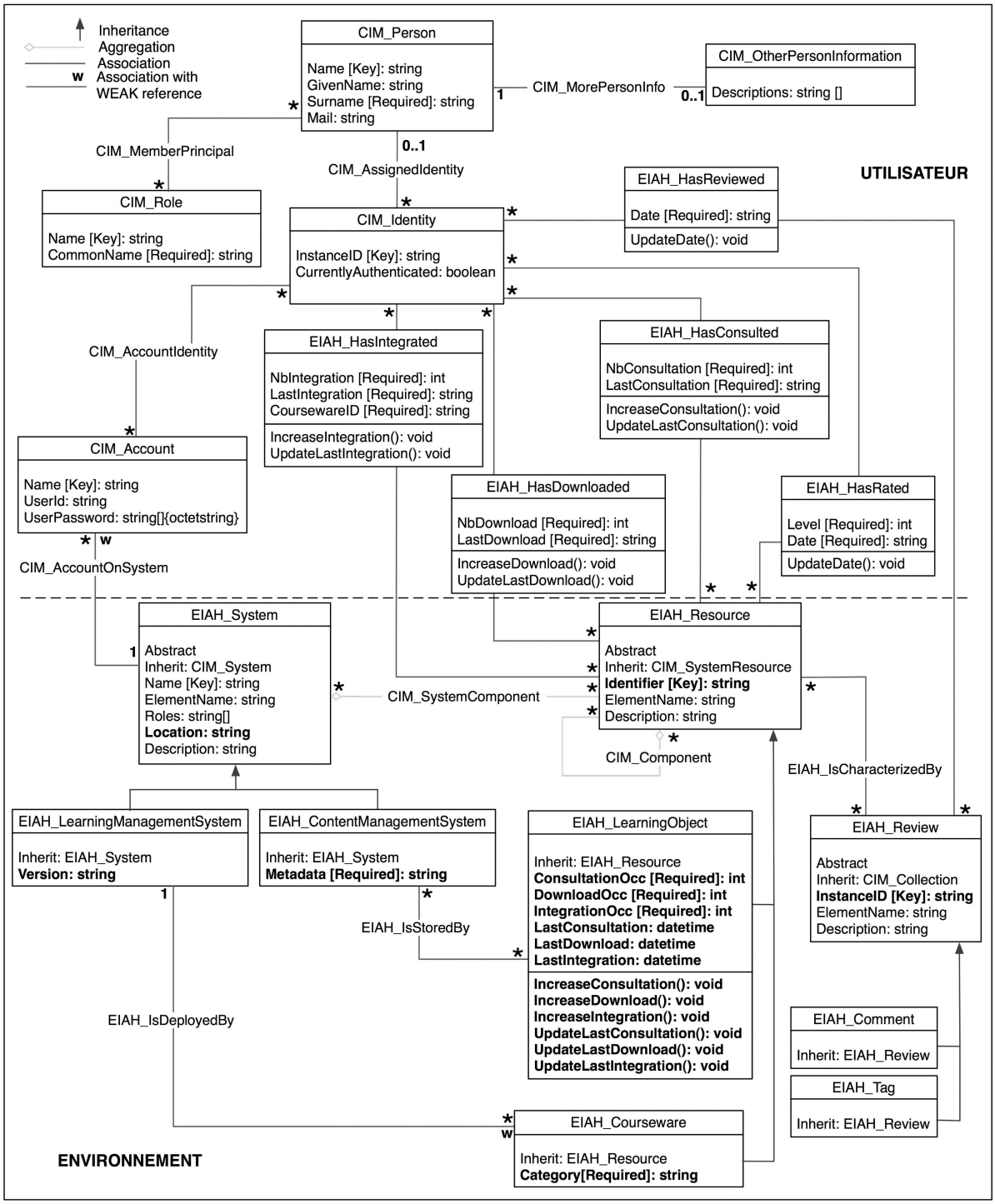


FIG. 7.4 – Modélisation d'un EIAH

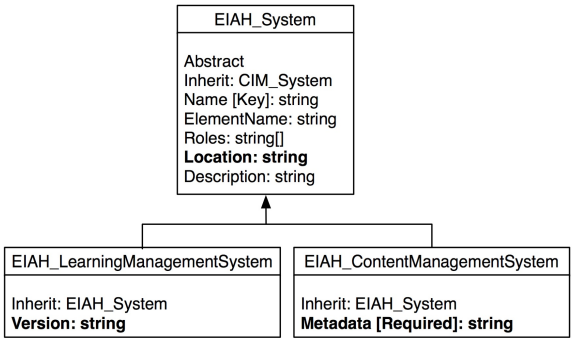
7.3.2 Modélisation de l'environnement d'apprentissage

7.3.2.1 Les objets gérés

Comme mentionné dans la section 7.2.1, toute classe CIM hérite de la classe abstraite CIM_ManagedElement qui représente, de la manière la plus générale, un élément géré dans le cadre d'une approche de gestion. Celui-ci est défini par trois attributs qui renseignent sur son nom (ElementName) et deux descriptions, l'une courte (Caption) et l'autre longue (Description). Ces attributs sont donc valables pour l'ensemble des objets gérés présentés ci-après.

Les systèmes d'apprentissage

La classe abstraite CIM_System représente une entité constituée de différents composants ; un exemple de CIM_System est un système de gestion de Base de Données ou une application J2ee. Dans notre EIAH, la classe abstraite EIAH_System de la figure 7.5 hérite de la classe CIM_System et modélise les systèmes intégrés dans l'environnement d'apprentissage. La partie droite de la figure donne le fichier MOf associé.




```
// ==================================================================
// EIAH_System
// ==================================================================

[Abstract, Description ("EIAH_System extends the CIM_System
in order to better match with systems included within an EIAH")]

class EIAH_System: CIM_System {

    [MaxLen(256), Description("The URL of the infrastructure system")]
    string Location;

     };
```

FIG. 7.5 – Représentation des systèmes d'apprentissage

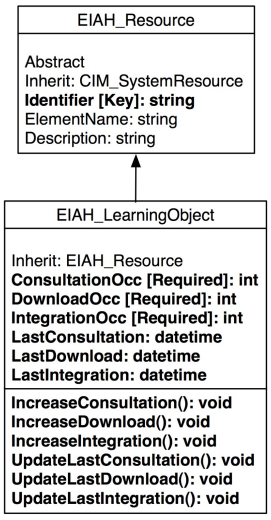
Cette classe offre la possibilité de modéliser les niveaux supérieur et inférieur de l'architecture de Virtualisation, c'est-à-dire les interfaces pédagogiques et les systèmes de stockage des objets pédagogiques. Elle présente trois attributs spécifiques :

- Name : le nom d'une instance représente la clé de cette classe. Il est défini au sein même de la classe CIM_System et représente une chaîne de caractères non lisible par les Humains.
- Roles : cet élément indique le rôle du système. Une valeur de cet attribut sera par exemple "stocke des ressources pédagogiques" ou "offre des services d'apprentissage".
- Location : dans la mesure où les systèmes d'apprentissage sont, pour la plupart, des systèmes basés sur les technologies *web*, nous avons ajouté cette propriété qui indique la

localisation du système sur l'Internet. Toutefois, cet attribut n'est pas obligatoire et laisse la possibilité de définir des systèmes comme les outils auteurs qui ne sont pas disponibles en ligne.

La classe EIAH_LearningManagementSystem spécifie la classe EIAH_System en indiquant la version du système, alors que la classe EIAH_ContentManagementSystem présente un attribut additionnel obligatoire qui renseigne le standard de métadonnées implémenté au sein du système de stockage.

Les objets pédagogiques




```
// ==================================================================
// EIAH_LearningObject
// ==================================================================

class EIAH_LearningObject: EIAH_Resource {

   [Required, Description("This property represents the number of users
having looked at the learning object's metadata")]
   uint32 ConsultationOcc;
   [Description("This method increases by 1 the ConsultationOcc
property")]
   uint32 IncreaseConsultationOcc();

   [Description("This property represents the datetime indicating the
last time the metadata have been read")]
   string LastConsultation;
   [Description("This method sets the LastConsultation property to the
current date")]
   string UpdateLastConsultation();

   [Required, Description("This property represents the number of users
having downloaded the learning object")]
   uint32 DownloadOcc;
   [Description("This method increases by 1 the DownloadOcc property")]
   uint32 IncreaseDownloadOcc();

   [Description("This property represents the datetime indicating the
last time the LO has been downloaded")]
   string LastDownload;
   [Description("This method sets the LastDownload property to the
current date")]
   string UpdateLastDownload();

   [Required, Description("This property represents the number of
cursus integrating the learning object")]
   uint32 IntegrationOcc;
   [Description("This method increases by 1 the IntegrationOcc
property")]
   uint32 IncreaseIntegrationOcc();

   [Description("This property represents the datetime indicating the
last time the LO has been integrated within a learning design")]
   string LastIntegration;
   [Description("This method sets the LastIntegration property to the
current date")]
   string UpdateLastIntegration();

    };
```

FIG. 7.6 – Représentation des ressources d'apprentissage

La classe abstraite EIAH_Resource décrit les ressources intégrées dans l'environnement d'apprentissage. Elle hérite de la classe CIM_SystemResource qui modélise toute entité gérée par un système d'exploitation ou un logiciel, et qui est disponible pour ce système ou logiciel. Le modèle CIM Core définit une relation d'agrégation CIM_SystemComponent entre cette classe et

la classe CIM_System évoquée dans la section précédente, donc cette relation est intrasèquement établie entre les classes EIAH_System et EIAH_Resource (cf. figure 7.9).

La classe abstraite CIM_SystemResource ne propose pas d'identifiant par défaut, c'est aux classes l'implémentant de le définir ; nous avons donc spécifié l'attribut additionnel Identifier qui représente la clé de la classe et qui permet de repérer de façon unique une ressource.

La classe EIAH_LearningObject illustrée par la figure 7.6 spécialise la classe EIAH_Resource afin de décrire de façon plus précise un objet pédagogique d'un point de vue supervision. Pour une instance de cette classe, trois attributs additionnels (ConsultationOcc, DownloadOcc et IntegrationOcc) donnent respectivement le nombre de consultations des métadonnées, le nombre de téléchargements et le nombre d'intégrations de cet objet au sein d'un cursus d'apprentissage. De plus, trois propriétés complémentaires (LastConsultation, LastDownload et LastIntegration) renseignent sur la date de dernière consultation, de téléchargement et d'intégration de l'objet pédagogique.
Un ensemble de méthodes sont associées aux attributs précités et permettent d'incrémenter de façon unitaire le nombre de consultations, téléchargements et intégrations, ou de mettre à jour les dates liées à ces mêmes événements.

A ce jour, seule la classe EIAH_LearningObject hérite de la classe EIAH_Resource pour représenter une ressource d'apprentissage, mais d'autres types de ressources pédagogiques comme par exemple celles issues du domaine des ontologies pourraient également spécialiser cette classe abstraite.

Les cursus pédagogiques

La classe EIAH_Courseware illustrée par la figure 7.7 hérite également de la classe EIAH_Resource et décrit un cursus pédagogique qui permet aux étudiants et enseignants d'apprendre et d'enseigner à travers un ensemble de services d'apprentissage délivrés par un LMS. En plus des propriétés définies par sa classe parente, elle spécifie la discipline associée (telle que "Informatique" ou "Littérature") au moyen de l'attribut obligatoire Category ; sa valeur est la même que celle indiquée au sein de la plate-forme d'enseignement dans laquelle ce cours est déployé.

Les labels et commentaires

L'objectif de la classe EIAH_Review représentée par la figure 7.8 est d'offrir aux utilisateurs finaux un service personnalisé de recherche et de classification des objets pédagogiques qui leur permet de spécifier leurs propres labels (EIAH_Tag) et commentaires (EIAH_Comment) pour

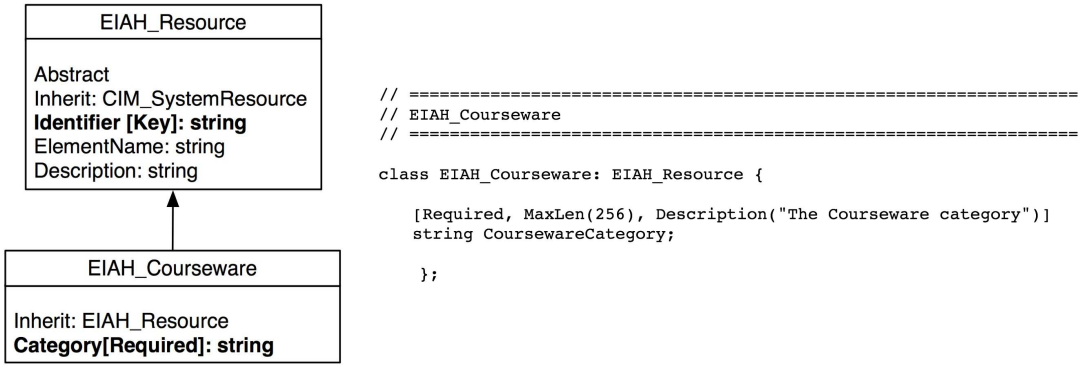


FIG. 7.7 – Représentation des cursus pédagogiques

une ressource donnée. De plus en plus de systèmes tels que Amazon™ ou le client de courrier électronique Google[6] offrent de tels services et permettent ainsi un système personnalisé de classification des ressources.

Le modèle CIM Core propose la classe abstraite CIM_Collection pour fournir un ensemble d'éléments représentant des collections d'objets gérés. La classe EIAH_Review spécialise cette classe et définit un ensemble de labels et commentaires qui peuvent être associés à une ressource pédagogique. Les attributs de cette classe sont ceux définis par la classe racine CIM_ManagedElement du métamodèle, complétés par la propriété InstanceID qui permet d'identifier de manière unique une instance de cette classe. Notons que l'attribut Description est un simple mot dans le cas d'un label, alors qu'il est composé d'un ensemble de mots dans le cas d'un commentaire.

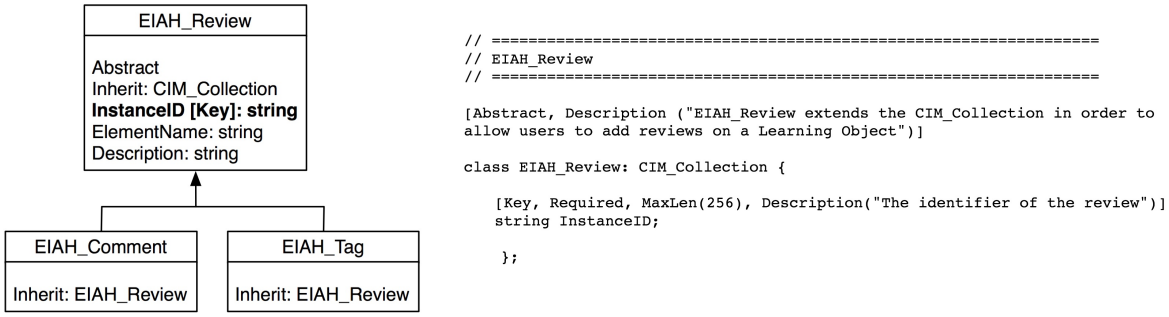


FIG. 7.8 – Représentation des labels et commentaires

Nous venons de modéliser les objets que nous souhaitons superviser dans un environnement informatique pour l'apprentissage humain, nous décrivons maintenant les différentes relations entre ces objets gérés.

[6] http://mail.google.com

7.3.2.2 Les associations

Les classes d'association expriment les relations entre classes gérées de l'environnement d'apprentissage. Elles jouent un rôle très important dans la mesure où ce sont ces relations qui permettent de retrouver les informations pertinentes d'observation et de gestion. Au sein du modèle d'information de l'environnement d'apprentissage, deux relations d'agrégation sont intrasèquement définies dans le modèle CIM Core, auxquelles nous avons apporté trois relations d'association illustrées par la figure 7.9.

La relation de composition CIM_Component de cardinalité * - * entre des instances de la classe EIAH_Resource permet d'exprimer d'une part le fait qu'un cursus d'apprentissage est constitué d'un ensemble de ressources pédagogiques, et d'autre part qu'un objet pédagogique peut être composé d'un ensemble d'autres objets pédagogiques. Cette relation permet donc de retrouver l'ensemble des ressources incorporées au sein d'un cursus particulier ou les cours déployant une ressource particulière, ainsi que les différents objets constituant une ressource particulière. L'agrégation CIM_SystemComponent indique qu'un système d'apprentissage est composé de ressources pédagogiques.

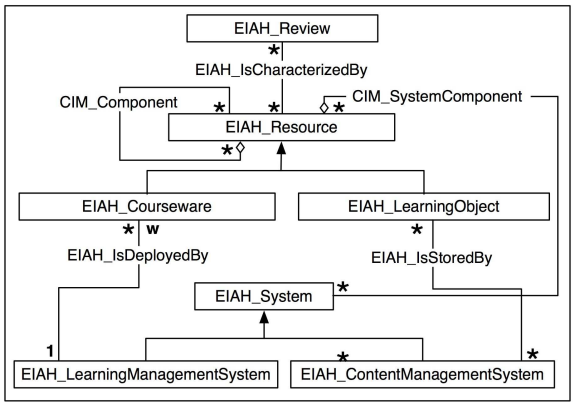


FIG. 7.9 – Les associations entre classes gérées de l'environnement d'apprentissage

Les trois associations spécifiques aux objets gérés décrits précédemment sont les suivantes :

- EIAH_IsStoredBy lie les ressources pédagogiques aux systèmes de stockage de ces mêmes ressources. Même si une ressource n'est, à priori, stockée que dans un seul vivier de connaissance, nous estimons qu'elle peut être dupliquée dans plusieurs LOR ; d'où la cardinalité * - * ce cette relation. Cette classe d'association permet de retrouver toutes les ressources renfermées dans un vivier de connaissance particulier, ou le(s) système(s)

stockant un objet pédagogique particulier.

- EIAH_IsDeployedBy associe les plates-formes d'enseignement et les cursus pédagogiques. La cardinalité 1 - w* exprime le fait qu'un cours particulier ne peut être déployé que dans une seule plate-forme, mais qu'un LMS peut contenir plusieurs cursus. L'association spécifie alors l'ensemble des cours déployés au sein d'un LMS donné, ainsi que le LMS intégrant un cursus particulier.
- EIAH_IsCharacterizedBy référence les ressources pédagogiques et les labels/commentaires spécifiés par les utilisateurs. Elle permet de connaître les labels (ou commentaires) caractérisant une ressource spécifique, ainsi que les objets ou cursus d'apprentissage qui sont associés à un label (ou commentaire) particulier. Cette relation a pour cardinalité * - * puisqu'un label peut être appliqué à plusieurs ressources, et qu'une ressource peut être caractérisée par plusieurs commentaires.

Les classes d'objets gérés et d'associations présentées ici permettent de modéliser les composants informatiques d'un EIAH ; dans la section qui suit nous nous intéressons à la modélisation des différents acteurs de l'e-formation qui interagissent avec ces composants.

7.3.3 Modélisation des utilisateurs

Nous ne décrivons pas les différentes classes et relations de la figure 7.10 impliquées dans la gestion des utilisateurs comme nous l'avons fait pour l'environnement d'apprentissage, nous ne présentons que les objectifs de chacune d'entre elles. Cet ensemble de classes étant défini dans le modèle existant CIM User [For03e], le lecteur est invité à consulter ce document pour plus de détails.

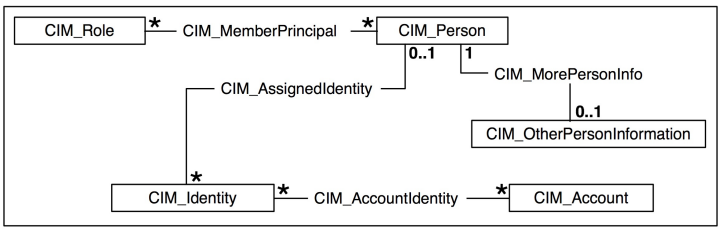


Fig. 7.10 – Représentation des utilisateurs dans le modèle CIM

La classe CIM_Person est utilisée pour représenter les personnes et permet notamment l'incorporation d'informations issues d'annuaires de type LDAP[7][WHK97]. L'ensemble des propriétés de cette classe sont définies par une autre classe, CIM_OtherPersonInformation, qui pro-

[7]Lightweight Directory Access Protocol

pose une vaste série d'attributs permettant de décrire de manière complète les multiples caractéristiques d'une personne. La classe CIM_MorePersonInfo établit la relation entre une personne et ses particularités.

La classe CIM_Role permet de définir une position ou un ensemble de responsabilités au sein d'une organisation ou d'un système ; elle est liée à une personne à travers la relation CIM_MemberPrincipal.

La classe CIM_Identity représente un élément géré du modèle d'information (par exemple une personne) par le biais de la relation CIM_AssignedIdentity, et vise des objectifs de sécurité au sein du domaine dans lequel cet élément est défini. Elle offre par exemple la possibilité pour une même personne de posséder plusieurs identités sur un même système.

Enfin, la classe CIM_Account modélise les comptes utilisateur et permet de vérifier les identités des utilisateurs ainsi que leurs privilèges à travers la relation CIM_AccountIdentity.

7.3.4 Relations entre l'EIAH et les utilisateurs

Nous avons exposé précédemment les deux sous-modèles de notre schéma global ; nous nous concentrons maintenant sur les différentes relations illustrées sur la figure 7.11 qui réunissent le modèle de l'environnement d'apprentissage et celui des utilisateurs.

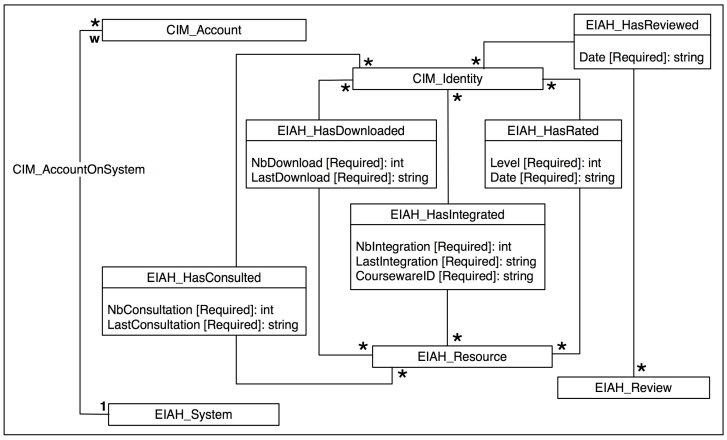


FIG. 7.11 – Les associations entre l'environnement d'apprentissage et les utilisateurs

La relation CIM_AccountOnSystem lie les classes CIM_Account et EIAH_System afin d'établir les correspondances entre comptes utilisateur et systèmes d'apprentissage.

L'association EIAH_HasReviewed référence les classes CIM_Identity et CIM_Resource, et renseigne les labels ou commentaires spécifiés par un utilisateur spécifique, ainsi que l'ensemble

des utilisateurs ayant défini un label ou commentaire particulier.

Les relations EIAH_HasConsulted, EIAH_HasDownloaded, EIAH_HasIntegrated et EIAH_HasRated associent les classes CIM_Identity et CIM_Resource. Elles permettent de connaître respectivement :

- Les utilisateurs ayant consulté les métadonnées spécifiques d'une ressource particulière, ou l'ensemble des ressources consultées par un utilisateur spécifique.
- Les utilisateurs ayant téléchargé une ressource donnée sur leur poste local, ainsi que les ressources pédagogiques téléchargées par un utilisateur spécifique.
- Les utilisateurs ayant importés une certaine ressource externe stockée dans un vivier de connaissance au sein d'un ou plusieurs cursus pédagogiques, ou les objets d'apprentissage importés par un utilisateur spécifique.
- Le niveau d'appréciation attribué à des ressources par un utilisateur donné, ou l'ensemble des utilisateurs ayant attribué un niveau de satisfaction à un objet spécifique.

Afin de valider notre approche par modélisation des systèmes et utilisateurs d'un environnement informatique pour l'apprentissage humain, nous avons complété l'architecture de virtualisation présentée dans le chapitre 6. Cet enrichissement est basé sur l'architecture de gestion WBEM spécifiée par le DMTF qui supporte les fonctionnalités nécessaires à la mise en oeuvre des concepts CIM présentés dans cette section.

7.4 Architecture globale de notre EIAH

7.4.1 L'architecture de gestion WBEM

Le DMTF propose au travers de l'architecture WBEM une organisation distribuée de l'application de gestion en introduisant les notions d'*Object Manager* et d'*Object Provider* (cf. figure 7.12) [JMBS01] proposant des interfaces génériques basées sur les concepts CIM de classe, propriété, méthode et instance. Nous retenons dans la suite du manuscrit les abréviations OM-OP pour des raisons de simplification.

Le monde hétérogène à gérer est instrumenté par l'intermédiaire d'**Agents**, accessibles via des protocoles de gestion tels que SNMP[8], CMIP[9] ou DMI[10], mais aussi à travers des protocoles spécifiques tels que LDAP.

[8]Simple Network Management Protocol
[9]Common Management Information Protocol
[10]Desktop Management Interface

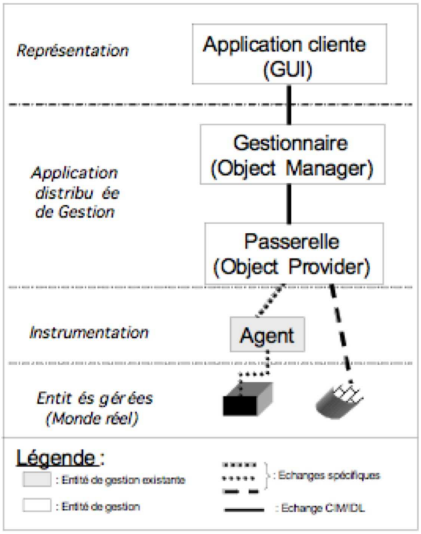


FIG. 7.12 – Les entités de gestion dans l'architecture WBEM

Des *applications clientes* restituent sous la forme d'interfaces graphiques orientées utilisateur les informations de gestion distribuées auprès des OM, et offrent aux gestionnaires humains la possibilité de réaliser des actions (mise à jour, invocation de méthode).

L'***Object Manager*** (cf. figure 7.13) est une application qui a charge d'un ou plusieurs domaines de gestion. Pour cela, l'OM dispose d'un référentiel contenant la description de sa connaissance de gestion (schéma CIM) ainsi que cette connaissance représentée sous la forme d'instances de classes CIM. Ce référentiel est exploité par des fonctions de gestion intégrées au sein de l'OM et définies sur la base des fonctions de gestion du modèle OSI[11]. Elles agissent sur les éléments définis dans le métamodèle et sont applicables sur n'importe quelle instance du référentiel ; chaque *Object Manager* dispose de l'ensemble de ces fonctions.

L'***Object Provider*** est une entité de gestion responsable de l'intégration d'un environnement à gérer comme par exemple l'environnement Unix, un environnement de base de données Oracle™ ou tout autre environnement comme LDAP. Il est constitué de deux applications : l'une serveur, l'autre servant. Pour chaque type d'environnement à intégrer, un OP est nécessaire. Ils permettent d'offrir une vue homogène des environnements hétérogènes à intégrer.

L'architecture WBEM permet donc de collecter les informations de supervision d'un domaine spécifique à travers le CIM *Object Provider*, de stocker ces informations dans le référentiel CIM par le biais du CIM *Object Manager*, et enfin d'exploiter ces informations en invoquant des fonctions et méthodes sur les classes et instances contenues dans le référentiel.

[11] Open Systems Interconnection

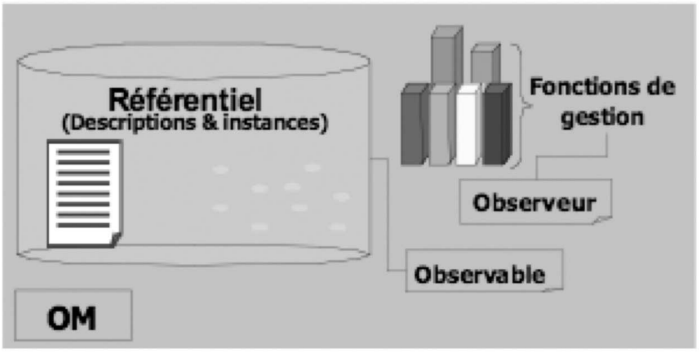


FIG. 7.13 – Structure d'un *Object Manager*

Alors pour être en mesure de traiter des informations de supervision spécifiques à l'EIAH, nous avons élaboré un *Object Provider* propre à ce domaine ; ce composant est constitué de deux interfaces : la première est dédiée à la réception de données transmises par les systèmes d'apprentissage, alors que la deuxième offre une communication avec un OM permettant de créer/modifier les instances CIM propres à l'EIAH.

7.4.2 Un CIM *Object Provider* pour l'EIAH

Le CIM *Object Provider* joue le rôle d'intermédiaire entre l'environnement d'apprentissage et l'environnement de Gestion. Afin de ne pas limiter la supervision à la couche de Fédération et de proposer un moyen simple pour permettre la collecte d'informations issues d'autres systèmes d'e-formation, notre OP est mis en oeuvre sous la forme d'un service *web*.

7.4.2.1 L'interface dédiée à l'EIAH

Cette interface serveur remplit deux fonctions bien distincts : d'une part elle offre un service capable de recevoir des informations de supervision transmises par des systèmes dédiés à l'e-formation, et d'autre part elle accepte des requêtes émises par le service d'authentification de la couche de Fédération présenté dans la section 6.2.6.

Le service de réception de données de supervision

Le service de réception des informations est constitué de la méthode receiveData. Tout système d'apprentissage désireux de transmettre ses informations de supervision au CIM *Object Provider* devra invoquer cette méthode dont la spécification apparaît dans le tableau 7.1 ; le

paramètre supervisionData contient les informations de supervision transmises par le système d'apprentissage.

Nom de la méthode	**receiveData**
Type de retour	Void
Paramètres	*Name* / *Type* supervisionData / String[] event / String
Erreurs	INVALID_SUPERVISION_DATA INVALID_EVENT METHOD_FAILURE

TAB. 7.1 – Spécification de la méthode receiveData

La méthode receiveData peut générer les erreurs suivantes :

- **INVALID_SUPERVISION_DATA** si les informations de supervision transmises ne sont pas valides, ou que des informations sont manquantes.
- **INVALID_EVENT** si le paramètre event n'est pas supporté par l'OP.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

Le traitement associé à cette méthode consiste en fait à invoquer la méthode de l'interface CIM présentée ci-après et qui correspond à l'événement mentionné par le paramètre event. En effet, la nature des informations émises par les systèmes d'apprentissage est fonction des actions effectuées par l'utilisateur du système, donc le traitement associé à la réception de données par l'OP est dépendant du paramètre event de la méthode receiveData.
Le service de réception des informations représente une interface native du domaine de l'EIAH permettant d'intégrer, dans notre cadre de supervision, les actions effectuées par un utilisateur sur n'importe quel système d'apprentissage.

Le service d'authentification

Le service permettant d'accepter des requêtes d'authentification correspond à la méthode getUserAccount présentée dans la section 6.2.6 (le tableau 7.2 rappelle sa spécification). Le traitement associé à cette méthode est effectué par l'interface CIM de l'OP, et plus spécifiquement par l'invocation de la méthode getAccountInstance décrite dans la section suivante. Encore une fois, ce service n'effectue aucun réel traitement, sa fonction est de permettre la communication avec des systèmes d'apprentissage hétérogènes mais basés sur les technologies *web*.

Nom de la méthode	**getUserAccount**
Type de retour	String[]
Paramètres	

Name	*Type*
userFirstName	String
userLastName	String
system	String
cimomLocation	String

Erreurs	INVALID_USER_FIRST_NAME INVALID_USER_LAST_NAME INVALID_SYSTEM NO_CIMOM_LOCATION METHOD_FAILURE

TAB. 7.2 – Spécification de la méthode getUserAccount

7.4.2.2 L'interface de communication avec le CIM *Object Manager*

Cette interface servant permet de créer de nouvelles instances de classes CIM propres au domaine de l'EIAH avant de les transmettre au CIM *Object Manager* qui les insérera dans le référentiel CIM, ou de modifier des instances existantes.

Le traitement associé au service de réception

Lorsque l'interface dédiée à l'EIAH reçoit des informations de supervision, elle invoque le service de l'interface CIM afin de créer/modifier les instances qui correspondent aux données transmises. Nous avons défini un ensemble de méthodes qui sont associées aux différents services de la couche de Fédération et qui impliquent la création ou la modification d'instances du modèle d'information défini précédemment (cf. figure 7.4) :

- **La méthode searchEvent** est invoquée lorsqu'un utilisateur exploite le service de recherche.
- **La méthode metadataEvent** correspond à la consultation des métadonnées d'un objet pédagogique particulier par un utilisateur.
- **La méthode downloadEvent** traduit le téléchargement d'une ressource d'apprentissage, ou la consultation de l'URL associée à une ressource.
- **La méthode importationEvent** est invoquée lorsqu'un utilisateur importe un objet pédagogique au sein d'un cursus d'apprentissage d'une plate-forme de téléformation.

- **La méthode indexationEvent** exprime l'indexation d'un nouvel objet pédagogique au sein d'un vivier de connaissance.

Les spécifications de chacune de ces méthodes sont identiques, seul le traitement qui leur est associé diffère. Le tableau 7.3 donne la spécification de la méthode searchEvent : le paramètre supervisionData contient les données de supervision qui ont été transmises par l'interface dédiée à l'EIAH, et la localisation du CIM *Object Manager* responsable du référentiel CIM est indiquée par le paramètre cimomLocation.

Nom de la méthode	**searchEvent**
Type de retour	Void
Paramètres	*Name* / *Type* supervisionData / String[] cimomLocation / String
Erreurs	INVALID_SUPERVISION_DATA NO_CIMOM_LOCATION METHOD_FAILURE

TAB. 7.3 – Spécification de la méthode searchEvent

La méthode searchEvent peut générer les erreurs suivantes :

- **INVALID_SUPERVISION_DATA** si les informations de supervision ne sont pas valides ou que des informations sont manquantes.
- **NO_CIMOM_LOCATION** dans le cas où la localisation de l'OM n'est pas une chaîne de caractères ou si cette entité n'est pas disponible.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

Si les spécifications des différentes méthodes sont identiques, en revanche les données de supervision diffèrent en fonction de la nature des actions effectuées par les utilisateurs et correspondent aux attributs des classes de notre modèle d'information. Les informations nécessaires à la bonne conduite du traitement associé à chacune de ces méthodes et renfermées dans le paramètre supervisionData sont détaillées dans la section 7.4.3.2.

Le traitement du service d'authentification

Une autre méthode de l'interface CIM permet de retrouver les comptes d'un utilisateur par rapport à un système particulier ; c'est elle qui est invoquée par l'interface dédiée à l'EIAH lorsqu'un utilisateur tente d'accéder à un service de la couche de Fédération nécessitant une

authentification préalable. La méthode getAccountInstance du tableau 7.4 envoie une requête à l'OM en spécifiant le nom et le prénom de la personne par le biais des paramètres userLastName et userFirstName, et précise également le nom du système concerné par la requête à l'aide du paramètre system. Le CIM *Object Manager*, localisé par le paramètre cimomLocation, retourne alors l'instance de la classe CIM_Account correspondante.

Nom de la méthode	**getAccountInstance**
Type de retour	CIMInstance
Paramètres	*Name* / *Type* userFirstName / String userLastName / String system / String cimomLocation / String
Erreurs	INVALID_USER_FIRST_NAME INVALID_USER_LAST_NAME INVALID_SYSTEM NO_CIMOM_LOCATION METHOD_FAILURE

TAB. 7.4 – Spécification de la méthode getAccountInstance

La méthode getAccountInstance peut générer les erreurs suivantes :

- **INVALID_USER_FIRST_NAME** si le prénom de l'usager n'est pas une chaîne de caractères, ou une chaîne de caractères vide.
- **INVALID_USER_LAST_NAME** si le nom de l'usager n'est pas une chaîne de caractères, ou une chaîne de caractères vide.
- **INVALID_SYSTEM** si le système cible n'est pas une chaîne de caractères, ou une chaîne de caractères vide.
- **NO_CIMOM_LOCATION** dans le cas où la localisation de l'OM n'est pas une chaîne de caractères ou si cette entité n'est pas disponible.
- **METHOD_FAILURE** si l'invocation de la méthode échoue pour toute autre raison.

7.4.3 Intégration du cadre de gestion WBEM au sein de notre architecture

7.4.3.1 Architecture globale de notre EIAH

L'architecture globale de notre EIAH illustrée par la figure 7.14 prend en compte l'architecture de Virtualisation présentée dans le chapitre 6, à laquelle nous avons intégré les composants

spécifiés par l'initiative WBEM.

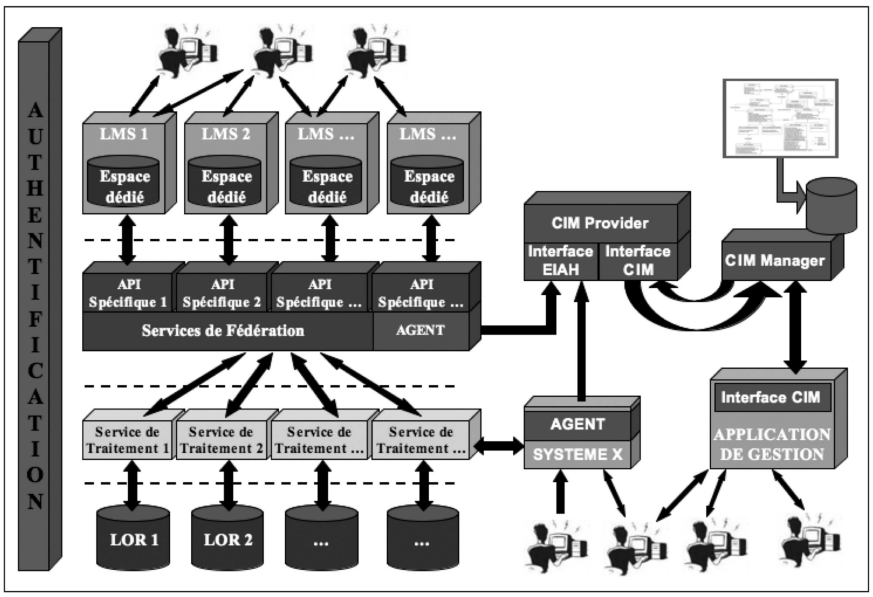


Fig. 7.14 – Architecture globale de notre EIAH

Pour permettre la supervision des entités exploitées par la couche de Virtualisation, nous y avons intégré un agent décrit dans la section suivante. Cet agent peut être adapté à d'autres systèmes dédiés à l'apprentissage, offrant ainsi la possibilité de recueillir des informations de supervision issues d'autres systèmes. Ce composant communique avec l'interface serveur de l'OP dédié à l'EIAH à travers les protocoles SOAP/HTTP.

L'autre interface de l'OP, l'interface CIM, communique avec un OM décrit dans la section 7.4.3.3 et qui a pour charge de maintenir la consistance et l'intégrité des informations qui sont contenues dans son référentiel CIM.

Le dernier composant que nous avons introduit, l'application de gestion sur la figure 7.14, offre sous la forme d'une interface graphique basée sur les technologies *web* une vue homogène des informations de gestion renfermées dans le référentiel CIM. Cette application, dont nous décrivons les fonctionnalités dans la section 7.4.3.4, permet l'émission de requêtes à l'OM grâce aux protocoles XML/HTTP et offre ainsi aux utilisateurs la possibilité de consulter l'ensemble des données de supervision.

7.4.3.2 Un agent dédié à l'EIAH

Les agents ont pour fonction de collecter les informations de gestion traduisant le comportement des objets gérés, puis de les transmettre à l'OP spécifique à leur environnement.

L'OP que nous avons présenté précédemment propose un processus serveur capable de recevoir les informations spécifiques à l'EIAH par le biais de la méthode receiveData. L'agent correspondant a donc pour charge de capturer les différentes informations nécessaires à la supervision des instances de notre modèle d'information. Le tableau 7.5 donne l'ensemble des informations capturées par l'agent en fonction des actions réalisées par l'utilisateur d'un système informatique dédié à l'e-formation.

Action effectuée par l'utilisateur	*Informations capturées par l'agent*
searchEvent	système d'apprentissage exploité par l'utilisateur, prénom de l'utilisateur, nom de l'utilisateur, adresse de courrier électronique de l'utilisateur, mots clés recherchés par l'utilisateur, rôle de l'utilisateur au sein du système, *login* de l'utilisateur, mot de passe de l'utilisateur, identifiant du système d'apprentissage, nom du système d'apprentissage, description du système d'apprentissage, type du système d'apprentissage, localisation du système d'apprentissage, version du système d'apprentissage (facultatif).
metadataEvent	Idem **searchEvent** + identifiant de la ressource consultée, titre de la ressource consultée, description de la ressource consultée, localisation de la ressource consultée, type du vivier de connaissance stockant la ressource, identifiant du vivier de connaissance, nom du vivier de connaissance, description du vivier de connaissance, localisation du vivier de connaissance, métadonnées implémentées au sein du vivier de connaissance.
downloadEvent	Idem **downloadEvent**
importationEvent	Idem **searchEvent** + identifiant de la plate-forme d'apprentissage, nom de la plate-forme d'apprentissage, description de la plate-forme d'apprentissage, type de la plate-forme d'apprentissage, localisation de la plate-forme d'apprentissage, version de la plate-forme d'apprentissage, identifiant du cursus pédagogique, nom du cursus pédagogique, description du cursus pédagogique, catégorie du cursus pédagogique.
indexationEvent	Idem **importationEvent**

TAB. 7.5 – Informations capturées par l'agent en fonction des actions réalisées par l'utilisateur

En intégrant cet agent dans un système d'e-formation, les informations traduisant le comportement des objets gérés au sein de ce système seront implicitement transmises à l'OP, et les instances correspondantes seront ensuite enregistrées dans le référentiel CIM par le biais de l'OM.

7.4.3.3 Le CIM *Object Manager* et le référentiel CIM

Un *Object Manager* n'est pas un composant spécifique à une application ou un environnement géré particulier, son rôle est d'assurer la consistance des informations de gestion renfermées dans le référentiel qui lui est associé, et d'offrir un ensemble de fonctions permettant d'exploiter ces informations.

Aujourd'hui, de nombreux outils conformes à l'architecture WBEM ont été mis en oeuvre par diverses organisations. Ces applications sont plus ou moins évoluées selon leur degré de maturité, mais toutes offrent la possibilité de facilement déployer un *Object Manager* auquel est associé un référentiel CIM. La connaissance de gestion, c'est-à-dire les schémas CIM existants, peuvent être directement inclus dans le référentiel, ou fournis sous la forme de fichiers MOF que le gestionnaire humain devra lui-même intégrer dans la base de connaissance à l'aide d'un compilateur dédié.

L'architecture globale de notre EIAH est donc composée d'un de ces outils qui est présenté plus en détail dans la section 9.1. L'intégration de notre modèle d'information au sein du référentiel CIM permet ainsi à l'OM associé de traiter les requêtes émises par l'interface CIM de l'OP spécifique à l'EIAH. La communication entre ces deux entités est basée sur le protocole XML/HTTP défini par le DMTF et présenté dans la section 7.2.3.

7.4.3.4 L'application de gestion

L'application de gestion permet de retrouver l'ensemble des informations de supervision contenues dans le référentiel et offre une vue homogène des instances de classes CIM.

L'application *web* que nous avons développée permet à un utilisateur final de ne consulter que les instances de notre modèle d'apprentissage, elle ne prend pas en compte les autres instances de classes CIM qui pourraient être créées par d'autres systèmes informatiques. L'IHM[12] de notre application propose quatre sections différentes qui sont toutes liées à une classe principale de notre modèle d'information :

1. La section principale est dédiée aux **ressources pédagogiques** : elle permet de retrouver l'ensemble des instances enregistrées dans le référentiel CIM et d'en afficher les propriétés.
2. Une section est dédiée aux **viviers de connaissance** : ici l'application permet de retrouver les objets pédagogiques stockés dans certains viviers de connaissance et qui ont été exploités par au moins un utilisateur.

[12] Interface Homme-Machine

3. La section consacrée aux **plates-formes d'apprentissage** permet de retrouver l'ensemble des cursus déployés au sein de chacune d'entre elles, ainsi que les ressources pédagogiques intégrées dans ces cursus.
4. Enfin, une section s'attache à proposer un **environnement personnalisé aux utilisateurs finaux**, leur offrant ainsi la possibilité de ne consulter que les objets pédagogiques qu'ils ont exploités, ou des ressource qui sont en relation avec leurs concepts ou cursus préférés.

Cette application possède une interface CIM capable de communiquer avec l'OM par l'envoi de requêtes ; celui-ci extrait les informations demandées du référentiel CIM et les retourne à l'application de gestion qui, finalement, les met en forme avant de les présenter à l'utilisateur final. Les fonctionnalités de notre application sont présentées en détail dans les sections 9.3 et 9.4 du manuscrit.

7.5 Synthèse

Nous avons proposé dans ce chapitre un cadre de supervision ouvert capable de collecter et d'exploiter les informations relatives à l'usage des différentes entités impliquées dans l'architecture de Virtualisation, et qui permet de franchir les barrières mentionnées dans les chapitres 4 et 5 :

- Le cadre de supervision offre **une vue homogène des entités hétérogènes** impliquées dans l'EIAH. La collecte d'informations de supervision émanant d'autres outils dédiés à l'exploitation de ressources pédagogiques est facilitée : seule l'intégration d'un agent au sein des outils cibles est nécessaire.
- Les informations de supervision étant transmises lors d'une action réalisée par un utilisateur, **les données contenues dans le référentiel CIM traduisent exactement les interactions** entre entités.
- **La mise à jour du référentiel CIM est effectuée en temps réel**, c'est-à-dire à chaque occurrence d'un événement particulier résultant des activités des utilisateurs.

Les propriétés et attributs définis dans les diverses classes de notre modèle d'information permettent d'obtenir différentes données statistiques comme :

- **Le nombre de consultations** des métadonnées d'un objet pédagogique particulier.
- **Le nombre de téléchargements** d'une ressource.

- **Le nombre d'intégrations** d'une ressource au sein d'un cursus d'apprentissage, ainsi que les caractéristiques de ces cursus.
- **Le nombre d'objets pédagogiques stockés dans un vivier de connaissance** particulier et ayant été exploités par au moins un utilisateur.

Ces données peuvent être exploitées différemment selon les objectifs à atteindre, et permettent notamment de sélectionner des ressources selon des critères complémentaires à ceux employés par les systèmes actuels. En effet, les objets pédagogiques peuvent être présentés aux éducateurs en fonction (a) de leur taux d'utilisation, (b) des concepts et préférences de ces éducateurs, et (c) du contexte d'apprentissage dans lequel s'effectue la recherche de ressources.

Enfin, le modèle d'information suggéré est un modèle extensible et offre la possibilité de définir d'autres classes répondant à d'autres objectifs.

Troisième partie

Expérimentations

Chapitre 8

Développement de la couche de Virtualisation des ressources pédagogiques

Sommaire

8.1 Fédération des viviers ARIADNE, MERLOT, EdNA, LRC **115**
8.1.1 Hétérogénéité des métadonnées 116
8.1.2 Accès aux ressources pédagogiques 117
8.2 Téléchargement et importation de ressources externes **118**
8.3 Indexation d'une nouvelle ressource **119**
8.4 Extraction et génération automatique de métadonnées LOM **119**
8.4.1 Indexation d'une ressource SCORM 119
8.4.2 Indexation d'un document arbitraire 121
8.5 Déploiement de la couche de Mutualisation **125**
8.5.1 INES et International E-Mi@ge 125
8.5.2 Moodle et CGIAR / IUT1A 126
8.6 Synthèse **127**

Nous exposons ici des expérimentations qui intègrent la couche de Virtualisation. La communication entre les composants de l'EIAH repose sur un ensemble de services *web*. La couche de Virtualisation étant destinée à intégrer des plates-formes d'enseignement implémentant le langage PHP, elle a été développée avec ce langage et s'appuie sur la bibliothèque NuSOAP[1]. L'architecture de Virtualisation et les technologies associées sont représentées sur la figure 8.1.

8.1 Fédération des viviers ARIADNE, MERLOT, EdNA, LRC

Nous avons expérimenté le scénario présenté dans la section 6.2.1 entre un vivier de documents et trois viviers de références, respectivement le KPS de la fondation ARIADNE et

[1] http ://dietrich.ganx4.com/nusoap/index.php

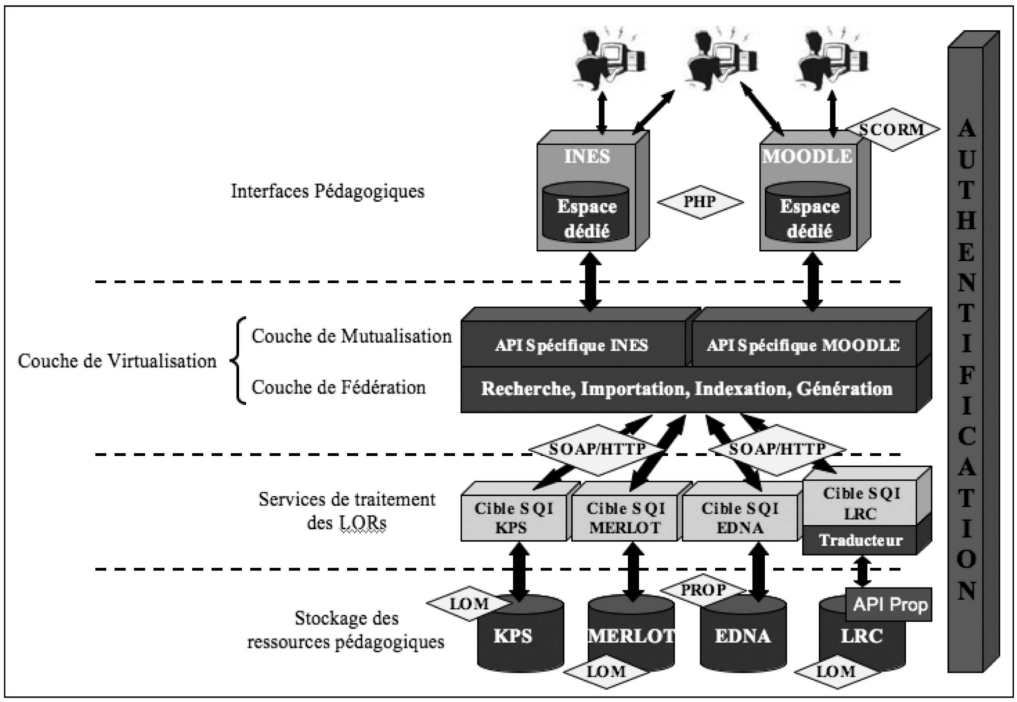


FIG. 8.1 – Déploiement de la couche de Virtualisation

les LOF MERLOT, EdNA et LRC[2]. Le LRC représente le système de stockage de l'initiative Learnet[3] dont l'objectif est de promouvoir et de faciliter le partage de ressources pédagogiques de haute qualité parmi les huit institutions regroupées dans l'University Grants Committee de Hongkong. Ces quatre viviers de connaissance sont étudiés dans le cadre de nos travaux au sein de la fondation ARIADNE et de groupes de travail définis par le réseau d'excellence Prolearn[4] dans lesquels nous sommes impliqués.

8.1.1 Hétérogénéité des métadonnées

Le KPS, MERLOT et le LRC exploitent le standard de métadonnées LOM pour décrire les ressources qu'ils renferment, tandis qu'EdNA met en oeuvre un ensemble de descripteurs propriétaire. Deux *parser* XML (la chaîne de caractères retournée par la méthode synchronousQuery() respecte ce format de données) ont donc été développés afin d'offrir à l'utilisateur final les métadonnées des quatre viviers : l'un pour retrouver les différents champs du standard LOM, l'autre pour identifier les descripteurs spécifiques à EdNA.

[2]Learning Resource Catalog - http://www.learnet.hku.hk :8050/
[3]http://learnet.hku.hk/
[4]http://www.prolearn-project.org/

8.1.2 Accès aux ressources pédagogiques

Pour être capable de rechercher des ressources dans le KPS, MERLOT et EdNA, nous invoquons des services *web* qui implémentent la spécification SQI. Nous avons donc développé un service *web* client conforme à cette spécification et qui permet de mettre en oeuvre le scénario de la figure 6.4. La figure 8.2 donne les principales lignes de code utilisées pour envoyer une requête à la cible SQI ARIADNE ; la recherche est effectuée sur le mot clé *metadata*. Dans le cas du LRC qui propose une API native constituée d'un ensemble spécifique de méthodes qui ne respectent pas SQI, nous avons mis en oeuvre une technique de correspondance largement répandue dans les systèmes informatiques [dMCS+05] et qui consiste à définir un traducteur établissant les corrélations entre les méthodes propres à l'API du LRC et celles définies par SQI (cf. figure 8.1). Ces expérimentations réalisées à partir de services de traitement hétérogènes ont permis de valider notre approche.

```
/// MODULE DE RECHERCHE DES OBJETS PEDAGOGIQUES ///

// Intégration des bibliothèques SQI
require_once("clientTarget.class.php");
require_once("sessionMgt.class.php");
// Création des services web clients
$clientTarget = new clientTarget(ARIADNESQI) ;
$sessionClient = new sessionMgt(ARIADNESESSION) ;
// Phase d'authentification auprès de la cible SQI
$sessionId = $sessionClient->createAnonymousSession() ;
// Négociation du contexte
$clientTarget->setQueryLanguage($sessionId, 'vsql') ;
$clientTarget->setResultsFormat($sessionId, 'lom') ;
// Construction de la requête
$query = '<simpleQuery><term>metadata</term></simpleQuery>' ;
// Envoi de la requête et obtention des résultats
$queryResults = $clientTarget->synchronousQuery($sessionId, $query, 0) ;
```

FIG. 8.2 – Implémentation du service de recherche en PHP

La figure 8.3 illustre le résultat de la requête précédente qui est exécutée dans les quatre viviers de connaissance fédérés ; un click sur le titre d'un document permet de consulter les métadonnées qui lui sont associées. L'interface expérimentale de recherche masque à l'utilisateur final la complexité du système mis en œuvre et propose une vue unique de toutes les ressources disponibles dans les viviers fédérés. Notre moteur de recherche inspiré de Google™ permet de réaliser le processus d'obtention d'un objet pédagogique puisqu'il permet de rechercher les métadonnées pertinentes à des critères de recherche, de les consulter, et de localiser les documents pédagogiques correspondants. Les utilisateurs finaux ont alors la possibilité de télécharger ou d'importer des ressources externes grâce aux services de la couche de Fédération.

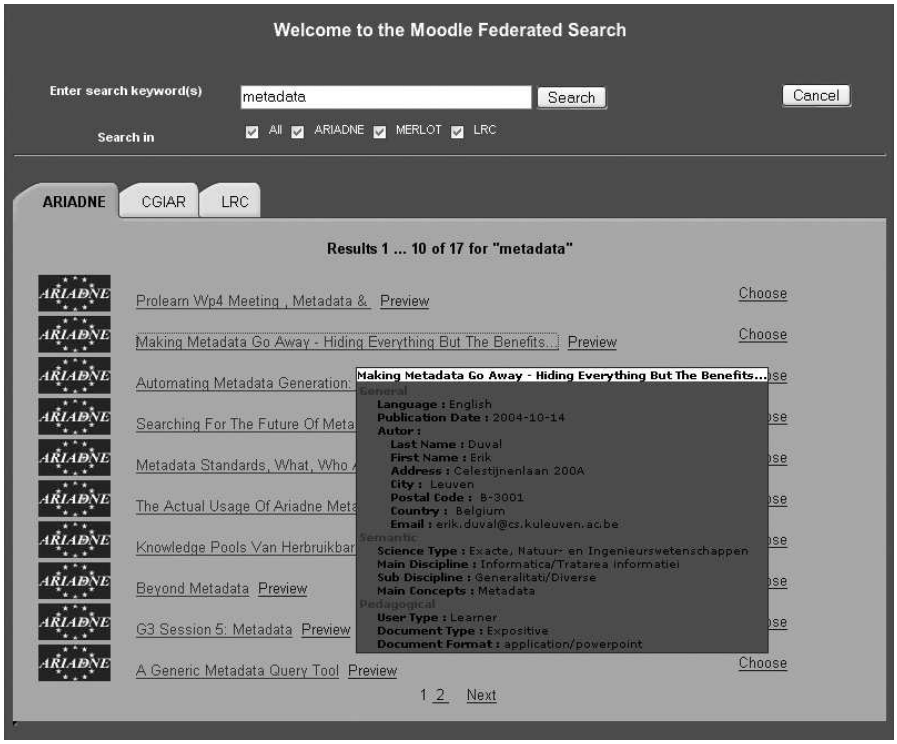


FIG. 8.3 – Interface de la couche de Fédération

8.2 Téléchargement et importation de ressources externes

Les liens "Preview" et "Choose" (cf. figure 8.3) permettent d'invoquer respectivement les services de téléchargement et d'importation d'une ressource au sein de la plate-forme dans le cursus courant. Dans le cas du LOR Ariadne, l'ensemble des métadonnées peuvent être consultées librement mais l'accès à certains documents est limité à des utilisateurs particuliers. Il est donc nécessaire de mettre en oeuvre le mécanisme d'authentification présenté dans la section 6.2.6 qui permet d'identifier l'usager du LMS afin de connaître ses droits vis-à-vis du KPS. La figure 8.4 illustre le morceau de code permettant d'obtenir les caractéristiques de l'utilisateur vis-à-vis du KPS (méthode getUserAccount), puis d'invoquer le service de téléchargement (méthode downloadDocument).

Un scénario identique est mis en oeuvre dans le cas du service d'importation, mis à part la dernière instruction qui permet d'invoquer la méthode importDocument dans le cas du KPS, ou la méthode importUrl dans le cas d'un des viviers de connaissance MERLOT, EdNA ou LRC.

```
/// MODULE DE TELECHARGEMENT D'UNE RESSOURCE EXTERNE ///

// Création du service web d'authentification client
$clientAuth = new soapclient("http://lkptest.ariadne-eu.org/webservices/authService.php") ;
// Construction des paramètres
$params = array ('userFirstName' => $fname, 'userLastName' => $lname, 'system' => 'kps');
// Invocation de la méthode getUserAccount
$userAccount = $clientAuth->call('getUserAccounnt', $params) ;
// Extraction du nom d'utilisateur et du mot de passe
$login = $userAccount[0] ;
$pwd = $userAccount[1] ;
// Création du service web client de téléchargement
$downloadClient = new soapclient("http://lkptest.ariadne-eu.org/webservices/downloadService.php") ;
// Construction des paramètres
$params = array ('docIdentifier' => $idResource, 'login' => $login, 'pwd' => $pwd) ;
// Invocation de la méthode downloadDocument
$file = $downloadClient->call('downloadDocument', $params) ;
```

FIG. 8.4 – Implémentation du service de téléchargement en PHP

8.3 Indexation d'une nouvelle ressource

Nos expérimentations relatives à l'indexation de nouveaux objets pédagogiques reposent sur la collaboration entre différents LMS (INES et Moodle) détaillés dans la section 8.5 et le vivier de connaissance ARIADNE (le KPS).

L'interface du service d'indexation permet de sélectionner le nouveau document sur le disque local de l'utilisateur, et propose ensuite un formulaire correspondant au profil de métadonnées ARIADNE conforme au standard IEEE-LOM afin d'être en mesure d'insérer le nouveau document au sein du KPS ; notons que si le document est inséré à la fois dans le LMS et dans le KPS, les métadonnées ne sont stockées que dans le KPS. La figure 8.5 présente les principales instructions réalisées pour indexer au nouvel objet pédagogique au sein du KPS.

Ainsi l'indexeur doit renseigner au minimum un ensemble de dix huit valeurs obligatoires pour l'indexation de la nouvelle ressource ; nous proposons donc une aide à ce processus en générant la quasi-totalité des métadonnées obligatoires à l'indexation d'un nouvel objet pédagogique au sein du KPS.

8.4 Extraction et génération automatique de métadonnées LOM

8.4.1 Indexation d'une ressource SCORM

Lorsque les utilisateurs insèrent de nouvelles activités SCORM au sein du LMS, il est possible d'extraire l'ensemble des métadonnées qui décrivent la ressource SCORM selon le standard LOM de l'IEEE en analysant le fichier XML *imsmanifest.xml* contenu dans le package

```
// Récupération des métadonnées renseignées dans le formulaire
$metadata = array (
             "docLanguage" => $chooseLanguage,
             "metadataLanguage" => $chooseLanguage,
             "title" => $docTitle,
             "lastName" => $lastName,
             "firstName" => $firstName,
             "datePub" => $datePub,
             "rights" => $rights,
             "scienceType" => $scienceType,
             "mainDiscipline" => $mainDiscipline,
             "subDiscipline" => $subDiscipline,
             ...
             "installationNotes" => $installationNotes,
             "otherConstraints" => $otherConstraints) ;
// Création du service web client d'indexation
$insertClient = new soapclient("http://lkptest.ariadne-eu.org/webservices/insertService.php") ;
// Construction des paramètres
$params = array ('document' => $tmpFile, 'metadata' => $metadata, 'folderLocation' => $rep, 'login' =>
                 $login, 'pwd' => $pwd) ;
// Invocation de la méthode indexLearningObject
$resultInsert = $insertClient->call('indextLearningObject', $params) ;
```

FIG. 8.5 – Implémentation du service d'indexation en PHP

et qui renferme les métadonnées décrivant le package. La figure 8.6 décrit l'appel des services d'extraction des métadonnées contenues dans un fichier *imsmanifest.xml*.

```
/// MODULE D'EXTRACTION DES MÉTADONNÉES D'UN PACKAGE SCORM ///

// Intégration de la bibliothèque contenant les méthodes d'extraction SCORM
require_once("lib/scormExtractor.class.php") ;
// Création de l'objet scormExtractor
$scormExt = new scormExtractor() ;
// Récupération du fichier imsmanifest.xml
$archive = new PclZip($uploadedFile) ;
$fileList = $archive->listContent() ;
for ($i = 0 ; $i < sizeof($fileList) ; $i++)
  {
   if ($listFile[$i]['filename'] == "imsmanifest.xml")
     {
       $imsFile = $archive->extract(PCLZIP_OPT_BY_NAME, $fileList[$i]["stored_filename"],
                                    PCLZIP_OPT_PATH, "tmp") ;
     }
  }
// Extraction des metadonnées
$metadata = $scormExt->extractScormMetadata($imsFile) ;
// Extraction des informations relatives à l'auteur
$author = $scormExt->getAuthorInfo($metadata['$lomLifeCycleContributeCentity']) ;
// Affectation des valeurs retrouvées aux différents champs du formulaire
...
```

FIG. 8.6 – Implémentation du service d'extraction des métadonnées d'un package SCORM en PHP

Ainsi, le formulaire électronique est automatiquement et entièrement rempli (cf. figure 8.7), et les objets pédagogiques SCORM peuvent être immédiatement indexés dans le KPS

en invoquant le service d'indexation.

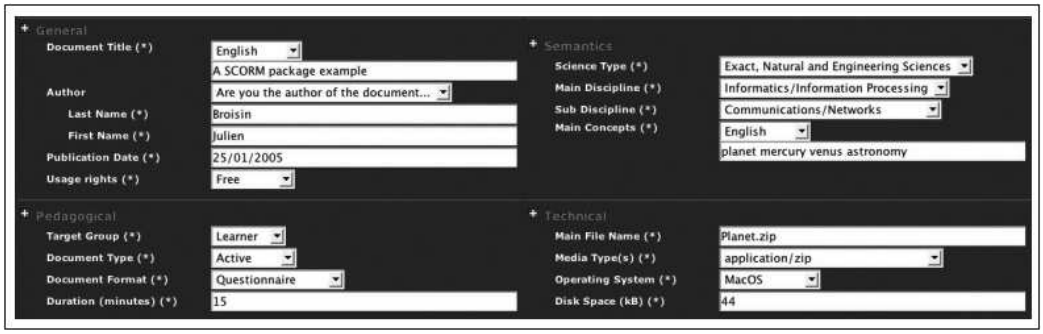


FIG. 8.7 – Résultat du service d'extraction de métadonnées à partir d'un fichier *imsmanifest.xml*

Les métadonnées spécifiées pour indexer le package SCORM au sein du KPS coïncident avec celles définies dans le fichier *imsmanifest.xml*. Ainsi, l'extraction de métadonnées :

- **Inhibe les interprétations et erreurs humaines** : alors que les indexeurs humains ne décrivent pas nécessairement les métadonnées comme le font les auteurs, elle assure la consistance des métadonnées en ne prenant en compte que celles générées par les auteurs.
- **Associe une description unique à un document unique** : il n'est pas rare de trouver différentes descriptions pour un même document.

Toutefois, un fichier *imsmanifest.xml* ne renferme pas nécessairement l'ensemble des métadonnées LOM qui représentent un élément optionnel. Dans ce cas, le package SCORM doit être traité comme un document arbitraire.

8.4.2 Indexation d'un document arbitraire

A partir des diverses informations vitales pour le bon fonctionnement d'un LMS, il est possible de produire automatiquement certaines métadonnées obligatoires à l'indexation d'un document dans le Knowledge Pool System. Cette tâche consiste à extraire des mots clés issus de l'environnement d'apprentissage du LMS et à les associer aux métadonnées du profil ARIADNE-LOM réparties en quatre catégories : Général, Sémantique, Pédagogique et Technique (cf. annexe A.2). Nous détaillons ici la génération des métadonnées obligatoires même si quelques descripteurs optionnels peuvent être produits.

8.4.2.1 Métadonnées générales

Si l'indexeur est aussi l'auteur de la ressource, il est possible de collecter ses nom et prénom à partir de son profil dans le LMS. En effet, une base de données dédiée à la gestion

des utilisateurs est souvent présente au sein des plates-formes d'apprentissage, comme c'est le cas pour les LMS déployés dans le cadre de nos expérimentations. De plus, des métadonnées optionnelles telles que l'email, le téléphone ou le pays peuvent souvent être retrouvées et appliquées au profil ARIADNE-LOM. D'autre part, le champ *Langue* des métadonnées peut être déduit de la langue courante utilisée dans le LMS. Selon la politique définie au sein de l'EIAH, les *Droits d'usage* prennent pour valeur "Libre" ou "Restreint". D'autres métadonnées générées automatiquement ne sont pas liées au contexte d'apprentissage. En supprimant l'extension du fichier, le *Titre du document* peut être déduit à partir du nom du fichier correspondant à la ressource à indexer, alors que la *Date de publication* prend pour valeur la date du jour.

8.4.2.2 Métadonnées sémantiques

Cet ensemble de métadonnées définit quatre champs obligatoires : *Type de Science*, *Discipline Principale*, *Discipline Secondaire*, et *Concepts Principaux*. Le profil ARIADNE-LOM fournit des listes prédéfinies de choix pour les trois premières métadonnées ci-dessus, offrant ainsi la possibilité d'associer les métadonnées au contexte d'apprentissage. En effet, la formation ou catégorie en cours peut facilement être extraite du contexte du LMS et être associée aux deux métadonnées *Type de Science* et *Discipline Principale*. Par exemple, la catégorie "Informatique" ou la formation "Doctorat en Informatique" doit être associée au type de science "Sciences Exactes, Naturelles, et de l'Ingénieur", mais aussi à la discipline principale "Traitement de l'Informations / Informatique". Finalement, la *Discipline Secondaire* peut être déduite à partir de la combinaison des noms de la catégorie et du cursus courants. Ces associations sont stockées dans un fichier XML (cf. figure 8.8) qui décrit tous les types de science, disciplines principales et disciplines secondaires définies dans le profil ARIADNE-LOM. Enfin, les mots clés extraits des descriptions de la catégorie et du cursus courants à partir de la méthode getContextInfo sont associés à la métadonnée *Concepts Principaux*. Ainsi l'ensemble des métadonnées Sémantiques obligatoires sont automatiquement produites.

8.4.2.3 Métadonnées pédagogiques

Le *Groupe cible* peut être déduit à partir du rôle de l'utilisateur au sein du LMS : lorsque le rôle représente un enseignant, nous pouvons supposer que la nouvelle ressource constitue une brique du cursus courant et qu'elle est destinée à des apprenants, alors que si le rôle de l'utilisateur décrit un apprenant, nous pouvons supposer que la nouvelle ressource désigne un devoir à rendre et qu'elle est adressée à des enseignants. Ce raisonnement ne peut être appliqué dans le cas d'un travail collaboratif, mais l'utilisateur a la possibilité de modifier les métadonnées générées automatiquement.

```
<Classification>
  <ScienceTypes>
    <ExactNaturalAndEngineeringSciences>
      <category>Informatique</category>
    </ExactNaturalAndEngineeringSciences>
  </ScienceTypes>
  <MainDiscipline>
    <InformaticsInformationProcessing>
      <category>Informatique</category>
    </InformaticsInformationProcessing>
  </MainDiscipline>
  <SubDiscipline>
    <Networks>
      <item>
        <category>Informatique</category>
        <course>Architecture Réseau</course>
      </item>
    </Networks>
  </SubDiscipline>
</Classification>
```

FIG. 8.8 – Extrait du fichier XML pour la génération automatique de métadonnées Sémantiques

```
function getDocType($extension, $language)
  {
   $assocArray = array ("gif" => array ("mime"=>"image/gif", "docType"=>array
("en"=>"Expositive","fr"=>"Expositif"), "docFormat"=>array("en"=>"Photograph","fr"=>"Photographie"),
"operatingSystem"=>"Multi-OS"),
                        "html" => array ("mime"=>"text/html", "docType"=>array
("en"=>"Expositive","fr"=>"Expositif"), "docFormat"=>array("en"=>"Hypertext","fr"=>"Hypertexte"),
"operatingSystem"=>"Multi-OS"),
                        "txt" => array ("mime"=>"text/plain", "docType"=>array
("en"=>"Expositive","fr"=>"Expositif"), "docFormat"=>array("en"=>"Narrative Text","fr"=>"Texte"),
"operatingSystem"=>"Multi-OS"),
                        "pdf" => array ("mime"=>"application/pdf", "docType"=>array
("en"=>"Expositive","fr"=>"Expositif"), "docFormat"=>array("en"=>"Narrative Text","fr"=>"Texte"),
"operatingSystem"=>"Multi-OS"),
                        ...
   return $assocArray[strtolower($extension)]['docType'][$language];
  }
```

FIG. 8.9 – Extrait de la bibliothèque exploitant l'extension d'un fichier

La librairie évoquée dans la section 6.2.5 dont un extrait est illustré sur la figure 8.9 permet de connaître, à partir d'une extension particulière, le type mime, le type de document, le format de document, et le système d'exploitation associés à cette extension. Ainsi d'autres métadonnées obligatoires peuvent être déduites :

- un fichier texte ou un fichier PDF indique un document expositif, alors qu'un fichier EXE décrit un document actif ; l'une des valeurs "Actif" ou "Expositif" peut donc être automatiquement affectée au *Type de document*.
- de nombreux types mime sont largement répandus et peuvent être associés à un format

de document. Par exemple, un fichier TXT représente un fichier ne contenant que du texte et la métadonnée *Format de document* prendra pour valeur "Texte narratif".

La seule métadonnée pour laquelle nous ne pouvons produire une valeur pertinente est la *Durée pédagogique*. En effet, ce descripteur est lié au contenu du document lui-même ainsi qu'au niveau d'apprentissage des apprenants.

8.4.2.4 Métadonnées techniques

La dernière section de la classification définit quatre métadonnées obligatoires qui peuvent toutes être produites à partir du fichier lui-même :

- le *Nom principal* du fichier correspond au nom du fichier représentant la ressource à indexer.
- le *Type mime* et le *Système d'exploitation* sont obtenus à partir de l'extension du fichier et de la librairie mentionnée dans la section précédente.
- l'*Espace disque* nécessaire correspond à la taille du document.

8.4.2.5 Synthèse

Si l'indexeur est aussi l'auteur du document et à partir d'un certain nombre de conventions récapitulées dans le tableau 8.1, notre EIAH permet de générer automatiquement dix sept des dix huit métadonnées nécessaires à l'insertion d'un objet pédagogique dans le KPS.

Métadonnée	*Convention*
Nom et prénom de l'utilisateur	Son profil au sein du LMS
Langue	La langue courante au sein du LMS
Droit d'usage	Selon la politique définie au sein de l'EIAH
Titre du document	Le nom du fichier sans son extension
Date de publication	La date du jour
Concepts principaux	Mots-clés extraits du contexte du LMS
Groupe cible	A partir du rôle de l'utilisateur au sein du LMS
Type de document	Selon l'extension du fichier
Format de document	Selon l'extension du fichier
Nom principal	Le nom du fichier avec son extension
Type mime	Selon l'extension du fichier
Système d'exploitation	Selon l'extension du fichier
Espace disque	La taille du fichier

TAB. 8.1 – Récapitulatif des conventions définies

Alors que la tâche d'indexation de documents peut être longue et parfois difficile à réaliser dans certains contextes, elle représente un processus trivial dans notre EIAH et encourage les

indexeurs humains à créer de réels objets pédagogiques plus faciles à retrouver et à réutiliser : notre service d'indexation offre un moyen pour permettre aux métadonnées d'exister et de rendre ainsi le processus de partage et réutilisation réalisable. Les efforts déployés dans la création d'une interface de démonstration, c'est-à-dire huit jours de développement d'un unique programmeur, sont insignifiants comparés aux bénéfices apportés par un service de génération automatique. En effet, même si les métadonnées produites automatiquement ne sont pas aussi satisfaisantes que celles générées par les indexeurs humains, elles n'ont pas besoin d'être parfaites mais suffisamment justes pour permettre le partage et la réutilisation [DH04].

8.5 Déploiement de la couche de Mutualisation

8.5.1 INES et International E-Mi@ge

La première expérimentation avec la plate-forme INES[5] [CSPB03], un système de gestion d'apprentissage *open source* développé par l'Université de Picardie d'Amiens, a été motivée par les besoins exprimés par le projet International E-Mi@ge (IEM) [CM04]. IEM, issu des appels à propositions Campus Numériques Français de la Direction de la Technologie des années 2000, 2001 et 2002, est un programme de mise à distance de la filière d'enseignement MIAGE[6].

La couche de Mutualisation apparaît dans l'espace de travail dédié à la gestion des ressources de la plate-forme qui est illustré par la figure 8.10.

Fig. 8.10 – Déploiement de la couche d'Intégration au sein de la plate-forme INES

Les quatre variables nécessaires à la réalisation des services de la couche de Fédération ont facilement été acquises :

- **Les informations des utilisateurs** sont stockées dans la table *personnels* de la base de données principale d'INES.

[5] INteractive E-learning management System
[6] Méthodes Informatiques Appliquées à la Gestion des Entreprises

- Le **répertoire racine** dans lequel sont stockés les documents pédagogiques est le même pour chaque instance d'INES, il se nomme *foadF*.
- L'**identifiant du répertoire de la formation** en cours apparaît dans la barre d'adresse du navigateur sous le nom *VNForm*.
- Le **champ du formulaire** destiné à recevoir la valeur de la localisation de la ressource pédagogique a été précisée par nos soins.

L'organisation architecturale d'IEM, dédiée à une exploitation en commun par les membres du consortium IEM et des partenaires, est constituée de deux composants majeurs : un unique vivier de connaissance géré par l'Université Paul Sabatier (UPS - Toulouse 3) dispose des dernières versions des contenus pédagogiques, et une instance de la plate-forme INES pour chaque centre d'exploitation. Le vivier de connaissance est en fait celui de la fondation ARIADNE : un accord entre le consortium IEM et la fondation ARIADNE a été conclu en 2004. Jusqu'à la réalisation de nos travaux, les différents centres d'exploitation s'approvisionnaient à partir de ce vivier en téléchargeant puis en installant manuellement les documents sur leur propre plate-forme INES. La couche de Virtualisation est désormais mise en place au coeur de l'EIAH de l'UPS et permet de rechercher, d'importer des documents existants au sein de la plate-forme INES et d'indexer de nouveaux modules dans le KPS automatiquement [BVMC06]. Actuellement en phase de test, elle sera déployée dans les autres centres d'exploitation à la prochaine rentrée universitaire.

8.5.2 Moodle et CGIAR / IUT1A

Dans la deuxième expérimentation nous nous sommes intéressés au système MOODLE, une plate-forme d'enseignement *open source* largement utilisée à travers le monde.

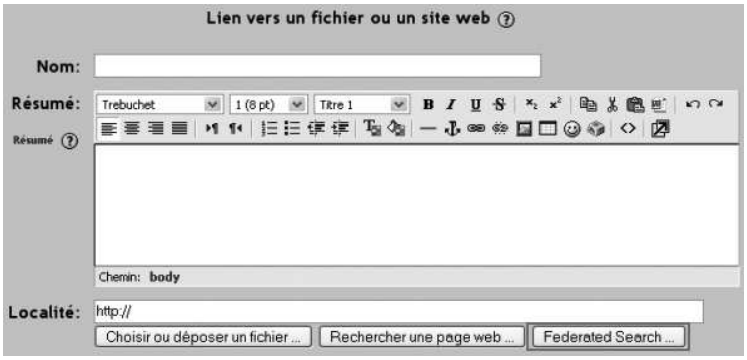


FIG. 8.11 – Déploiement de la couche d'Intégration au sein de la plate-forme Moodle

A partir de l'intégration existante et spécifique à INES, l'approche a consisté à établir

l'équivalence entre les variables spécifiques à INES et celles propres à MOODLE. Dans un premier temps, nous avons identifié l'espace de travail réservé à la gestion des ressources afin d'y insérer la couche de Mutualisation (cf. figure 8.11). Ensuite, nous avons utilisé la table *mdl_user* pour obtenir les informations de l'utilisateur connecté à MOODLE. Le répertoire racine dans lequel sont stockés les documents est défini dans le fichier de configuration de la plate-forme, tandis que l'identifiant du répertoire de la formation en cours apparaît dans la barre d'adresse du navigateur sous le nom *wdir*. Enfin, le champ du formulaire a facilement été identifié au sein du fichier HTML correspondant à l'interface graphique de la figure 8.11.

L'adaptation à la plate-forme MOODLE a nécessité un coût de développement minimal comparé aux bénéfices qu'apportent la solution proposée ; elle démontre aussi le caractère générique de notre couche de Virtualisation.

La solution MOODLE/Ariadne représente la base de l'EIAH de la Structure Universitaire de Pédagogie[7] qui a été mise en place au mois de Janvier 2005 afin de susciter, d'accompagner, et de valoriser les projets portés par des équipes d'enseignants et de favoriser de nouvelles situations d'apprentissage au sein de l'Université Paul Sabatier. Cette solution a également été adoptée par l'alliance stratégique CGIAR[8] qui s'appuie sur 15 centres internationaux de recherche agricole pour réduire la pauvreté, promouvoir le bien-être des populations, stimuler la croissance agricole, et protéger l'environnement. Aussi, l'IUT[9] 1A a choisi cette solution au début de l'année 2006 pour mettre en place un système d'e-formation au sein de la structure pédagogique.

8.6 Synthèse

Nous avons développé dans ce chapitre la couche de Virtualisation présentée dans le chapitre 6. L'ensemble des services définis au sein de la couche de Fédération ont été implémentés avec le langage PHP, et les résultats obtenus ont été exposés à travers une interface graphique simple masquant à l'utilisateur final la complexité du système global. Nous avons ensuite validé le caractère générique de notre architecture par l'intégration de la couche de Mutualisation au sein de deux plates-formes d'apprentissage distinctes et hétérogènes.

Le déploiement de la couche de Virtualisation au sein de l'ensemble des centres d'exploitation du projet IEM devrait se dérouler dans le courant de l'été à venir. Ces travaux vont faciliter l'acquisition des dernières versions des différents modules de la formation qui sont en perpétuelle évolution ; en effet, de nombreux auteurs collaborent pour produire un module global et son contenu est sujet à de régulières modifications.

[7] http ://sup.ups-tlse.fr
[8] Consultative Group on International Agricultural Research - http://www.cgiar.org
[9] Institut Universitaire de Technologie

D'autre part, nos travaux effectués avec la plate-forme MOODLE pourraient être valorisés par l'intégration de la couche de Virtualisation dans la prochaine version 1.6 de ce système d'apprentissage. En effet, suite à une présentation de nos travaux devant des acteurs impliqués dans le développement et le maintient de cette plate-forme, des discussions relatives à l'incorporation de nos développements sont en cours et pourraient se concrétiser lors de la prochaine rencontre des utilisateurs français de cette plate-forme en Juin prochain[10]. Cette contribution permettrait de sensibiliser un nombre très important d'utilisateurs à la nécessité de partager et de réutiliser les documents pédagogiques tout en leur facilitant l'accès à des ressources de natures variées.

[10]http://conferences.enst-bretagne.fr/moodlemoot/

Chapitre 9

Développement d'une plate-forme de supervision des ressources pédagogiques

Sommaire

9.1 Les technologies retenues **130**
9.2 Scénarios de gestion **131**
9.3 Usage des entités gérées **132**
9.3.1 Les objets pédagogiques 132
9.3.2 Les viviers de connaissance 133
9.3.3 Les plates-formes d'apprentissage 135
9.4 Recommandations personnalisées de ressources pédagogiques **136**
9.4.1 Recommandation basée sur les concepts 136
9.4.2 Recommandation basée sur les disciplines 137
9.4.3 Recommandation basée sur les labels et commentaires 138
9.4.4 Evaluation des ressources pédagogiques 139
9.5 Synthèse **139**

Dans ce chapitre nous présentons l'outil de supervision LOMA[1] que nous avons développé afin de permettre l'observation de l'usage des ressources pédagogiques exploitées dans l'architecture de Virtualisation.

Nous exposons dans un premier temps les technologies et solutions existantes intégrées dans notre cadre de supervision qui permettent de prendre en compte les activités d'autres systèmes d'apprentissage. Dans un second temps, nous détaillons les différents événements qui induisent des créations/modifications de certaines classes présentées dans le chapitre 7. Les deux dernières sections présentent les services qui sont intégrés dans notre application graphique de supervision : le premier offre des statistiques relatives à l'utilisation des ressources pédagogiques mais également des autres composants de l'architecture de Virtualisation, alors que le second

[1] Learning Object MAnagement

service recommande aux utilisateurs finaux des ressources pédagogiques correspondant à leurs préférences.

9.1 Les technologies retenues

Sur la base du modèle informationnel spécifié en section 7.3, nous devons réaliser :

1. **Un CIM *Object Manager*** capable d'effectuer des opérations de gestion sur les informations.
2. **Un CIM *Object Provider*** capable d'intégrer notre EIAH.
3. **Un agent** d'instrumentation capable de remonter les informations au plus proche des éléments gérés.
4. **Une interface graphique orientée utilisateur** capable d'offrir une vue des informations.

Parmi les outils existants compatibles avec l'architecture WBEM présentée dans la section 7.4.1 tels que OpenPegasus[2] ou OpenWBEM[3], le projet WBEM Services[4] est un effort *open source* visant le développement d'une implémentation Java™ de WBEM pour des applications commerciales ou non. Ce projet consiste en un ensemble d'API et d'applications client/serveur basées sur la spécification JSR 48[5], et permet le déploiement aisé d'un CIM *Object Manager*. L'outil WBEM Services, développé par Sun Microsystems™, a été déployé au sein de notre cadre de supervision.

Cet outil fournit également un compilateur qui permet d'intégrer n'importe quel fichier respectant la spécification MOF au sein du référentiel de gestion CIM. Afin d'y intégrer notre modèle d'information (cf. figure 7.4), nous avons décrit chaque classe contenant le préfixe EIAH selon cette spécification avant de compiler l'ensemble de ces fichiers dans le référentiel. Le CIMOM est alors capable de manipuler des classes et instances spécifiques à un environnement informatique pour l'apprentissage humain.

D'autre part, WBEM Services fournit des recommandations pour mettre en oeuvre un CIM *Object Provider*. A partir de celles-ci, nous avons développé en langage Java™ un OP spécifique à l'EIAH. Celui-ci possède deux interfaces distinctes : l'une, qui joue le rôle de serveur, se présente sous la forme d'un service *web* dédié à la réception (1) d'informations transmises

[2]http://www.openpegasus.org/
[3]http://openwbem.sourceforge.net/
[4]http://wbemservices.sourceforge.net/
[5]Java Specification Request - http://jcp.org/en/jsr/detail?id=48

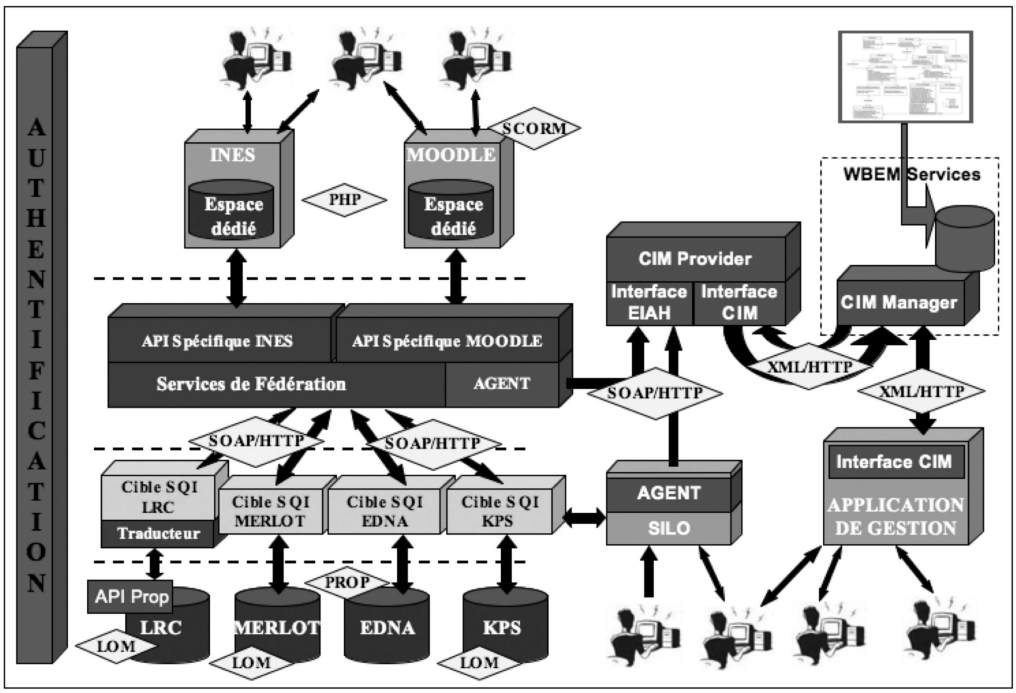


FIG. 9.1 – Un cadre de supervision des ressources pédagogiques

par les agents intégrés dans les éléments à gérer (méthode receiveData), et (2) de requêtes d'authentifcation émises par les services de la couche de Fédération (méthode getUserAccount), alors que l'autre, servant, fournit ces informations à l'OM sous la forme d'instances de classes CIM. La première interface implémente les protocoles standards SOAP/HTTP, alors que la deuxième met en oeuvre le protocole standard CIM présenté dans la section 7.2.3 pour acheminer les instances de classe jusqu'au CIM *Object Manager*.

Enfin, nous avons mis en oeuvre deux agents fournissant les informations à l'interface serveur du CIM OP et qui sont spécifiques à un système particulier : le premier, développé en PHP, est intégré dans la couche de Virtualisation et supervise les actions exécutées à travers la couche de Fédération, alors que le second, développé en Java^{TM}, est dédié à l'observation des interactions de l'outil SILO[6] de la fondation ARIADNE.

9.2 Scénarios de gestion

Comme exposé dans le chapitre précédent, les utilisateurs ont la possibilité, à partir d'une plate-forme d'apprentissage et par le biais de la couche de Virtualisation, de :

[6]Search & Index Learning Objects

- **Rechercher** des objets pédagogiques renfermés dans des viviers de connaissance et qui correspondent à des mots clés particuliers.
- **Consulter** les métadonnées de ces objets d'apprentissage.
- **Télécharger** les documents correspondants sur leur poste local.
- **Importer** ces ressources au sein d'un cursus de la plate-forme d'apprentissage.
- **Indexer** de nouveaux objets au sein de la plate-forme (sans les métadonnées) et du vivier de connaissance cible.

Selon les actions effectuées par les utilisateurs, l'agent envoie à l'OP certaines informations pour que celui-ci crée/modifie les instances de classe CIM correspondantes avant de les acheminer à l'OM qui les insèrera dans le référentiel de connaissance CIM. Nous ne détaillons pas ici l'ensemble des créations/modifications spécifiques à chaque opération précitée et qui figurent en annexe B, nous donnons les grandes lignes des scénarios mis en oeuvre.

Qu'il s'agisse de classes de l'environnement d'apprentissage ou des utilisateurs, le CIM *Object Provider* interroge le CIM *Object Manager* pour savoir si les instances concernées par l'action exécutée par l'utilisateur sont déjà enregistrées dans le référentiel CIM. Si les instances sont retrouvées, alors les propriétés affectées par l'action sont mises à jour par le biais de requêtes envoyées au CIM OM ; si elles n'existent pas, une série d'instances d'objets gérés et d'associations sont crées puis initialisées par le CIM OP avec les valeurs transmises par l'agent avant d'être insérées dans le référentiel CIM à travers le CIM OM.

9.3 Usage des entités gérées

La section suivante présente l'application de supervision LOMA qui exploite le modèle d'information CIM spécifique à l'EIAH en offrant une vue homogène des différentes instances contenues dans le référentiel CIM qui ont été créées à partir des scénarios de supervision exposés dans la section précédente. Cette application dispose de quatre fonctionnalités principales qui sont : (1) la recherche par objets pédagogiques, (2) la recherche par viviers de connaissance, (3) la recherche par plates-formes d'apprentissage, et (4) la recherche par utilisateur. Nous ne présentons ici que les trois premières, la dernière faisant l'objet de la section 9.4.

9.3.1 Les objets pédagogiques

La recherche par objet pédagogique permet de retrouver les instances de la classe EIAH_Resource correspondant à un ou plusieurs mots clés spécifiés par l'utilisateur. La figure 9.2

donne les principales instructions nécessaires à la création de la requête et à sa transmission à l'OM.

```
/* Création d'un client HTTP pour se connecter à l'OM */
CIMNameSpace httpNameSpace = new CIMNameSpace("http://" + cimOmLoc + "//interop");
CIMClient httpClient = new CIMClient(httpNameSpace, cimUser, cimPwd);
/* Création de la requête à envoyer à l'OM */
String whereClause = "Identifier LIKE '%" + keyWords.toLowerCase() + "%'";
whereClause = whereClause + " OR ElementName LIKE '%" + keyWords.toLowerCase() + "%'";
whereClause = whereClause + " OR Description LIKE '%" + keyWords.toLowerCase() + "%'";
String query = new String("SELECT * FROM EIAH_LearningObject WHERE " + whereClause);
/* Envoi de la requête à l'OM */
CIMObjectPath loPath = new CIMObjectPath (new String("EIAH_LearningObject"));
Enumeration loEnum = httpClient.execQuery(loPath, query, "WQL");
/* Mise en forme des résultats */
...
```

FIG. 9.2 – Construction et envoi d'une requête au CIM *Object Manager*

Les résultats de la recherche peuvent être triés par nombre de consultations, téléchargements ou intégrations au sein d'un cursus d'apprentissage (cf. figure 9.3). Pour chaque ressource correspondant aux critères de recherche, l'application offre le titre (attribut ElementName), la description (attribut Description), un hyperlien conduisant au document lui-même (attribut Location), et les informations relatives aux statistiques de consultation, téléchargement ou importation (attributs ConsultationOcc, DownloadOcc et IntegrationOcc) ; les dates associées à ces caractéristiques sont aussi disponibles (attributs LastConsultation, LastDownload, LastIntegration). La colonne située à gauche de l'application de supervision donne les objets les plus récemment consultés, téléchargés et importés dans un cursus, tandis que la colonne de droite affiche les ressources les plus consultées, téléchargées et intégrées dans un cours en ligne.

En activant l'hyperlien du titre d'une ressource, l'application LOMA affiche le(s) vivier(s) de connaissance stockant la ressource (à partir des instances des classes EIAH_IsStoredBy et EIAH_ContentManagementSystem) et, le cas échéant, le(s) cursus pédagogique(s) intégrant l'objet d'apprentissage ainsi que la ou les plate(s)-forme(s) d'apprentissage déployant ces différents cursus (à partir des instances des classes CIM_Component, EIAH_Courseware, EIAH_IsDeployedBy et EIAH_LearningManagementSystem).

Enfin, les hyperliens *Tags*, *Comments* et *Ratings* permettent respectivement de consulter les différents labels, commentaires et évaluations qui ont été affectés à la ressource par les utilisateurs de l'application LOMA ; ces fonctionnalités sont précisément détaillées dans la section 9.4.

9.3.2 Les viviers de connaissance

La recherche par viviers de connaissance offre les mêmes fonctionnalités que celles de

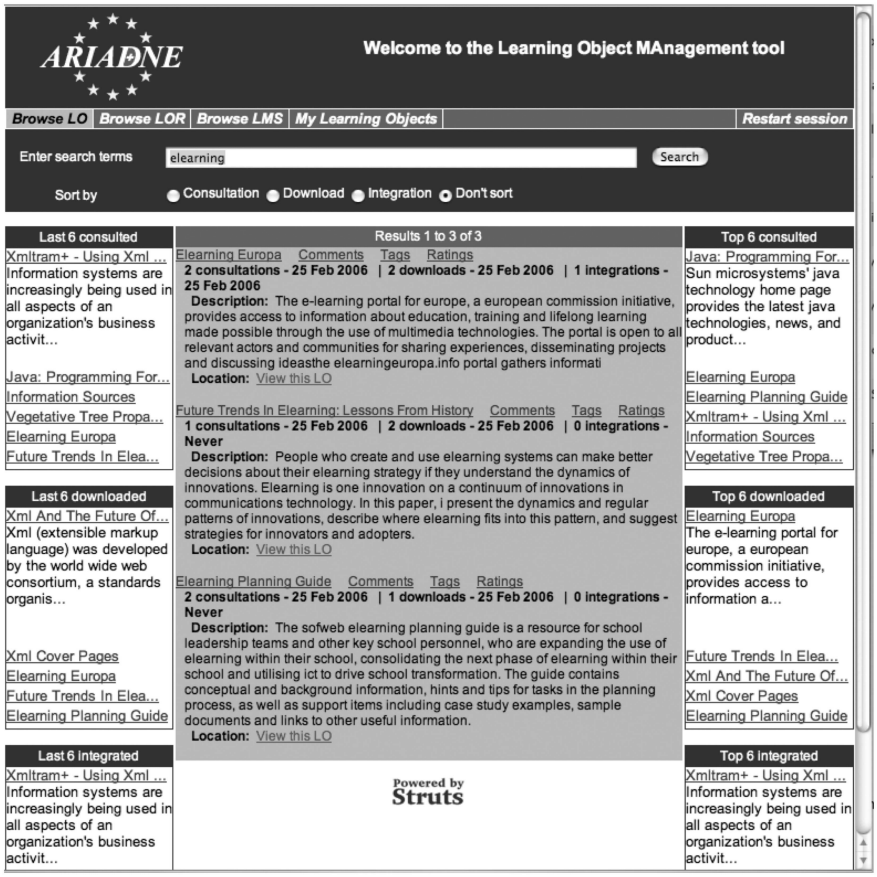


Fig. 9.3 – Usage des objets pédagogiques

la recherche par objets pédagogiques, mais offre la possibilité de rechercher des ressources stockées dans un LOR particulier. Les ressources retournées par l'application dans la section précédente sont stockées dans un (ou plusieurs) vivier de connaissance étudiés dans le cadre de l'expérimentation de virtualisation (le KPS, MERLOT, EdNA ou le LRC). Ici, l'utilisateur peut chercher les ressources renfermées dans un vivier de connaissance spécifique.

A la différence de la couche de Fédération qui retourne toutes les ressources satisfaisant les critères de recherche, l'application LOMA ne retourne que les objets pédagogiques qui ont déjà été consultés, téléchargés ou intégrés dans un cursus d'apprentissage. En effet, les instances de la classe EIAH_Resource ne sont créées dans le référentiel qu'à partir du moment où un utilisateur exploite réellement une ressource pédagogique. Par exemple, les résultats retournés par la couche de Fédération peuvent contenir des objets pédagogiques de test, c'est-à-dire des ressources qui ont été introduites dans un vivier de connaissance lors de phases de développement ; ces ressources, qui sont généralement facilement identifiables par leur titre (*Objet de test du 12/2/05*, *Test #1*, etc.), ne représentent pas de réels objets pédagogiques et ne sont pas utiles à l'apprentissage.

Nous pouvons émettre l'hypothèse que les utilisateurs de la couche de Fédération n'exploiterons pas ce type de ressources, celles-ci ne seront donc pas introduites dans le référentiel de supervision CIM.

9.3.3 Les plates-formes d'apprentissage

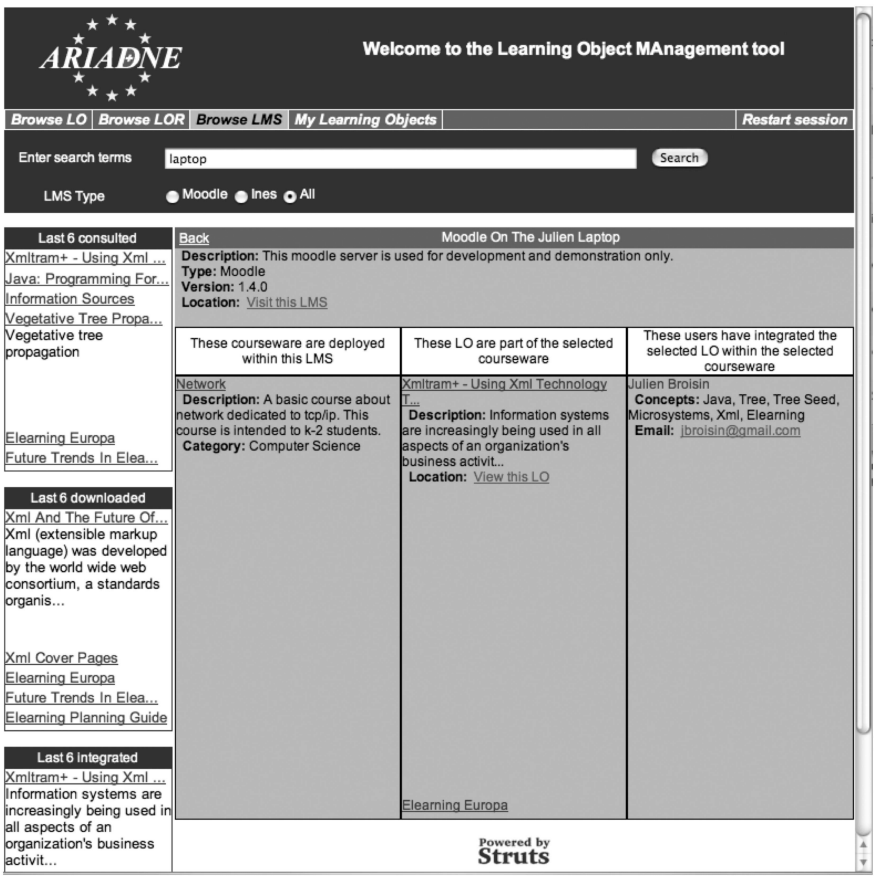


FIG. 9.4 – Recherche par plate-forme d'apprentissage

La recherche par plate-forme d'apprentissage exploite les instances des classes EIAH_LearningManagementSystem et EIAH_Courseware. Cette fonctionnalité permet d'obtenir l'ensemble des plates-formes d'apprentissage enregistrées dans le référentiel CIM, ou seulement les plates-formes de type MOODLE ou INES. Outre le titre du LMS (attribut ElementName), l'application LOMA affiche sa description (attribut Description), son rôle (attribut Roles), sa version (attribut Version) et sa localisation (attribut Location).

Un hyperlien placé sur le titre de la plate-forme permet de retrouver l'ensemble des cursus d'apprentissage déployés au sein de cette plate-forme (par le biais des instances de la classe

d'association EIAH_IsDeployedBy), les ressources pédagogiques intégrées au sein de ces cursus (par le biais des instances de la classe d'agrégation CIM_Component), ainsi que les utilisateurs ayant introduits ces ressources au sein de ces cursus (par le biais des instances de la classe d'association EIAH_HasIntegrated). La figure 9.4 illustre l'interface de l'application LOMA qui permet de visualiser l'ensemble de ces informations de supervision.

La fonctionnalité présentée ici permet donc de retrouver des objets pédagogiques en fonction d'une plate-forme d'apprentissage, et plus particulièrement d'un cursus pédagogique. Dans la mesure où la description et la catégorie des cursus sont disponibles, elle facilite l'obtention de ressources correspondant à un contexte d'apprentissage spécifique. De plus, les objets présentés étant déjà exploités dans au moins un cursus, les utilisateurs sont assurés de la pertinence et de la qualité de ces ressources.

9.4 Recommandations personnalisées de ressources pédagogiques

La figure 9.5 illustre la fonctionnalité de personnalisation de l'application LOMA. Cette partie de l'application nécessite une authentification, et propose à l'utilisateur connecté les entités d'apprentissage avec lesquelles il a coûtume de travailler. Ainsi, la colonne de gauche de la figure 9.5 expose les objets pédagogiques les plus récemment consultés, téléchargés ou intégrés dans un cursus d'apprentissage par l'utilisateur actuellement authentifié.

L'objectif des fonctionnalités présentées dans cette section est de fournir à l'utilisateur un moyen de retrouver des ressources d'apprentissage correspondant à ses préférences et domaines pédagogiques sans que celui-ci n'ait besoin de spécifier des critères de recherche ; nous souhaitons également offrir la possibilité de classifier et d'évaluer les ressources de façon personnalisée, c'est-à-dire selon le propre point de vue de l'utilisateur authentifié [BV06a].

9.4.1 Recommandation basée sur les concepts

L'attribut Descriptions de la classe EIAH_OtherPersonInformation contient l'ensemble des mots clés employés par un utilisateur pour rechercher des objets pédagogiques à l'aide de la couche de Fédération ou de l'outil SILO. L'hyperlien *My Concepts* (cf. figure 9.5) permet d'afficher l'ensemble de ces termes, ou concepts, spécifiques à l'utilisateur authentifié. L'application LOMA offre ensuite la possibilité de sélectionner un de ces concepts et délivre une liste de ressources pédagogiques en sélectionnant au sein du référentiel CIM l'ensemble des objets (pas seulement ceux exploités par l'usager connecté) correspondant au concept choisi ; pour qu'une ressource soit intégrée dans la liste retournée à l'utilisateur, le concept doit apparaître dans les attributs

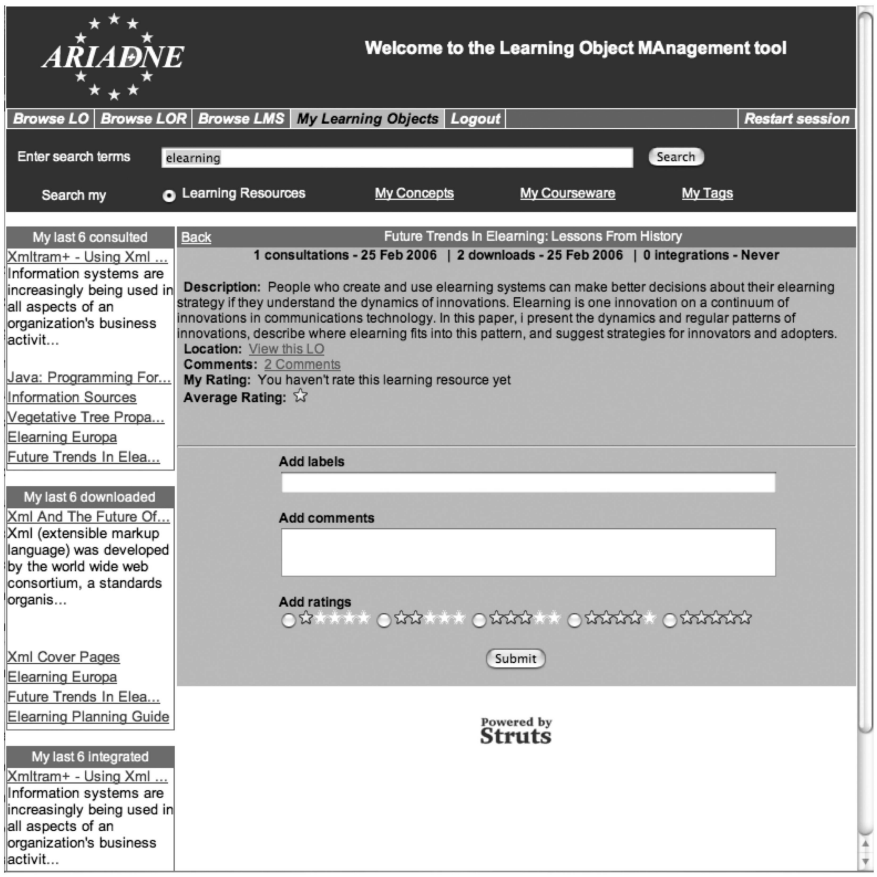


Fig. 9.5 – Un outil personnalisé pour la recherche d'objets pédagogiques

Description ou ElementName de la classe EIAH_Resource.

L'utilisateur dispose ainsi de ressources pédagogiques se rapportant à ses concepts préférés sans avoir à remplir un formulaire électronique. De plus, seulement deux clicks de souris sont nécessaires (le premier sur l'hyperlien *My Concepts*, le second sur le concept d'apprentissage) pour obtenir des ressources appropriées à la notion recherchée.

9.4.2 Recommandation basée sur les disciplines

La même approche peut être appliquée pour recommander des objets pédagogiques en rapport avec les disciplines favorites de l'utilisateur authentifié. En effet, les cursus d'apprentissage contenant des ressources importées par un utilisateur particulier peuvent être facilement retrouvés en exploitant les instances des classes CIM_Identity, EIAH_HasIntegrated, et EIAH_Courseware. A partir de ces informations, le lien *My courseware* (cf. figure 9.5) affiche l'ensemble des cours dans lesquels l'utilisateur connecté à LOMA a intégré au moins une ressource.

Deux opportunités sont ensuite offertes à l'utilisateur pour retrouver du matériel pédagogique pertinent à son contexte :

1. Un premier hyperlien placé sur le titre de chaque cursus propose une liste de toutes les ressources intégrées dans le cours sélectionné. Il n'est pas rare dans le cadre d'une formation en ligne de partager l'élaboration d'un cursus d'apprentissage entre plusieurs enseignants, alors cette fonctionnalité offre une vue de l'ensemble des objets pédagogiques intégrés dans le cursus.
2. Un autre hyperlien permet d'obtenir une liste de ressources intégrées dans d'autres cursus d'apprentissage dont les caractéristiques correspondent avec celles du cursus sélectionné, c'est-à-dire appartenant à la même catégorie (attribut Category de la classe EIAH_Courseware) ou ayant des mots clés communs (attribut Description de la même classe EIAH_Courseware).

Ici aussi, seulement deux clicks de souris sont nécessaires à l'utilisateur pour retrouver du matériel pédagogique pertinent à son contexte d'apprentissage, et aucun formulaire électronique n'a besoin d'être spécifiié.

9.4.3 Recommandation basée sur les labels et commentaires

Les valeurs des attributs Identifier, ElementName et Description de la classe EIAH_Resource sont obtenues à partir des métadonnées qui décrivent les objets pédagogiques au sein des différents viviers de connaissance. Dans la mesure où les métadonnées ne sont pas toujours renseignées par des acteurs de la formation en ligne possédant les compétences nécessaires pour réaliser cette tâche, ou par des systèmes informatiques générant des valeurs de métadonnées approximatives (comme c'est le cas de notre mécanisme proposé dans la section 8.4), celles-ci ne décrivent pas toujours les ressources pédagogiques de façon adéquate ou ne correspondent pas aux attentes d'un utilisateur particulier.

Les labels, qui peuvent être assimilés à un glossaire (même si aucune définition n'est associée à ces termes), permettent de franchir cette barrière et représentent un moyen élémentaire pour facilement classifier des informations selon un point de vue personnel. A travers l'interface de l'application LOMA (cf. figure 9.5), un utilisateur peut définir un ou plusieurs labels qui décrivent de façon appropriée un objet pédagogique selon sa propre vision. L'hyperlien *My tags* (cf. figure 9.5) expose alors la liste des différents labels spécifiés par l'utilisateur authentifié en exploitant les instances des classes EIAH_HasReviewed et EIAH_Tag, et propose ensuite deux fonctionnalités distinctes :

1. La première permet de retrouver les objets pédagogiques exploités par l'utilisateur et qui correspondent au label sélectionné. L'application LOMA exploite ici les instances des classes EIAH_HasConsulted, EIAH_Integrated, EIAH_HasIntegrated, EIAH_Resource, CIM_Identity et EIAH_IsCharacterizedBy.
2. La seconde offre l'ensemble des objets pédagogiques se rapportant au label sélectionné, mais qui ont été exploités par n'importe quel utilisateur impliqué dans notre EIAH. Ici, seules les instances des classes EIAH_Resource et EIAH_IsCharacterizedBy sont utilisées.

Cette fonctionnalité permet donc de simplement classifier les ressources pédagogiques et d'accéder rapidement à l'ensemble du vocabulaire de classification d'une part, et au matériel pédagogique se rapportant à ces labels d'autre part.

L'ensemble des métadonnées décrivant un objet pédagogique ne figure pas dans notre modèle d'information (l'attribut Location de la classe EIAH_Resource désigne la localisation de l'objet et permet de consulter sa fiche descriptive). Dans ce cas, les commentaires permettent aux utilisateurs d'ajouter leurs propres annotations pour faciliter la classification personnalisée d'une ressource, mais donnent également des indications aux autres usagers quant au sujet abordé par l'objet d'apprentissage.
LOMA exploite les instances des classes EIAH_HasReviewed, EIAH_Comment et EIAH_IsCharacterizedBy pour apporter aux utilisateurs une telle opportunité.

9.4.4 Evaluation des ressources pédagogiques

De plus en plus de systèmes basés sur les technologies Internet offrent aux utilisateurs finaux un moyen d'évaluer le contenu renfermé par ces systèmes. Cette fonctionnalité est proposée par l'application LOMA à travers les classes d'objets EIAH_Resource et EIAH_HasRated, où l'attribut Level de cette dernière indique le niveau d'évaluation attribué par un utilisateur à une ressource pédagogique.

Ainsi, un utilisateur peut trier les ressources qu'il a exploitées en fonction du niveau d'évaluation qu'il leur a attribuées, ou selon le niveau moyen d'évaluation qui prend en compte les appréciations de tous les utilisateurs du système. En effet, en sélectionnant l'ensemble des instances EIAH_HasRated référençant une ressource particulière, il est possible de calculer son estimation globale.

9.5 Synthèse

Nous avons implémenté dans ce chapitre l'infrastructure de supervision présentée dans le

chapitre 7. Les agents ont été intégrés avec succès dans la couche de Virtualisation mais également dans l'outil de recherche et d'indexation d'objets pédagogiques de la fondation ARIADNE, offrant ainsi la possibilité d'acquérir des informations statistiques issues de différents systèmes et résultant des activités des utilisateurs. De plus, notre expérimentation montre que d'autres systèmes peuvent facilement être intégrés au sein de notre cadre de supervision et valide le caractère extensible et interopérable de notre solution.

L'agent développé pour l'outil SILO est désormais intégré dans la dernière version de l'application mise à jour en Mars 2006. Le déploiement de celle-ci dans différents centres d'exploitation va d'une part apporter un nombre conséquent d'informations statistiques au sein du référentiel CIM, et risque d'autre part de soulever des problématiques liées à la surcharge de l'OM responsable de ce référentiel ; dans ce cas, une architecture constituée d'un ensemble d'OM chacun responsable d'un référentiel spécifique devra être mise en place afin de diminuer la charge de traitement de l'OM actuel, et de garantir la consistance et l'intégrité des informations de supervision.

Enfin, nous avons présenté une application graphique de supervision basée sur les technologies *web* qui propose aujourd'hui un ensemble de quatre fonctionnalités distinctes pour parcourir les informations contenues dans le référentiel. Une autre fonctionnalité en cours de réalisation vise à offrir la possibilité de retrouver les données statistiques en fonction des cursus pédagogiques enregistrés dans le référentiel. Les principaux *bugs* ont été identifiés, et une version publique sera accessible en ligne avant l'été 2006.

Conclusions et perspectives de recherche

Chapitre 10

Conclusions et perspectives de recherche

Sommaire

10.1 La fédération des viviers de connaissance **143**
10.2 La mutualisation des ressources pédagogiques **144**
10.3 La supervision des systèmes et utilisateurs impliqués dans un EIAH **145**
10.4 Des métriques pour optimiser l'exploitation des ressources **147**

Nous avons contribué à la réalisation d'un environnement informatique pour l'apprentissage humain au service de la Virtualisation et de la Gestion des ressources pédagogiques. La flexibilité du cadre de travail proposé, en partie due à l'adoption et à la facilité d'utilisation des protocoles et technologies retenus, offre une vue unique de ressources d'apprentissage stockées dans des viviers de connaissance distincts et fournit un accès facilité à celles-ci à travers les plates-formes de téléformation. Aussi, notre approche par modélisation des systèmes et utilisateurs impliqués dans un EIAH a permis de superviser l'ensemble des interactions se produisant entre les différentes entités, et de recueillir les informations statistiques liées à l'exploitation des ressources pédagogiques.

Ces travaux de recherche ont porté sur quatre axes spécifiques qui contribuent au développement de l'e-formation et qui sont : la fédération de différents viviers de connaissance, la mutualisation des ressources fédérées à travers les plates-formes d'apprentissage, la supervision des interactions entre entités, et l'exploitation des informations statistiques recueillies. Pour chacun d'entre eux nous proposons un résumé, une analyse et quelques perspectives de travaux futurs.

10.1 La fédération des viviers de connaissance

Afin de proposer un large éventail de ressources pédagogiques aux concepteurs de cur-

sus d'apprentissage et de favoriser ainsi le développement de l'e-formation, nous avons défini une couche de Fédération composée d'un ensemble de services capable de rechercher des objets pédagogiques localisés dans des viviers de connaissance hétérogènes. Cette fédération nous a permis d'augmenter considérablement la masse de matériel pédagogique disponible à partir d'un outil unique.

Toutes les ressources accessibles à travers la fédération sont décrites à l'aide de métadonnées qui respectent le plus souvent le standard LOM, mais nous avons également intégré dans la couche de Fédération un vivier de connaissance présentant un format de données propriétaire pour la description des ressources qu'il renferme. Ainsi nous avons établi une représentation homogène de ces descripteurs hétérogènes afin de proposer aux concepteurs de cursus d'apprentissage une recherche et une découverte uniformes du matériel pédagogique. Au niveau des services de traitement associés à chacun des viviers fédérés, nous avons considéré à la fois le standard SQI en cours d'élaboration et déjà déployé au sein de nombreux viviers, mais également des spécifications propriétaires comme celle proposée par l'initiative Learnet. Basée sur la technologie des services *web* qui offre un accès transparent à des systèmes distants, nous avons conçu et expérimenté avec succès une application pour rechercher des ressources pédagogiques dont les descripteurs et services associés sont hétérogènes.

Les différents acteurs du *web* ont pris conscience depuis quelques années de la nécessité d'ajouter aux ressources du *web* des informations sur leur contenu, c'est-à-dire des métadonnées. A l'origine du *web* sémantique, cette valeur ajoutée favorise une recherche plus efficace et pertinente des ressources pédagogiques. Actuellement en plein essor, une interopérabilité entre le *web* sémantique et les viviers de connaissance augmenterait de façon incommensurable le nombre de ressources disponibles. Cette fédération pourrait devenir une source vivante et grandissante de riches objets pédagogiques qui pourrait servir de fondation globale pour l'apprentissage, parallèlement au *World Wide Web* mais apportant les avantages de structuration et maîtrise de la recherche pour des domaines spécifiques métiers.

10.2 La mutualisation des ressources pédagogiques

Nous avons rendu possible la communication verticale entre les composants impliqués dans un EIAH par la mutualisation des ressources pédagogiques fédérées à travers l'usage des plates-formes d'apprentissage. Nous avons introduit une couche de Mutualisation au sein des plates-formes de téléformation leur offrant une ouverture vers la couche de Fédération, et qui leur permet d'interagir avec les viviers de connaissance tout en masquant la complexité des composants mis en jeu.

Nous avons conçu un mécanisme d'authentification capable d'identifier les caractéristiques des utilisateurs au sein de différents systèmes. Nous avons ainsi conservé le mécanisme *single sign-on* et respecté les rôles et privilèges des usagers vis-à-vis des plates-formes d'apprentissage et des viviers de connaissance intégrés dans notre EIAH. Ce système est la clef de voûte de la définition d'un ensemble de services exploitant de façon transparente, et à partir des plates-formes d'apprentissage, des ressources localisées dans des viviers de connaissance. En particulier, nous avons proposé un service d'importation offrant la possibilité d'intégrer simplement des ressources externes au sein d'un cursus pédagogique. Nous avons ensuite démontré le caractère générique de la couche de Mutualisation à travers deux expérimentations déployées avec des plates-formes d'apprentissage différentes au sein de projets nationaux et internationaux. Ces études ont montré que l'approche par couche d'abstraction peut être appliquée avec succès et réutilisée par d'autres environnements d'apprentissage.

Une perspective de nos travaux consiste à séparer les problèmes de technologies employées et l'interface de mutualisation : c'est l'approche MDA[1] [MM03]. De celle-ci, nous retenons la séparation entre une modélisation indépendante de toute plate-forme et une modélisation spécifique à une plate-forme donnée qui confère un caractère générique et donc réutilisable aux solutions. L'idée avancée dans MDA est d'utiliser des traducteurs ou passerelles pour basculer d'une modélisation PIM[2] vers une modélisation PSM[3].

10.3 La supervision des systèmes et utilisateurs impliqués dans un EIAH

La fédération des différents viviers de connaissance a mis à disposition des concepteurs de cursus d'apprentissage une masse importante d'objets pédagogiques, et offre ainsi un nombre considérable de ressources pour une requête particulière. Afin de proposer aux éducateurs du matériel pédagogique de qualité et adapté à leur contexte à partir d'informations statistiques liées à l'exploitation des objets d'apprentissage, nous avons défini un modèle d'information extensible décrivant les systèmes et utilisateurs impliqués dans un EIAH, ainsi qu'une architecture ouverte associée qui supervise l'ensemble des interactions entre ces entités.

Nous nous sommes appuyés sur le modèle d'information existant CIM en cours de standardisation et qui peut intégrer n'importe quel domaine métier ou système particulier. Le modèle pour l'EIAH que nous avons établi se focalise sur les propriétés utiles et nécessaires à l'exploitation des entités de l'environnement d'apprentissage, mais d'autres classes modélisant d'autres

[1] Model Driven Architecture
[2] Platform Independent Model
[3] Platform Specific Model

systèmes comme les outils auteurs, ou d'autres attributs visant des objectifs différents peuvent facilement être pris en compte dans notre approche. Notre modèle ne se limite donc pas aux composants intégrés dans notre EIAH et apporte une forte flexibilité qui garantit la possibilité de spécifier d'autres éléments.

Nous avons également montré comment notre architecture rend réalisable la supervision de systèmes d'apprentissage de nature hétérogène et fortement distribuée. Le protocole de transport retenu offre une interopérabilité entre éléments à gérer et composants de gestion, et assure la supervision de systèmes sur une grande échelle. L'implémentation mise en oeuvre a suggéré la centralisation des informations de supervision au sein d'un référentiel unique, mais une architecture décentralisée et constituée de plusieurs référentiels qui renferment chacun les informations liées à des systèmes ou domaines particuliers offrirait un partage de la connaissance de gestion et une délégation des différentes tâches à effectuer.

Dans l'objectif de faciliter la mise à disposition de documents pertinents aux éducateurs, une perspective de ces travaux consiste à mettre en oeuvre d'autres fonctionnalités de Gestion qui exploitent les concepts du modèle d'information CIM. Ce dernier définit des associations entre objets gérés qui expriment des relations de dépendance reliant au moins deux classes d'objet : l'une identifiée comme *Antécédent*, l'autre comme *Dépendant*. Par exemple, une relation de dépendance est intrasèquement définie entre un système (l'*Antécédent*) et un service (le *Dépendant*) hébergé par ce système. De plus, nous avons proposé dans des travaux précédents [Bro03] [SdSB+04] la possibilité d'associer des actions à effectuer sur l'objet *Dépendant* en fonction d'événements survenant sur l'objet *Antécédent*. En spécifiant des relations de dépendance entre certaines classes de notre modèle d'information, nous serons en mesure d'automatiser des actions à exécuter en fonction d'événements se produisant au sein de notre modèle, comme le propose Stephen Downes[4] avec les fils RSS[5] pour générer automatiquement un flux d'information en direction d'abonnés. Lorsqu'une nouvelle ressource pédagogique correspondant aux concepts préférés de certains éducateurs est créée au sein du référentiel de supervision, un courrier électronique pourra par exemple être envoyé aux concepteurs de cursus pédagogiques afin de les avertir de la disponibilité d'une ressource supplémentaire. La sémantique forte du modèle CIM et les travaux qui visent à ajouter de la dynamique permettraient d'aller plus loin dans le processus de conception de cursus pédagogiques.

[4] http://www.downes.ca/xml/edu_rss.htm
[5] Rich Site Summary

10.4 Des métriques pour optimiser l'exploitation des ressources

La supervision des composants d'un EIAH réalisée dans ce travail contribue à l'amélioration d'une recherche pertinente d'objets pédagogiques parmi une masse importante de ressources. Des informations statistiques telles que le nombre de consultations des métadonnées d'une ressource particulière, le nombre de téléchargements ou d'intégrations de celle-ci au sein de cursus pédagogiques déployés sur différentes plates-formes d'e-formation, sont stockées dans un référentiel dédié et peuvent être consultées et exploitées à travers une application que nous avons développée. Les différentes formes de recherche induites par ces informations ne s'appuient donc pas seulement sur des correspondances entre termes spécifiés par un utilisateur et ceux décrivant les caractéristiques d'un objet pédagogique, mais se basent sur des critères liés au taux d'utilisation d'un système ou d'une ressource ainsi que sur le contexte d'apprentissage visé. Ces nouvelles méthodes contribuent à l'obtention rapide de matériel d'apprentissage de qualité et pertinent pour un objectif pédagogique donné, et favorise l'adoption de l'e-formation par une plus grande majorité d'éducateurs. Dans cet objectif, nous avons proposé une recherche personnalisée et personnalisable de ressources pédagogiques basée sur le profil des utilisateurs finaux.

Nous avons exploité les informations statistiques dans le but de proposer aux concepteurs de cursus pédagogiques des ressources pertinentes à leurs objectifs. Cependant, les métriques pourraient être utilisées pour améliorer le processus d'apprentissage des apprenants. Dans la mesure où notre modèle d'information est extensible, des données visant à mesurer l'efficacité d'apprentissage d'un utilisateur pourraient être à la base de la mise en oeuvre de systèmes d'aide intelligents qui proposeraient aux étudiants, de façon automatisée et en fonction des métriques calculées, les ressources ou tutorats les plus appropriés à un instant donné de l'apprentissage. Enfin, la spécification d'autres métriques définies avec des experts des Sciences Humaines leur offrirait un moyen pour évaluer la qualité d'une ressource pédagogique ou de tout autre composant d'un EIAH.

Bibliographie

[AHL+02] M. Arnaud, H. Hudrisier, E. Lazarides, J. Perriault, L. Romarie, J.M. Saillant, and A. Zaoui. L'impact de la normalisation sur les dispositifs d'enseignement. Technical report, Association Française pour la NORmalisation, February 2002, http://www.gis-gemme.org/rapports/norm_standard2.pdf.

[BANK05] D. Burgos, M. Arnaud, P. Neuhauser, and R. Koper. Ims learning design : la flexibilité pédagogique au service des besoins de l'e-formation. *Revue Association Enseignement Public & Informatique*, 2005, http://www.epi.asso.fr/revue/articles/a0512c.htm.

[BBG+05] D. Burgos, N. Berbegal, D. Griffiths, C. Tattersall, and R. Koper. Ims learning design : How the specifications can change the current e-learning landscape. *E-learning World*, Issue 2, Mars-Avril 2005.

[BBH06] P. Baumgartner, S. Bobrowski, and S. Heyer. Towards a model of learning objects. *Technology, Instruction, Cognition and Learning*, Special issue : Learning Objects and Pedagogy, 2006, http://www.campuscontent.de/dokumente/upload/TICL-Model_of_LOs.pdf.

[BGC02] X. Bouchet and H. Gillarès-Calliat. *SAN et NAS - Solutions de stockage, sécurité, infrastructures*. Dunod, 2002.

[BPSM+04a] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible markup language (xml) 1.0. Technical report, World Wide Web Consortium, 2004, http://www.w3.org/TR/2004/REC-xml-20040204/.

[BPSM+04b] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible markup language (xml) 1.0 (third edition). Technical report, World Wide Web Consortium, February 2004, http://www.w3.org/TR/2004/REC-xml-20040204/.

[BR01] A.E. Barron and C. Rickelman. Management systems. *Handbook of Information technologies for education and training*, 2001.

[Bro03] J. Broisin. Fédération de systèmes de gestion pour une gestion proactive de la qualité de service applicative. Rapport de dea, Université Paul Sabatier, Septembre 2003.

[BV05] J. Broisin and P. Vidal. Un environnement informatique pour l'apprentissage humain au service de la virtualisation des objets pédagogiques. *Revue Sciences et Techniques de l'Information et de la Communication pour l'Éducation et la Formation*, 12, 2005, http://www.sticef.org.

[BV06a] J. Broisin and P. Vidal. A management framework to recommend and review learning objects in a web-based learning environment. In *6th IEEE International Conference on Advanced Learning Technologies - ICALT 2006, Kerkrade*, page To appear. IEEE Computer Society, 5-7 Juillet 2006.

[BV06b] J. Broisin and P. Vidal. A single sign-on mechanism for authenticating users across a distributed web-based learning environment. In *7th International Conference on Information Technology Based Higher Education and Training - ITHET06, Sydney*, page To appear. IEEE Computer Society, 10-13 Juillet 2006.

[BVBD05] J. Broisin, P. Vidal, P. Baqué, and E. Duval. Sharing and re-using learning objects : Learning management systems and learning object repositories. In *Educational Multimedia, Hypermedia & Telecommunications - EDMEDIA05, Montréal*, pages 4558–4565. Association for the Advancement of Computing in Education, 27 Juin - 2 Juillet 2005.

[BVD05] J. Broisin, P. Vidal, and E. Duval. A management framework for tracking learning objects' usage. In *Intelligent, Interactive, Learning Object Repositories 2005 - LORNET Research Network, Vancouver*, 15-18 Novembre 2005.

[BVMC06] J. Broisin, P. Vidal, D. Marquié, and O. Catteau. The international e-miage project : from an isolated framework to an architecture based on learning standards. In *Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services - AICT/ICIW 2006, Guadeloupe*. IEEE Computer Society, 19-25 Février 2006.

[BVMD05] J. Broisin, P. Vidal, M. Meire, and E. Duval. Bridging the gap between learning management systems and learning object repositories : exploiting learning context information. In *E-Learning on Telecommunications - ELETE05, Lisbonne*, pages 478–483. IEEE Computer Society, 17-20 Juillet 2005.

[BVS06] J. Broisin, P. Vidal, and M. Sibilla. A management framework for tracking user activities in a web-based learning environment based on a model driven approach. In *Educational Multimedia, Hypermedia & Telecommunications - EDMEDIA06,*

Orlando, page To appear. Association for the Advancement of Computing in Education, 27-30 Juin 2006.

[CCMW01] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web services description language (wsdl) 1.1. Technical report, World Wide Web Consortium, March 2001, http://www.w3.org/TR/wsdl.

[Cho02] R. Chouinard. Les normes et standards de la formation en ligne - etat des lieux et enjeux. Technical report, Groupe de travail sur les normes et standards de la formation en ligne (STIC), 2002, http://profetic.org/file/norm-0210-d-RAPPORT.pdf.

[CHV+98] K. Cardinaels, K. Hendrikx, E. Vervaet, E. Duval, H. Olivie, F. Haenni, K.Warkentyne, M. Wentland Forte, and E. Forte. A knowledge pool system of reusable pedagogical elements. In *Fourth International Conference on Computer Aided Learning and Instruction in Science and Engineering - CALISCE'98, Göteborg*, pages 54–62, 15-17 Juin 1998.

[Cla97] R. Clarke. Beyond the dublin core : Rich meta-data and convenience-of-use are compatible after all. 1997, http://www.anu.edu.au/people/Roger.Clarke/II/DublinCore.html.

[CM04] G.M. Cochard and D. Marquie. An e-learning version of the french higher educationn curriculum "computer methods for the companies management". In *18th IFIP World Computer Congress - WCC 2004, Toulouse*, pages 557–572. Kluwer Academic Publishers, 22-27 Août 2004.

[Con02] IMS Global Learning Consortium. Ims reusable definition of competency or educational objective - information model. Technical report, Instructional Management Systems, October 2002, http://www.imsglobal.org/competencies/rdceov1p0/imsrdceo_infov1p0.html.

[Con03a] IMS Global Learning Consortium. Ims digital repositories interoperability - core functions information model. Technical report, Instructional Management Systems, January 2003, http://imsglobal.org/digitalrepositories/driv1p0/imsdri_infov1p0.html.

[Con03b] IMS Global Learning Consortium. Ims learning design best practice and implementation guide : Final. Technical report, Instructional Management Systems, January 2003, http://www.imsglobal.org/learningdesign/ldv1p0/imsld_bestv1p0.html.

[Con03c] IMS Global Learning Consortium. Ims learning design binding and schema's : Final. Technical report, Instructional Management Systems, January 2003, http://www.imsglobal.org/learningdesign/ldv1p0/imsld_bindv1p0.html.

[Con03d] IMS Global Learning Consortium. Ims learning design information model : Final. Technical report, Instructional Management Systems, January 2003, http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html.

[Con04] IMS Global Learning Consortium. Ims vocabulary definition exchange best practice and implementation guide. Technical report, Instructional Management Systems, February 2004, http://www.imsglobal.org/vdex/vdexv1p0/imsvdex_bestv1p0.html.

[Con05] IMS Global Learning Consortium. Ims content packaging overview. Technical report, Instructional Management Systems, November 2005, http://imsglobal.org/content/packaging/cpv1p2pd/imscp_oviewv1p2pd.html.

[Con06] IMS Global Learning Consortium. Ims question and test interoperability overview. Technical report, Instructional Management Systems, January 2006, http://imsglobal.org/question/qtiv2p1pd/imsqti_oviewv2p1pd.html.

[CSPB03] G.M. Cochard, M. Sidir, L. Pires, and S. Baudoux. Ines, une plate-forme orientée vers la gestion des acteurs et des activités. In *Colloque Campus Numériques et Universités Numériques en région, Montpellier*, 1-2 Octobre 2003.

[Dal03] J.R. Dalziel. Implementing learning design : The learning activity management system (lams). In *20th Annual Conference of the Australian Society for Computers in Learning in Tertiary Education, Adélaïde*, pages 593–596. G. Crisp, D. Thiele, I. Scholten, S. Barker & J. Baron, 7-10 Décembre 2003.

[Dav03] J.P. David. Modélisation et production d'objets pédagogiques. *Revue Sciences et Techniques Educatives*, Numéro spécial "Ressources Numériques, XML et Education" :69–104, 2003.

[DCR05] A. Derycke, V. Chevrin, and J. Rouillard. Intermédiations multicanales et multimodales pour l'e-formation : l'architecture du projet ubi-learn. In *Environnement Informatique pour l'Apprentissage Humain, Montpellier*, pages 407–412. Institut National de Recherche Pédagogique, 25-27 Mai 2005.

[DFC+01] E. Duval, E. Forte, K. Cardinaels, B. Verhoeven, V. Durm, K. Hendrikx, M.W. Forte, N. Ebel, M. Macowicz, K. Warkentyne, and F. Haenni. The ariadne knowledge pool system. *Association for Computing Machinery*, 44(5) :72–78, Mai 2001.

[DH98] F. Dawson and T. Howes. vcard mime directory profile. Technical report, Internet Engineering Task Force, September 1998, http://www.ietf.org/rfc/rfc2426.txt.

[DH04] E. Duval and W. Hodgins. Making metadata go away. hiding everything but the benefits. In *Dublin Core Conference, Shangaï*, pages 29–35. W. Jianzhong, 4 Octobre 2004, http://purl.org/metadataresearch/dcconf2004/papers/Paper_15.pdf.

[DHSW02] E. Duval, W. Hodgins, S. Sutton, and S.L. Weibel. Metadata principles and practicalities. *D-Lib Magazine*, 8(4), Avril 2002, http://www.dlib.org/dlib/april02/weibel/04weibel.html.

[DK05] L. DeMichiel and M. Keith. Ejb core contracts and requirements. Technical report, Sun Microsystems, June 2005, http://www.mpowerit.com/ejb-3_0-pr-spec-simplified-editable.pdf.

[dLPJ04] B. de La Passardière and P. Jarraud. Manuel, un profil d'application du lom pour c@mpusciences. *Revue Sciences et Techniques de l'Information et de la Communication pour l'Éducation et la Formation*, 11 :11–57, 2004, http://www.sticef.org.

[dMCS+05] S. L. de Moura, F. Coutinho, S. W. M. Siqueira, R. N. Melo, and S. V. Nunes Jr. Integrating repositories of learning objects using web-services to implement mediators and wrappers. In *International Conference on Next Generation Web Services Practices - NWeSP'05, Séoul*, pages 82–87. IEEE Computer Society, 22-26 Août 2005.

[DMH+00] S. Decker, S. Melnik, F. Van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks. The semantic web : The roles of xml and rdf. *IEEE Internet Computing*, 15(3) :63–74, October 2000.

[Duv04] E. Duval. Ariadne - a european learning (object) community. In *International Symposium Networks without Borders : Towards Cross-cultural Learning Communities*, pages 238–244. K. Konde, 2004.

[DVV+00] E. Duval, E. Vervaet, B. Verhoeven, K. Hendrikx, K. Cardinaels, H. Olivié, E. Forte, F. Haenni, K. Warkentyne, and M. Wentland Forte. Managing digital educational resources with the ariadne metadata system. *Journal of Internet Cataloging : The International Quarterly of Digital Organization, Classification, and AccessThe International Quarterly of Digital Organization, Classification, and Access*, 3(2/3) :145–173, Novembre 2000.

[Eic94] D. Eichmann. The rbse spider - balancing effective search against web load. In *First International World Wide Web Conference, Genève*, pages 113–120. Elsevier Science BV, 25-27 Mai 1994, http://mingo.info-science.uiowa.edu/eichmann/www94/Spider.ps.

[EK01] W. Emmerich and N. Kaveh. Component technologies : Java beans, com, corba, rmi, ejb and the corba component model. In *8th European Software Engineering Conference, Vienne*, pages 311–312. ACM Press, 10-14 Septembre 2001.

[Ell00] C. Eller. *An Associative Repository for the Administration of Course Material.* PhD thesis, Institute for Information processing and Computer supported new Media, 2000, http://www.iicm.edu/thesis/celler.pdf.

[Fae01] R. Faerber. Accompagner les apprentissages à distance et collaborer en petits groupes. In *Quatrième congrès de l'Association des Chercheurs et Enseignants en Science de l'Education - AECSE 2001, Villeneuve d'Ascq*, 5-8 Septembre 2001, http://faerber.u-strasbg.fr/publi/aecse_faerber.PDF.

[FGM+99] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol - http/1.1. Technical report, World Wide Web Consortium, June 1999, http://www.w3.org/Protocols/rfc2616/rfc2616.html.

[Fie94] R. Fielding. Maintaining distributed hypertext infostructures : Welcome to momspider's web. In *First International World Wide Web Conference, Genève*, pages 193–204. Elsevier Science BV, 25-27 Mai 1994, http://citeseer.ist.psu.edu/fielding94maintaining.html.

[FM04] K. Friesen and N. Mazloumi. Integration of learning management systems and web applications using web services. *Advanced Technology for Learning*, 1(1) :16–24, 2004.

[FME+02] E. Forte, M. Macowicz, N. Ebel, K. Warkentyne, F. Haenni, E. Duval, T. Cleenewerck, K. Cardinaels, B. Verhoeven, R. Van Durm, F. Neven, S. Ternier, M. Wentland Forte, E. Fernandes, F. Simillion, B. Zaerpour, J.P. Peyrin, J.Ph. Pernin, J.M. Cagnat, J.P. David, A. Cogne, C. Guilloux, Ph. Vidal, Ph. Baqué, H. Apcher, and A. Alibert. Guided tour of the ariadne environment : Tools and methods. In *Technologies of Information and Communication in Education - TICE02, Lyon*, pages 353–357. INSA Lyon, 13-15 Novembre 2002.

[FMMW05] E. Fernandes, H. Madhour, S. Miniaoui, and M. Wentland. Phoenix : An xml-based tool for e-learning environment. In *Educational Multimedia, Hypermedia & Telecommunications - EDMEDIA05, Montréal*, pages 3439–3446. Association for the Advancement of Computing in Education, 27 Juin - 2 Juillet 2005.

[For00] Distributed Management Task Force. Common information model (cim) core model. Technical report, DSP0111, August 2000, http://www.dmtf.org/standards/documents/CIM/DSP0111.pdf.

[For03a] Distributed Management Task Force. Cim netwrok model white paper. Technical report, DSP0152, December 2003, http://www.dmtf.org/standards/published_documents/DSP0152.pdf.

[For03b] Distributed Management Task Force. Cim application model white paper. Technical report, DSP0140, June 2003, http://www.dmtf.org/standards/documents/CIM/DSP0140.pdf.

[For03c] Distributed Management Task Force. Cim policy model white paper. Technical report, DSP0108, June 2003, http://www.dmtf.org/standards/documents/CIM/DSP0108.pdf.

[For03d] Distributed Management Task Force. Cim system model white paper. Technical report, DSP0150, June 2003, http://www.dmtf.org/standards/documents/CIM/DSP0150.pdf.

[For03e] Distributed Management Task Force. User and security model white paper. Technical report, DSP0139, June 2003, http://www.dmtf.org/standards/documents/CIM/DSP0139.pdf.

[FR04] K. Fansler and R. Riegle. A model of online instructional design analytics. In *20th Annual Conference on Distance Teaching and Learning, Madison*. The Board of Regents of the University of Wisconsin System, 4-6 Août 2004, http://uwex.edu/disted/conference/Resource_library/proceedings/04_1069.pdf.

[Fri04] N. Friesen. International review of research in open and distance learning. Technical report, Athabasca University, 2004, http://www.irrodl.org/content/v5.3/technote5.html.

[FT98] F. Farance and J. Tonkel. Learning technology systems architecture (ltsa) specification. Technical report, Farance Inc, Edutool Division, May 1998, http://ltsc.ieee.org/wg1/files/ltsa-400.html.

[ftAoSIS04] Organization for the Advancement of Structured Information Standards. Uddi executive overview : Enabling service-oriented architecture. Technical report, Organization for the Advancement of Structured Information Standards, October 2004, http://uddi.org/pubs/uddi-exec-wp.pdf.

[FWD97a] E. Forte, M.H.K Wentland, and E. Duval. The ariadne project (part 1) : Knowledge pools for computer-based and telematics supported classical, open and distance education. *European Journal of Engineering Education*, 22(1) :61–74, 1997.

[FWD97b] E. Forte, M.H.K Wentland, and E. Duval. The ariadne project (part 2) : Knowledge pools for computer-based and telematics supported classical, open and distance education. *European Journal of Engineering Education*, 22(2) :153–166, 1997.

[FWFD96] E. Forte, W. Wentland-Forte, and E. Duval. Ariadne : a framework for technology-based open and distance education. In *7th EAEEIE International Conference on Telematics for Future Education and Training, Oulu*, pages 69–72, 12-14 Juin 1996.

[GD04] P. Gounon and X. Dubourg. A descriptive model to organise tutoring for learning environments. In *4th IEEE International Conference on Advanced Learning Technologies - ICALT'04, Joensuu*, pages 630–632. IEEE Computer Society, 30 Août - 1er Septembre 2004.

[GDL04] P. Gounon, X. Dubourg, and P. Leroux. Un modèle d'organisation du tutorat pour la conception de dispositifs informatiques d'accompagnement des apprenants. In *Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie - TICE 2004, Compiègne*, pages 369–376. Université de Technologie de Compiègne, 20-22 Octobre 2004.

[Ghi04] E. Ghiglione. *LORS, A Standards and Specifications Compliant Learning Object Repository.* PhD thesis, The University of Sydney, 2004, http://weg.ee.usyd.edu.au/people/ernieg/thesis/final/#_Toc81648324.

[GHM+03] M. Gudgin, M. Hadley, N. Mendelsohn, J.J. Moreau, and H. Frystyk Nielsen. Soap version 1.2 part 1 : Messaging framework. Technical report, World Wide Web Consortium, June 2003, http://www.w3.org/TR/soap12-part1/.

[Gla02] M. Gladwell. *The tipping point : How Little Things can Make a Big Difference.* Paperback, 2002.

[GLD05] P. Gounon, P. Leroux, and X. Dubourg. Proposition d'une extension du langage de modélisation pédagogique ims-learning design. In *Environnement Informatique pour l'Apprentissage Humain - EIAH 05, Montpellier*, pages 261–272. Institut National de Recherche Pédagogique, 25-27 Mai 2005.

[Gro04] Object Management Group. Common object request broker architecture : Core specification. Editorial revision : Corba 3, Object Management Group, 2004, http ://www.omg.org/docs/formal/04-03-01.pdf.

[GTZ05] E. Giacomini, P. Trigano, and C. Zamfirescu. An ims ld graphic editor using the graphs representation for modifying the course structures. *Journal of Interactive Media in Education*, 15(Special issue Advances in Learning Design), 2005, http://www-jime.open.ac.uk/2005/15/giacomini-2005-15.pdf.

[HEW04] M. Hatala, G. Richards T. Eap, and J. Willms. The interoperability of learning object repositories and services : Standards, implementations and lessons learned. In *13th International World Wide Web Conference, New York*, pages 19–27. ACM Press, 17-22 Mai 2004, http://www2004.org/proceedings/docs/2p19.pdf.

[HF99] F. Van Harmelen and D. Fensel. Practical knowledge representation for the web. In *Workshop on Intelligent Information Workshop on Intelligent Informa-*

tion Integration - IJCAI'99, Stockholm. M. Kaufmann, 31 Juillet - 6 Août 1999, http://www.cs.vu.nl/ frankh/postscript/IJCAI99-III.html.

[HK97] M. Horstmann and M. Kirtland. Dcom architecture. Technical report, Microsoft, July 1997, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn_dcomarch.asp.

[HSD98] T. Howes, M. Smith, and F. Dawson. A mime content-type for directory information. Technical Report RFC2425, Internet Engineering Task Force, 1998, http://www.ietf.org/rfc/rfc2425.txt.

[HSH+03] M. J. Halm, R. Shuey, M. Hatala, G. Richards, and M. J. Dooris. Lionshare : Connecting and extending peer-to-peer networks. Technical report, Andrew W. Mellon Foundation, September 2003, http://lionshare.its.psu.edu/main/info/docspresentation/lionshare_mellon_pdf.

[Inc03] Novasys Inc. La description normalisée des ressources : vers un patrimoine éducatif – normetic, version 1.0. Technical report, Conférence des Recteurs Et des Principaux des Universités du Québec, October 2003, http://213.186.34.36/espaces/formateur/eforma/normetic.pdf.

[Ini04] Dublin Core Metadata Initiative. Dublin core metadata element set, version 1.1 : Reference description. Technical report, Dublin Core Metadata Initiative, 2004, http ://dublincore.org/documents/2004/12/20/dces/.

[JJBW99] C. Jenkins, M. Jackson, P. Burden, and J. Wallis. Automatic rdf metadata generation for resource discovery. *Computer Networks*, 31(15) :11–16, 1999, http://www8.org/w8-papers/2c-search-discover/automatic/automatic.html.

[JMBS01] F. Jocteur-Monrozier, D. Benech, and M. Sibilla. Les technologies orientées objet au coeur des systèmes de gestion : modèlisation, organisation et répartition. *Electronic Journal on Networks and Distributed Processing*, 11 :211–227, Mars 2001.

[KM04] E.J.R. Koper and J.M. Manderveld. Educational modelling language : modelling reusable, interoperable, rich and personalised units of learning. *British Journal of Educational Technology*, 35(5) :537–552, September 2004.

[Kop00] E.J.R. Koper. From change to renewal : Educational technology foundations of electronic environments. Technical report, Open University of the Netherlands, 2000, http://eml.ou.nl/introduction/docs/koper-inaugural-address.pdf.

[Kop01] E.J.R. Koper. Modelling units of study from a pedagogical perspective : the pedagogical metamodel behind eml. Technical report, Open University of the Netherlands, June 2001, http://eml.ou.nl/introduction/docs/ped-metamodel.pdf.

[KT05] R. Koper and C. Tattersall. *Learning Design : A Handbook on Modelling and Delivering Networked Education and Training.* Springer Verlag, 2005.

[Lea04a] Advanced Distributed Learning. Modèle de référence scorm (sharable content object reference model information) 2004 - 2ème édition. Technical report, Advanced Distributed Learning, July 2004, http://www.adlnet.gov/downloads/files/207.cfm.

[Lea04b] Advanced Distributed Learning. Sharable content object reference model (scorm) 2004 2nd edition document suite. Technical report, Advanced Distributed Learning, July 2004, http://www.adlnet.org/downloads/files/67.cfm.

[Lea05a] Advanced Distributed Learning. Scorm 2004 conformance test suite version 1.3.3. Technical report, Advanced Distributed Learning, May 2005, http://www.adlnet.org/downloads/199.cfm.

[Lea05b] Advanced Distributed Learning. Sharable content object reference model - sample run-time environment version 1.3.3. Technical report, Advanced Distributed Learning, May 2005, http://www.adlnet.org/downloads/files/194.cfm.

[Lej04] A. Lejeune. Ims learning design : Etude d'un langage de modélisation pédagogique. *Distance et Savoirs*, 2(4) :409–450, 2004.

[LG04] J. Liu and J. Greer. Individualized selection of learning object. In *Workshop on Applications of Semantic Web Technologies for E-Learning - SWEL@ITS'04, Maceió*, 30 Août - 3 Septembre 2004, http://www.win.tue.nl/SW-EL/2004/ITS-SWEL-Camera-ready/SWEL04-ITS-PDF/

[LP05] A.J. Lopez and R. Pérez. Learning statistics in a shared virtual campus. summarizing a five-year experience. *International Journal of Instructional Technology And Distance Learning*, 2(3) :29–39, Mars 2005, http://www.itdl.org/Journal/Mar_05/Mar_05.pdf.

[Mas05] D. Massart. Accessing learning contents using a "simple query interface" adapter. In *Prolearn-iClass Thematic Workshop, Leuven*, 3-4 Mars 2005, http://www.cs.kuleuven.ac.be/ hmdb/ProlearnIClass/papers/Massart.pdf.

[McC03] R. McCormick. Keeping the pedagogy out of learning objects. In *Designing Virtual Learning Material Symposium, Padoue*, 26-30 Août 2003, http://www.eun.org/eun.org2/eun/Include_to_content/celebrate/file/KeepingPedagogyOutOfLOs3v2.doc.

[MFC+04] C. Martel, C. Ferraris, B. Caron, T. Carron, G. Chabert, C. Courtin, J.-C. Marty, and L. Vignollet. A model for cscl allowing tailorability : Implementation in the electronic schoolbag groupware. In *10th International Workshop on Groupware - CRIWG'04, San Carlos*, pages 322–338. LNCS 3198, 5-9 Septembre 2004.

[MM03] J. Mukerji and J. Miller. Mda guide version 1.0.1. Technical report, Object Management Group, 2003, http ://www.omg.org/docs/omg/03-06-01.pdf.

[Moo01] M.G. Moore. Standards and learning objects. *American Journal of Distance Education*, 15(3) :1–4, 2001.

[MP03] F. Michau and S.S. Poix. Proposition de carte organisationnelle - décrire le rôle d'un environnement informatique au sein d'un dispositif pédagogique. In *Environnement Informatique pour l'Apprentissage Humain - EIAH 03, Strasbourg*, pages 331–342. Institut National de Recherche Pédagogique & ATIEF, 15-17 Avril 2003, http://archiveseiah.univ-lemans.fr/EIAH2003/.

[NLL00] H. Nielsen, P. Leach, and S. Lawrence. An http extension framework. Technical report, Internet Engineering Task Force, February 2000, http://www.ietf.org/rfc/rfc2774.txt.

[NTD04] J. Najjar, S. Ternier, and E. Duval. User behavior in learning object repositories : An empirical analysis. In *Educational Multimedia, Hypermedia & Telecommunications - EDMEDIA04, Lugano*, pages 4373–4379. Association for the Advancement of Computing in Education, 16-21 Juin 2004.

[NWQ+02] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch. Edutella : a p2p networking infrastructure based on rdf. In *11th International World Wide Web Conference - WWW2002, Hawaï*. ACM Press, 7-11 Mai 2002, http://edutella.jxta.org/reports/edutella-whitepaper.pdf.

[oEEE02] Institute of Electrical and Inc Electronics Engineers. Draft standard for learning object metadata. Technical report, IEEE Standards Department, July 2002, http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf.

[OP01] J. Campos Oriola and P. Pannetier. Club interopérabilité. Technical report, Agence pour les Technologies de l'Information et de la Communication dans l'Administration, 2001, http://www.adae.gouv.fr/upload/documents/club_interop_20011025.PDF.

[Paq00] G. Paquette. Construction de portails de télé-apprentissage explor@ - une diversité de modèles pédagogiques. *Revue Sciences et Techniques Educatives*, 7(1) :207–226, 2000.

[Pau02] M.F. Paulsen. Online education systems : Discussion and definition of terms. Technical report, NKI Distance Education, 2002, http://www.nettskolen.com/forskning/Definition

[Pau03] M.F. Paulsen. Experiences with learning management systems in 113 european institutions. *Educational Technology and Society*, 6(4) :134–148, 2003, http://ifets.massey.ac.nz/periodical/6_4/13.pdf.

[PCMG02] D. Porter, J. Curry, B. Muirhead, and N. Galan. A report on learning object repositories. Technical report, CANARIE Inc., Mars 2002, http://www.canarie.ca/funding/elearning/lor.pdf.

[Per03] J.P. Pernin. Objets pédagogiques : unités d'apprentissage, activités ou ressources ? *Revue Sciences et Techniques Educatives*, Numéro spécial "Ressources Numériques, XML et Education" :179–210, 2003.

[PlTL05] G. Paquette, I. De la Teja, and M. Léonard. An instructional engineering method and tool for the design of units of learning. *Learning Design : A handbook on Modelling and Delivering Networked Education and Training*, pages 161–183, 2005.

[PMlT+05] G. Paquette, O. Marino, I. De la Teja, M. Léonard, and K. Lundgren Cayrol. Delivery of learning design : the explor system's case. *Learning Design : A handbook on Modelling and Delivering Networked Education and Training*, pages 311–325, 2005.

[SdSB+04] M. Sibilla, A. Barros de Sales, J. Broisin, P. Vidal, and F. Jocteur-Monrozier. Behaviour modelling : a contribution to cim. In *DMTF Academic Alliance Paper Competition, Winner*, 2004, http://www.dmtf.org/education/academicalliance/sibilla_2004.pdf.

[Sim02] C. Simard. La normalisation de la formation en ligne : Enjeux, tendances et perspectives. Technical report, Bureau Amérique du Nord, Agence Universitaire de la Francophonie, February 2002.

[SJM04] M. Sibilla and F. Jocteur-Monrozier. L'expérience de la supervision : utilisation des modèles au coeur des systèmes. *Ingénierie des modèles logiciels et systèmes*, pages 151–175, 2004.

[SMA+05] B. Simon, D. Massart, F. Van Asscheand, S. Ternier, and E. Duval. Simple query interface specification. Technical report, Prolearn, April 2005, http://nm.wu-wien.ac.at/e-learning/interoperability/SQI_V1.0beta_2005_04_13.pdf.

[SSS+99] C. Seeberg, A. El Saddik, A. Steinacker, K. Reichenberger, S. Fischer, and R. Steinmetz. From the user's needs to adaptive documents. In *4th World Conference on Integrated Design & Process Technology - IDPT'99, Kusadasi*, 27 Juin - 2 Juillet 1999, http://www.kom.e-technik.tu-darmstadt.de/publications/abstracts/SSR+99-3.html.

[Ste05] R. Steinmetz. Learning objects, learning object repositories and beyong. In *Intelligent, Interactive, Learning Object Repositories 2005 - LORNET Research Network, Vancouver*, 15-18 Novembre 2005.

[Tch02] P. Tchounikine. Pour une ingénierie des environnements informatiques pour l'apprentissage humain. *Revue I3 information - interaction - intelligence*, 2(1) :59–95, 2002.

[VAB03] P. Vidal, A. Alibert, and P. Baqué. Towards e-learning platforms interoperability : an opening solution. In *4th International Conference on Information Technology Based Higher Education and Training - ITHET'03, Marakech*, pages 193–199. IEEE Computer Society, 7-9 Juillet 2003.

[VB05] P. Vidal and J. Broisin. Vers la fédération des ressources pédagogiques : une architecture fédérée de systèmes de gestion de contenus d'apprentissage. In *Environnement Informatique pour l'Apprentissage Humain - EIAH 05, Montpellier*, pages 117–128. Institut National de Recherche Pédagogique, 25-27 Mai 2005.

[VBD04] P. Vidal, J. Broisin, and E. Duval. Learning objects : the ariadne experience. In *18th IFIP World Computer Congress - WCC 2004, Toulouse*, pages 551–556. Kluwer Academic Publishers, 22-27 Août 2004.

[VRD+03] P. Vandepitte, L. V. Rentergem, Erik Duval, S. Ternier, and F. Neven. Bridging an lcms and an lms : a blackboard building block for the ariadne knowledge pool system. In *Educational Multimedia, Hypermedia & Telecommunications - EDMEDIA03, Honolulu*, pages 423–424. Association for the Advancement of Computing in Education, 23-28 Juin 2003, http://www.cs.kuleuven.ac.be/ hmdb/publications/files/pdfversion/41254.pdf.

[War97] C. Warwick. The dublin core : simplicity or complexity ? In *National Library of Australia Staff Presentation Papers, Sydney*, 30 Juillet 1997, http://www.nla.gov.au/nla/staffpaper/cathro2.html.

[WHK97] M. Wahl, T. Howes, and S. Kille. Lightweight directory access protocol (v3). Technical report, Internet Engineering Task Force, December 1997, http://ietfreport.isoc.org/rfc/rfc2251.txt.

[WKG04] F. Weitl, R. Kammerl, and M. Gostl. Context aware reuse of learning resources. In *Educational Multimedia, Hypermedia & Telecommunications - EDMEDIA04, Lugano*, pages 2119–2126. Association for the Advancement of Computing in Education, 21-26 Juin 2004.

Glossaire

ADL	*Advanced Distributed Learning* : Une initiative du ministère américain de la défense et du Bureau des Sciences et des Technologies de la Maison Blanche à l'origine du standard SCORM.	41
API	*Application Program Interface* : Une interface de programmation.	9
ARIADNE	*Alliance of Remote Instructional Authoring and Distribution Networks for Europe* : Une association européenne ouverte au monde pour partager et réutiliser la connaissance.	9
AUF	*Agence Universitaire de la Francophonie* : Un réseau mondial de 577 établissements d'enseignement supérieur et de recherche.	23
CGI	*Common Gateway Interface* : Une interface d'exécution des programmes sur un serveur.	52
CIM	*Common Information Model* : Proposition du DMTF qui comporte un formalisme et un modèle de l'information pour les systèmes informatiques.	86
DC	*Dublin Core* : Un ensemble de métadonnées visant à faciliter la découverte de ressources pédagogiques.	171
DMTF	*Distributed Management Task Force* : Un consortium d'industriels dont les travaux visent le développement de standards de gestion et la promotion de l'interopérabilité entre environnements d'entreprise ou Internet.	89
DNS	*Domain Name Server* : Un service de résolution des noms de machines qui les traduit en adresses IP.	34
EdNA	*Education Network Australia* : Une organisation qui a pour objectif de supporter et de promouvoir les avantages de l'Internet pour l'Aprrentissage et l'Education en Australie.	29

EIAH	*Environnement Informatique pour l'Apprentissage Humain* : Environnement informatique conçu dans le but de favoriser l'apprentissage humain, c'est-à-dire la construction de connaissances chez un apprenant.	5
FOAD	*Formation Ouverte et A Distance* : Dispositif de formation utilisant l'Internet ou des Intranets comme vecteurs de connaissances et de diffusion de formation.	4
HTML	*HyperText Mark-up Language* : Langage de description des pages *web* composé d'une suite de signe ASCII dans laquelle sont incluses les commandes spéciales concernant le formatage des pages, la police de caractères et les multimédia.	81
HTTP	*HyperText Transfer Protocol* : Protocole utilisé pour le transport de données sur le réseau.	52
IEEE	*Institute of Electrical and Electronic Engineers* : Organisme américain de normalisation.	23
IETF	*Internet Engineering Task Force* : Organisme composé de chercheurs et d'industriels qui travaillent sur l'évolution de l'Internet à travers la proposition de standards.	34
IMS/GLC	*Instructional Management Systems Global Learning Consortium* : Ce consortium développe et promeut l'adoption de spécifications techniques ouvertes visant l'interopérabilité des technologies d'apprentissage.	30
IP	*Internet Protocol* : Protocole standard de routage de niveau réseau. En outre, il définit un format d'adresse logique des éléments.	52
IRIT	*Institut de Recherche en Informatique de Toulouse* : Le laboratoire qui m'a accueilli durant ma thèse. C'est une unité mixte de recherche (UMR 5505) commune à quatre établissements dont l'organisation autour de 7 thèmes couvre l'ensemble des problématiques de recherche de l'informatique actuelle.	3
ISO	*International Organisation for Standardization* : Organisation internationale de standardisation regroupant les organismes similaires de 89 nations. L'ISO se charge des standards qui régissent l'Internet actuellement.	26

KPS	*Knowledge Pool System* : C'est le vivier de connaissance de la fondation ARIADNE.	28
LDAP	*Lightweight Directory Access Protocol* : Structure d'annuaire standard pour les réseaux TCP/IP.	98
LMS	*Learning Management System* : Un système qui organise et fournit un accès à des services d'apprentissage en ligne à des étudiants, enseignants, et administrateurs.	36
LOF	*Learning Object reFeratory* : Système de stockage de ressources pédagogiques ne renfermant que les métadonnées, les documents consistant en une URL représentée par un champ du modèle de métadonnées.	28
LOM	*Learning Object Metadata* : Standard de l'IEEE qui spécifie la syntaxe et la sémantique des métadonnées pédagogiques et définit les attributs nécessaires à une description complète des objets pédagogiques.	23
LOR	*Learning Object Repository* : Système de stockage de documents numériques et des métadonées qui leur sont associées.	28
LRC	*Learning Resource Catalog* : Le vivier de connaissance de l'initiative Learnet dont l'objectif est de promouvoir et de faciliter le partage de ressources pédagogiques de haute qualité parmi les huit institutions regroupées dans l'University Grants Committee de Hongkong.	116
MERLOT	*Multimedia and Educational Resource for Learning and Online Teaching* : Un vivier de connaissance américain dont le but est de donner accès à des documents pédagogiques aux étudiants de l'Enseignement Supérieur.	29
MOF	*Managed Object Format* : Une spécification du DMTF pour décrire les définitions d'objet CIM de façon textuelle.	90
OM	*Object Manager* : Une entité de gestion qui a charge d'un ou plusieurs domaines de gestion.	100
OP	*Object Provider* : Une entité de gestion responsable de l'intégration d'un environnement à gérer comme par exemple l'environnement Unix, un environnement de base de données Oracle^{TM} ou tout autre environnement comme LDAP.	100

PHP	*PHP : Hypertext Preprocessor* : Un langage de programmation interactif orienté serveur et permettant la création de pages *web* au contenu dynamique.	52
RDF	*Resource Description Framework* : Un langage qui permet de définir les informations qu'un utilisateur souhaite divulguer à un serveur et leur mode d'utilisation.	26
SCORM	*Sharable Content Object Reference Model* : Un modèle d'agrégation de contenus et d'environnement d'exécution pour les objets d'apprentissage sur le *web*.	41
SGBD	*Systèmes de Gestion de Bases de Données* : Logiciel permettant de stocker les données, de les mettre à jour et de les consulter.	27
SIERA	*Service IntEgration and netwoRk Administration* : L'équipe de recherche au sein de laquelle j'ai effectué ma thèse. Sa thématique concerne le contrôle et la maîtrise des infrastructures et services de communication de dernières générations, mais aussi des systèmes et applications complexes dynamiquement agrégés, distribués et trans-organisationnels.	3
SILO	*Search & Index Learning Objects* : L'outil graphique de la fondation ARIADNE qui offre un service de recherche et d'indexation d'objets pédagogiques au sein du KPS.	131
SMTP	*Simple Mail Transfer Protocol* : Un protocole de la famille TCP/IP utilisé pour le transfert de courrier électronique.	34
SOAP	*Simple Object Access Protocol* : Une recommandation du W3C permettant l'appel de méthodes sur un objet distribué sur le réseau Internet. Les paramètres nécessaires à l'appel sont transférés via le protocole HTTP qui contient un message XML.	34
SQI	*Simple Query Interface* : Une initiative de standardisation ayant pour objectif de spécifier un ensemble d'opérations simples qui permettent d'interroger des viviers de connaissance hétérogènes.	31
UDDI	*Universal Description Discovery and Integration* : Une spécification en langage XML d'un catalogue des services *web* offerts par une organisation qui doit permettre à un logiciel de reconnaître automatiquement les services et de s'interfacer avec eux.	34

UML	*Unified Modeling Language* : Un concept de modélisation orienté objet. Il simplifie le processus complexe de conception d'un système informatique en définissant 9 points de vue (ou diagrammes) de modélisation.	33
URI	*Uniform Resource Identifier* : Indique précisément la position de chaque ressource d'Internet, les deux types d'URI sont les URL et les URN.	179
URL	*Uniform Resource Locators* : Adresse Internet exploitée par les navigateurs. C'est l'adressage standard de n'importe quel document, sur n'importe quel ordinateur en local ou sur Internet.	28
W3C	*World Wide Web Consortium* : Un consortium qui vise à proposer des applications, des standards et des recommandations pour les applications *web*.	34
WBEM	*Web Based Enterprise Management* : Une approche de gestion de réseaux et services proposée par le DMTF.	100
WSDL	*Web Service Description Language* : Un langage de description des services *web* qui définit une grammaire et donne un sens à chaque élément de cette grammaire.	34
XML	*eXtensible Markup Language* : Un méta langage standardisé par le W3C qui permet de générer des langages à base de balises.	26

Annexes

Annexe A
Standardisation des métadonnées

Dans cette annexe, nous présentons les principales initiatives de standardisation des métadonnées qui ont vu le jour depuis 1997 et dont la synthèse figure en section 3.2.4.

A.1 Dublin Core (DC)

Le nom *Dublin Core* (DC) est une abréviation de *Dublin Core Metadata Element Set.* Cet ensemble d'éléments a été établi par un Consensus International coordonné par le OCLC[1], et réunissant des professionnels tels que des libraires, chercheurs dans le domaine des librairies digitales, ainsi que spécialistes du text-markup. L'objectif de ce consortium était d'adresser le problème de la découverte des grandes variétés de ressources à travers le réseau.

Quinze ensembles de descripteurs forment aujourd'hui le DC pour satisfaire les buts et spécifications suivants [Ini04] :

1. **Simplicité** de création et de maintenance, même pour les non-spécialistes.
2. **Sémantique commune** à travers les différents champs du "savoir".
3. **Conformité** aux standards existants et émergents.
4. **Etendue et application internationales** par mise en correspondance du DC avec divers langages.
5. **Extensibilité**.
6. **Interopérabilité** entre les différents systèmes d'indexation et de collection d'objets pédagogiques.

Chacun des 15 éléments est optionnel et peut être répété/associé à un nombre limité de qualifiants afin d'en affiner la description sémantique. Le tableau A.1 énumère ces éléments ainsi que leur description associée. Les éléments du DC adressent les aspects relatifs au contenu de

[1]Online Computer Library Center - http://www.oclc.org

la ressource, aux droits de la propriété intellectuelle, et enfin à l'instanciation particulière de la ressource :

(i) **Contenu** : Title, Coverage, Description, Type, Relation, Source, Subject.

(ii) **Propriété intellectuelle** : Creator, Publisher, Contributor, Rights.

(iii) **Instanciation** : Date, Format, Identifier, Language.

Elément	*Description*
Title	Nom donné à la ressource.
Creator	Entité responsable du contenu de la ressource.
Subject	Domaine abordé dans le contenu de la ressource.
Description	Aperçu du contenu de la ressource.
Publisher	Entité responsable de la mise à disposition de la ressource.
Contributor	Entité qui a contribué au contenu de la ressource.
Date	Date associée à un événement du cycle de vie (par exemple creation).
Type	Nature ou genre du contenu de la ressource.
Format	Format (physique ou digital) de la ressource.
Identifier	Référence non ambigüe à la ressource dans un contexte donné (par exemple URL).
Source	Référence à partir de laquelle la ressource est dérivée.
Language	Langage du contenu intellectuel de la ressource.
Relation	Référence à une ressource relative à la présente ressource.
Coverage	Ampleur ou étendue du contenu de la ressource.
Rights	Information concernant les droits associés à la ressource.

TAB. A.1 – Le Dublin Core Metadata Set

Un des principaux avantages du Dublin Core réside dans sa flexibilité. A l'origine, la conception du DC était suffisamment simple pour que tout public puisse l'utiliser sans formation particulière. Ce caractère simpliste contraste avec les catalogues traditionnels et leurs règles d'application complexes associées à des formats de base de données élaborés.

Cependant le Dublin Core n'est pas suffisant pour satisfaire une description détaillée de documents pédagogiques qui requiert des attributs bien définis dans le but de décrire pleinement et de façon adéquate les objets pédagogiques [Cla97] [War97]. Alors d'autres groupes de travail tels que ARIADNE et IMS/GLC ont élaboré leur propre schéma de métadonnées afin d'enrichir celui proposé par le Dublin Core ; nous les présentons dans les deux sections suivantes.

A.2 ARIADNE Metadata Set

Le système ARIADNE est le résultat d'un projet de recherche et développement portant sur la formation flexible et à distance initié en 1996. Ses objectifs étaient le développement

d'outils et méthodologies d'apprentissage favorisant le partage et la réutilisation de documents pédagogiques hypermédias. Depuis 2002, la fondation ARIADNE propose l'exploitation de son système à ses membres et ses outils Open Source sont disponibles en libre téléchargement sur le site de Source Forge[2].

Dès son commencement, le projet ARIADNE a joué un rôle important dans l'élaboration des standards de métadonnées puisqu'il fût l'une des premières initiatives visant à décrire spécifiquement des ressources pédagogiques. Ainsi, les initiateurs du projet ont défini un ensemble de métadonnées nommé ARIADNE Metadata Set [DVV+00] composé de 39 descripteurs et divisé en 4 catégories :

- **Général** : cette catégorie inclut des éléments tels que le titre, la langue, les droits d'usage ou le(s) auteur(s) de l'objet pédagogique.
- **Sémantique** : cette catégorie traite des thèmes abordés par la ressource pédagogique. Des éléments comme la discipline principale, la discipline secondaire ou les concepts principaux sont mentionnés ici.
- **Pédagogique** : cette catégorie de descripteurs indique le caractère actif ou expositif du document, le temps nécessaire à la bonne compréhension du document (même si ce temps est bien évidemment dépendant de l'individu) ou le contexte didactique dans lequel cette ressource peut être utilisée.
- **Technique** : les métadonnées précisent ici la taille, le nom, ou encore le type MIME du fichier associé à la ressource.

Le modèle de métadonnées proposé par ARIADNE a suscité un fort intérêt dans la communauté EIAH puisque les spécifications proposées jusqu'ici se voulaient plus généralistes. De ce fait, d'autres initiatives sont apparues et se sont basées sur les travaux proposés dans ARIADNE.

A.3 Instructional Management Systems (IMS)

Le IMS Global Learning Consortium « développe et promeut l'adoption de spécifications techniques ouvertes visant l'interopérabilité des technologies d'apprentissage ». IMS est une organisation à but non lucratif incluant plus de 40 membres et associés dont la majorité est issue d'entités commerciales américaines et anglaises, même si certains membres représentent des universités et agences gouvernementales fédérales. Cette connotation anglosaxone a donc influencé sur la langue utilisée au sein d'IMS, ce qui a parfois conduit à l'altération des aspects

[2] http://cvs.sourceforge.net/viewcvs.py/ariadnekps/#dirlist

multilingue et multiculturel. Cependant, les nationalités et secteurs représentés par l'ensemble des membres ont évolué ; le Canada est désormais représenté au sein d'IMS à travers le partenariat avec Industry Canada[3], et au moins deux groupes de travail d'IMS sont co-présidés par des représentants canadiens.

IMS a réalisé un ensemble de spécifications pour l'enseignement, et un grand nombre d'institutions privées et publiques les ont adoptées. Parmi ces spécifications, figurent :

- **IMS Content Packaging**[4] (IMS-CP) [Con05] a pour objectif de standardiser la façon dont les objets pédagogiques et les métadonnées sont morcelés afin d'être facilement retrouvés et exploités par différentes plates-formes de gestion d'apprentissage. Nous décrivons en détail cette spécification dans la suite de cette section.
- **IMS Digital Repositories Interoperability**[5] (IMS-DRI) [Con03a] recommande des méthodes facilitant la communication et l'interopérabilité des viviers de connaissance afin de simplifier leur utilisation. IMS-DRI a été exposé dans la section 3.3.3.
- **IMS Learning Design**[6] (IMS-LD) [Con03b] est la spécification la plus récente et présente un grand intérêt pour les communautés EIAH. Cette spécification ambitieuse et complexe a pour but d'offrir l'expression standardisée d'unités d'apprentissage (incluant activités, environnements, rôles, ressources et services) pour que celles-ci soient facilement partagées, réutilisées, ou exécutées par les systèmes de e-formation.

Le consortium IMS a défini d'autres spécifications telles que IMS Question and Test Interoperability[7] [Con06] qui spécifie une structure facilitant la représentation des questions et des tests (ou évaluation), IMS Reusable Definition of Competency or Educational Objective[8] [Con02] pour exprimer de façon standardisée des compétences et des résultats, ou encore IMS Vocabulary and Definition Exchange[9] [Con04] afin de permettre la description et la création de vocabulaires contrôlables et facilement interchangeables. Ces spécifications relatives aux processus d'apprentissage ne nous intéressent pas directement puisqu'elles ne concernent pas objectivement les ressources pédagogiques. Le lecteur est invité à consulter les sites Internet mentionnés pour plus de détails à propos de leurs rôles et objectifs.

Dans cette section nous nous intéressons à la spécification IMS Content Packaging[10]. Celle-ci est constituée de deux composantes incluant le fichier particulier *manifest* au format

[3] http://www.ic.gc.ca/
[4] http://imsglobal.org/content/packaging/index.html
[5] http://www.imsglobal.org/digitalrepositories/index.html
[6] http://www.imsglobal.org/learningdesign/index.html
[7] http://www.imsglobal.org/question/
[8] http://www.imsglobal.org/competencies/
[9] http://www.imsglobal.org/vdex/
[10] http://www.imsglobal.org/content/packaging/

XML et nommé *imsmanifest.xml*, ainsi que les fichiers physiques. Le fichier *manifest* est essentiel dans la mesure où il détient les clefs nécessaires à l'agrégation et à la désagrégation d'une ressource conforme à la spécification. Ce fichier est composé de quatre sections illustrées par la figure A.1 :

- **Métadonnées** : décrit le package en entier selon les spécifications de IMS Learning Resource Meta-data.
- **Organisation** : fournit la structure de la ressource.
- **Ressources** : décrit à la fois les fichiers physiques mais aussi les ressources externes.
- **Sous-manifest** : le *manifest* de plus haut niveau peut contenir un autre document *manifest*.

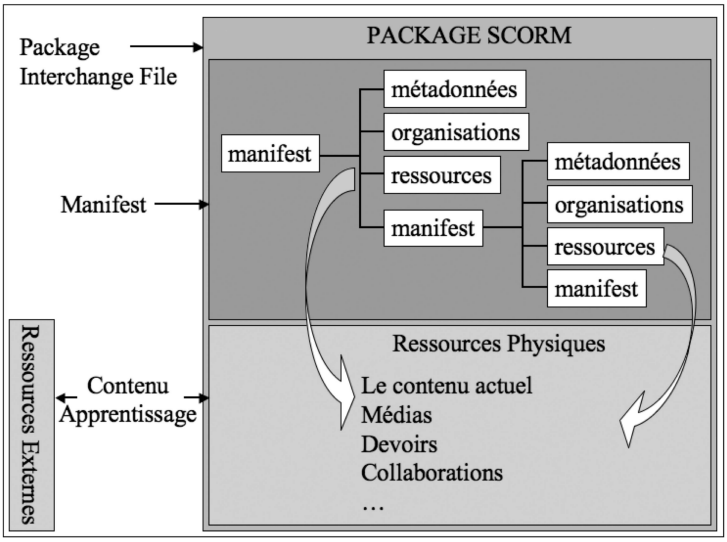


Fig. A.1 – Diagramme conceptuel de l'IMS Content Packaging [Con05]

La spécification IMS Learning Resource Meta-data[11] vise à décrire les métadonnées des ressources éducatives physiques et externes contenues dans un package conforme à IMS-CP ; la première version de cette spécification est apparue en 1999 à partir des travaux menés par la communauté ARIADNE et n'a cessé d'évoluer jusqu'en Mars 2004, date à laquelle le modèle d'information de cette spécification a été remplacé par le standard IEEE Learning Object Metadata que nous exposons maintenant.

[11] http://www.imsglobal.org/metadata/

A.4 Le standard Learning Object Metadata (LOM)

L'association IEEE a créé en son sein le groupe LTSC[12] à partir du domaine d'action "Computer Society". Actuellement le standard LOM [oEEE02] (Learning Object Metadata) spécifie la syntaxe et la sémantique des métadonnées pédagogiques et définit les attributs nécessaires à une description complète des objets pédagogiques. Le sens du terme "objet pédagogique" du point de vue IEEE est celui donné dans la section 3.2.2. Les objectifs du LOM sont les suivants [DHSW02] :

- Permettre aux apprenants et instructeurs de **rechercher, évaluer, acquérir et utiliser** des objets pédagogiques.
- **Permettre le partage et l'échange** d'objets pédagogiques entre toute technologie supportant des systèmes d'apprentissage.
- **Permettre le développement** d'ojets pédagogiques en unités qui peuvent être combinées et décomposées de façon pertinente.
- Permettre aux agents informatiques de **composer automatiquement et dynamiquement** des leçons personnalisées pour un apprenant particulier.

Le schéma de structure de base du LOM définit neuf catégories, chacune d'entre elles regroupant des éléments sémantiquement liés aux données [oEEE02] :

1. **General** : regroupe les attributs indépendants du contexte et les descripteurs sémantiques de la ressource.
2. **Lifecycle** : regroupe les attributs liés au cycle de vie de la ressource.
3. **Meta-metadata** : regroupe les attributs de la description elle-même (et non pas ceux de la ressource elle-même).
4. **Technical** : regroupe les attributs techniques de la ressource.
5. **Educational** : regroupe les attributs éducationnels et pédagogiques de la ressource.
6. **Rights** : regroupe les attributs traitant des conditions d'utilisation de la ressource.
7. **Relation** : regroupe les attributs de la ressource qui l'associent à d'autres ressources.
8. **Annotation** : permet d'ajouter des commentaires liés à l'usage pédagogique de la ressource.
9. **Classification** : regroupe des caractéristiques de la ressource décrites au moyen d'entrées dans des classifications telles que la classification décimale Dewey[13].

[12] Learning Training System Commitee - http://ltsc.ieee.org/wgl2/index.html
[13] http://www.oclc.org/dewey/

La figure A.2 illustre la hiérarchie des éléments de données définis par le Learning Object Metadata.

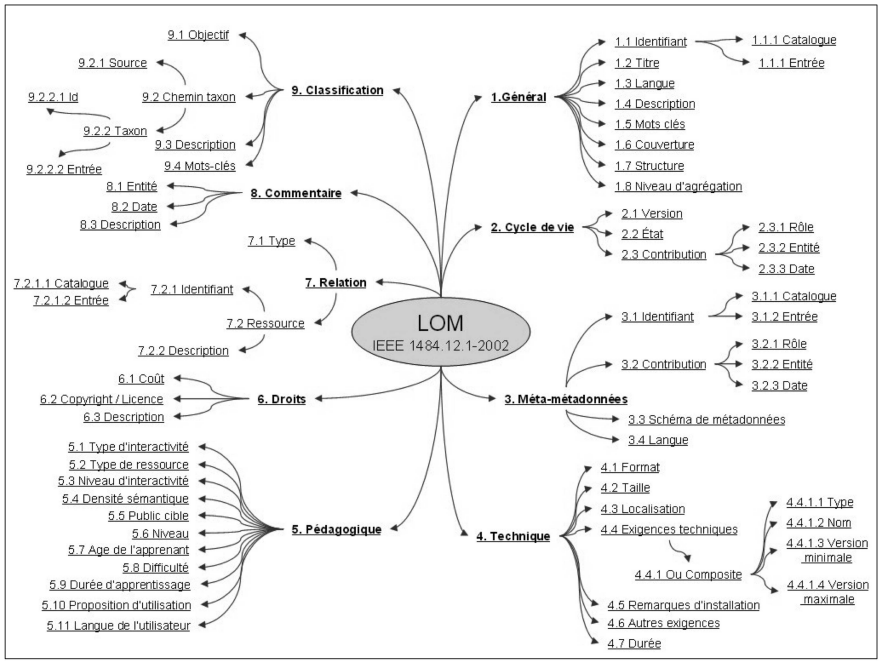


Fig. A.2 – Les éléments du Learning Object Metadata

Le nombre d'éléments de données défini au sein de ce schéma de structure de base est très important. Afin de simplifier les efforts d'implémentation, de nombreuses organisations recommandent un noyau sélectionné d'éléments de données identifiés [dLPJ04].

A.5 Représentation des standards de métadonnées

Les standards de métadonnées présentés dans cette annexe, et plus particulièrement le standard LOM vers lequel les autres spécifications convergent, doivent être représentés facilement afin de simplifier leur implémentation tout en permettant leur échange par le biais des technologies *web* actuelles. Ainsi, deux grandes représentations ont vu le jour : les représentations XML et RDF du LOM.

A.5.1 Représentation XML du standard LOM

Le langage XML [BPSM+04a] est devenu un standard pour échanger des données sur la toile. Associé à une définition de type de document, il permet à des utilisateurs de spécifier leur propres éléments afin de structurer le contenu de leurs ressources. XML vise à définir la

structure des documents et n'impose aucune interprétation commune de leur contenu, alors que les DTD décrivent la grammaire des documents. Un *binding*[14] XML du LOM a donc été défini afin de permettre l'échange d'instances de métadonnées conformes à ce standard entre différents systèmes informatiques. Ainsi, chaque élément défini par le LOM peut être représenté de manière structurée en utilisant le langage XML. La figure A.3 illustre un exemple d'une instance de métadonnées.

```
<?xml version="1.0" encoding="UTF-8" ?>
<lom xmlns="http://ltsc.ieee.org/xsd/LOMv1p0"
     xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
     xsi:schemaLocation="http://ltsc.ieee.org/xsd/LOMv1p0
     http://lkptest.ariadne-eu.org/ariadne/xs/lom.xsd">
  <general>
    <title>
      <string>Un exemple d'une instance LOM au format XML</string>
    </title>
    <description>
      <string>Cet exemple introduit le LOM/XML binding.</string>
    </description>
    <keyword>
      <string>lom xml binding</string>
    </keyword>
  </general>
  <lifeCycle>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>author</value>
      </role>
      <entity>BEGIN:VCARD\n VERSION:2.1\n N:Broisin; FN:Julien\n EMAIL ; PREF ; INTERNET:broisin@irit.fr\n
              TEL:33561638296\n TEL ; Fax:3316327996\n ORG:UPS; ToulouseIII\n ADR:118 route de Narbonne; 31000;
              Toulouse; France\n  END:VCARD\n
      </entity>
      <date>
        <dateTime>2002-06-03T12:00:00.0</dateTime>
      </date>
    </contribute>
  </lifeCycle>
  <technical>
    <format>text/html</format>
    <size>16384</size>
  </technical>
  <educational>
    <learningResourceType>
      <source>Ariadne</source>
      <value>Text</value>
    </learningResourceType>
  </educational>
</lom>
```

Fig. A.3 – Un exemple de représentation XML du LOM

Le *binding* XML du LOM présente cependant quelques lacunes comme les relations entre une DTD et le modèle de données du document qui sont perdues et qui ne peuvent pas être facilement retrouvées. D'autre part, il n'existe pas de moyen simple pour élargir ou réutiliser les métadonnées LOM de documents différents ; Resource Description Framework (RDF) vise à combler ces lacunes.

[14]Lien en français

A.5.2 Représentation RDF du standard LOM

Resource Description Framework est un langage de représentation des informations du *web* qui décrit les relations entre ressources en terme de propriété et de valeur. Ses principales caractéristiques sont les suivantes :

- **Langage commun** pour représenter des informations et plus particulièrement des métadonnées.
- **Facilite la création et l'échange** de métadonnées.
- **Permet de représenter des métadonnées** qui peuvent être gérées de façon décentralisée par des communautés indépendantes.
- **Permet d'étendre la description** de ressources existantes.
- **Langage neutre** de domaine.

Toute information décrite avec RDF est appelée *ressource*. Une *ressource* peut être une page *web*, un élément HTML ou XML, une collection entière de pages *web* ou un programme informatique. RDF identifie les ressources en utilisant les URI[15] et les décrit en terme de propriétés et de valeurs de propriété. Une propriété est une caractéristique spécifique utilisée pour décrire une ressource. Chaque propriété a une signification particulière, définit les valeurs qui peuvent lui être attribuées ainsi que les types de ressources qu'elle est capable de décrire. Une ressource spécifique, combinée à une propriété et une valeur, est appelée un *état* RDF.

RDF fournit un binding XML permettant d'écrire des graphes RDF sous forme de documents XML. A partir de ce binding XML, une initiative lancée par IMS tente de définir un binding LOM/RDF[16] à partir des éléments de métadonnées spécifiés dans le *Dublin Core Metadata Set*. En effet, la plupart des éléments du LOM sont sémantiquement similaires à ceux du Dublin Core, et l'Annexe B du standard LOM définit la correspondance entre ses éléments et ceux du DC. Le site Internet de IMS[17] donne un exemple de binding LOM/RDF.

[15] Uniform Resource Identifier
[16] http://www.imsglobal.org/rdf/
[17] http://www.imsglobal.org/rdf/RDF/lom-rdf1.rdf

Annexe B
Scénarios de gestion

Dans cette annexe, nous détaillons l'ensemble des créations/modifications spécifiques à chaque opération de recherche, consultation, téléchargement, importation et indexation de ressources pédagogiques.

B.1 Recherche

Lorsqu'un utilisateur recherche des ressources externes à partir de l'interface de la couche de Fédération ou de l'outil SILO, l'agent capture le login, le mot de passe (crypté), le nom, le prénom, l'adresse de courrier électronique et le rôle de cet utilisateur au sein du système, ainsi que le(s) mot(s) clé(s) renseigné(s) dans le formulaire électronique. De plus, les informations relatives au système d'apprentissage utilisé (son nom, son rôle, sa localisation, sa description et sa version) sont incluses dans un fichier de configuration. L'agent communique ensuite l'ensemble de ces données au CIM *Object Provider* qui effectue alors différentes opérations.

Au niveau de la supervision des utilisateurs, l'OP envoie une requête à l'OM afin de savoir si une instance CIM_Person correspondant à l'utilisateur est déjà présente au sein du référentiel. Si cette instance n'est pas retrouvée, toutes les instances relatives à la supervision des utilisateurs sont créées par l'OP puis acheminées vers l'OM avant d'être enregistrées dans le référentiel CIM :

- Une instance de la classe CIM_Person qui spécifie les nom, prénom et adresse de courrier électronique.
- Une instance de la classe CIM_OtherPersonInformation qui renseigne le(s) mot(s) clé(s) de recherche.
- Une instance de la classe CIM_MorePersonInfo afin d'établir l'association entre l'utilisateur et ses concepts recherchés.
- Une instance de la classe CIM_Role indiquant le rôle de l'utilisateur, par exemple "enseignant".

- Une instance de la classe CIM_MemberPrincipal afin d'associer le rôle avec l'utilisateur.
- Une instance des classes CIM_Identity et CIM_AssignedIdentity afin de faire correspondre l'identité de l'utilisateur au sein du LMS avec sa "personne".
- Une instance des classes CIM_Account (à partir du login et du mot de passe) et CIM_AccountIdentity pour associer le compte de l'utilisateur avec son identité.

La figure B.1 illustre la portion de code permettant à l'OP de créer une nouvelle instance de la classe CIM_Person puis de l'insérer dans le référentiel à travers l'OM.

```
/* Création d'un client HTTP pour se connecter à l'OM */
CIMNameSpace httpNameSpace = new CIMNameSpace("http://" + omLocation + "//interop");
CIMClient httpClient = new CIMClient(httpNameSpace, user, pwd);
/* Obtention du path de la classe CIM_Person au sein du référentiel */
CIMClass PersonClass = httpClient.getClass(new CIMObjectPath("CIM_Person"), false);
/* Cration d'une instance CIM_Person */
CIMInstance PersonInstance = PersonClass.newInstance();
/* Affectation d'une à la propriété CreationClassName */
PersonInstance.setProperty("CreationClassName", new CIMValue(PersonName)) ;
/* Affectation d'une à la propriété Name */
PersonInstance.setProperty("Name", new CIMValue(PersonName)) ;
/* Affectation d'une à la propriété GivenName */
PersonInstance.setProperty("GivenName", new CIMValue(PersonGivenName)) ;
/* Affectation d'une à la propriété Surname */
PersonInstance.setProperty("Surname", new CIMValue(PersonSurname)) ;
/* Affectation d'une à la propriété Mail */
PersonInstance.setProperty("Mail", new CIMValue(PersonEmail)) ;
/* Insertion de la nouvelle instance au sein du référentiel */
CIMObjectPath personPath = httpClient.createInstance(new CIMObjectPath(), PersonInstance);
```

FIG. B.1 – Création d'une instance de la classe CIM_Person

Dans le cas où l'instance CIM_Person de l'utilisateur existe déjà dans le référentiel, l'OP se contente de mettre à jour les classes précitées.

Au niveau du modèle de l'environnement d'apprentissage, le CIMOP vérifie l'existence de la classe EIAH_LearningManagementSystem ou EIAH_ContentManagementSystem (en fonction du système à l'origine de l'événement) par l'envoi d'une requête à l'OM. Dans le cas où l'instance correspondante n'est pas retrouvée, elle est créée par l'OP, ainsi qu'une instance de la classe d'association CIM_AccountOnSystem référençant les instances EIAH_LearningManagementSystem (ou EIAH_ContentManagementSystem) et CIM_Account précédemment produites. Dans le cas contraire, aucune opération spécifique n'est effectuée.

Les opérations décrites dans cette section sont également reproduites pour chacun des événements qui suivent, nous ne décrivons donc que les actions supplémentaires qui sont exécutées par l'OP et qui sont relatives aux ressources et cursus pédagogiques.

B.2 Consultation

Lorsqu'un utilisateur consulte les métadonnées d'une ressource d'apprentissage spécifique, l'agent collecte son identifiant, son titre et sa description, ainsi que les informations relatives au vivier de connaissance concerné par la recherche (mentionnées dans le fichier de configuration) ; l'agent transmet ensuite ces données à l'OP en invoquant la méthode receiveData. Celui-ci envoie alors une requête à l'OM afin de déterminer si l'instance correspondant à cet objet pédagogique est enregistrée dans le référentiel CIM. Si l'instance n'existe pas, l'OP crée une série d'instances :

- Une instance de la classe EIAH_LearningObject en attribuant la valeur *1* à la propriété ConsultationOcc et la *date du jour* à la propriété LastConsultation.
- Une instance de la classe d'association EIAH_HasConsulted qui référence la ressource et l'instance de la classe CIM_Identity précédemment créée (cf. section précédente) ; les propriétés NbConsultation et LastConsultation prennent respectivement les valeurs *1* et *date du jour*.
- Si l'instance de la classe EIAH_ContentManagementSystem correspondant au LOR n'existe pas, elle est crée par l'OP, ainsi qu'une instance EIAH_IsStoredBy référençant les instances correspondantes à la ressource et au LOR.

Si l'instance correspondant à l'objet pédagogique est présente dans le référentiel, l'OP invoque les méthodes IncreaseConsultation() et UpdateLastConsultation() de l'instance EIAH_LearningObject. Aussi, l'OP envoie une requête à l'OM pour savoir si l'utilisateur a déjà consulté cet objet pédagogique ; si oui, l'OP invoque les méthodes précitée sur l'instance EIAH_HasConsulted correspondante, sinon cette instance est construite en renseignant les propriétés NbConsultation et LastConsultation avec les valeurs *1* et *date du jour*.

B.3 Téléchargement

Lorsqu'un utilisateur télécharge un document sur son poste local, les opérations mentionnées dans la section précédente sont de nouveau exécutées, mais avec la classe EIAH_HasDownloaded et la propriété DownloadOcc de la classe EIAH_LearningObject. Les méthodes concernées sont naturellement IncreaseDownload() et UpdateLastDownload().

B.4 Importation

Lorsqu'un utilisateur importe une ressource externe au sein d'un cursus, les opérations de la section B.2 sont à nouveau exécutées, mais cette fois avec la classe EIAH_HasIntegrated et

les propriétés IntegrationOcc et LastIntegration de la classe EIAH_LearningObject ; les méthodes impliquées sont IncreaseIntegration() et UpdateLastIntegration().

De plus, l'agent recueille les informations relatives au cursus pédagogique courant (son nom, sa description et sa catégorie) et les transmet à l'OP. Celui-ci envoie alors une requête à l'OM afin de déterminer l'existence de l'instance EIAH_Courseware correspondante. Si cette instance n'est pas enregistrée dans le référentiel, l'OP la construit et crée également une instance de la classe EIAH_IsDeployedBy afin d'associer ce cursus avec la plate-forme d'apprentissage utilisée. Notons que l'identifiant de ce cursus est également mentionné au sein de l'instance EIAH_HasIntegrated afin de déterminer les cursus dans lesquels un utilisateur a intégré des ressources.

Dans la mesure où l'outil SILO ne permet pas d'intégrer d'objets au sein de cursus pédagogiques, cet événement ne peut être levé qu'à partir de la couche de Virtualisation.

B.5 Indexation

Le dernier scénario correspond à l'indexation d'un nouvel objet à la fois dans la plate-forme d'apprentissage et dans le vivier de connaissance (à l'heure actuelle, le KPS). Nous faisons une distinction entre l'indexation à partir de la couche de Virtualisation, et l'indexation à partir de l'outil SILO.

Dans le premier cas, les mêmes opérations que celles décrites dans la section précédente sont effectuées.

En revanche, lorsque l'indexation est effectuée à partir de l'outil SILO, l'OP effectue les opérations mentionnées dans le scénario de consultation, ainsi que celles exposées dans le scénario de téléchargement. En effet, l'indexation réalisée avec SILO insère le nouvel objet dans le KPS, mais cet objet n'est pas intégré dans un contexte d'apprentissage ; nous considérons en revanche que les métadonnées ont été consultées puisqu'elles ont été renseignées par l'indexeur, et que le document associé à ces métadonnées a été téléchargé puisque celui-ci a été visionné par ce même utilisateur.