

---

Dépôt Institutionnel de l'Université libre de Bruxelles /  
Université libre de Bruxelles Institutional Repository  
**Thèse de doctorat/ PhD Thesis**

**Citation APA:**

Mwepu Fyama, B. (2011). *Développement d'une méthodologie d'échange des métadonnées des objets numériques d'apprentissage pour une interopérabilité entre plates-formes d'elearning hétérogènes: cas de l'Université de Lubumbashi (R.D Congo) et ses partenaires belges* (Unpublished doctoral dissertation). Université libre de Bruxelles, Ecole polytechnique de Bruxelles – Informatique, Bruxelles.

**Disponible à / Available at permalink :** <https://dipot.ulb.ac.be/dspace/bitstream/2013/209855/4/bd901721-bcd1-4c23-99f7-01acb49cd8d8.txt>

---

(English version below)

Cette thèse de doctorat a été numérisée par l'Université libre de Bruxelles. L'auteur qui s'opposerait à sa mise en ligne dans DI-fusion est invité à prendre contact avec l'Université ([di-fusion@ulb.be](mailto:di-fusion@ulb.be)).

**Dans le cas où une version électronique native de la thèse existe, l'Université ne peut garantir que la présente version numérisée soit identique à la version électronique native, ni qu'elle soit la version officielle définitive de la thèse.**

DI-fusion, le Dépôt Institutionnel de l'Université libre de Bruxelles, recueille la production scientifique de l'Université, mise à disposition en libre accès autant que possible. Les œuvres accessibles dans DI-fusion sont protégées par la législation belge relative aux droits d'auteur et aux droits voisins. Toute personne peut, sans avoir à demander l'autorisation de l'auteur ou de l'ayant-droit, à des fins d'usage privé ou à des fins d'illustration de l'enseignement ou de recherche scientifique, dans la mesure justifiée par le but non lucratif poursuivi, lire, télécharger ou reproduire sur papier ou sur tout autre support, les articles ou des fragments d'autres œuvres, disponibles dans DI-fusion, pour autant que :

- Le nom des auteurs, le titre et la référence bibliographique complète soient cités;
- L'identifiant unique attribué aux métadonnées dans DI-fusion (permalink) soit indiqué;
- Le contenu ne soit pas modifié.

L'œuvre ne peut être stockée dans une autre base de données dans le but d'y donner accès ; l'identifiant unique (permalink) indiqué ci-dessus doit toujours être utilisé pour donner accès à l'œuvre. Toute autre utilisation non mentionnée ci-dessus nécessite l'autorisation de l'auteur de l'œuvre ou de l'ayant droit.

----- English Version -----

This Ph.D. thesis has been digitized by Université libre de Bruxelles. The author who would disagree on its online availability in DI-fusion is invited to contact the University ([di-fusion@ulb.be](mailto:di-fusion@ulb.be)).

**If a native electronic version of the thesis exists, the University can guarantee neither that the present digitized version is identical to the native electronic version, nor that it is the definitive official version of the thesis.**

DI-fusion is the Institutional Repository of Université libre de Bruxelles; it collects the research output of the University, available on open access as much as possible. The works included in DI-fusion are protected by the Belgian legislation relating to authors' rights and neighbouring rights. Any user may, without prior permission from the authors or copyright owners, for private usage or for educational or scientific research purposes, to the extent justified by the non-profit activity, read, download or reproduce on paper or on any other media, the articles or fragments of other works, available in DI-fusion, provided:

- The authors, title and full bibliographic details are credited in any copy;
- The unique identifier (permalink) for the original metadata page in DI-fusion is indicated;
- The content is not changed in any way.

It is not permitted to store the work in another database in order to provide access to it; the unique identifier (permalink) indicated above must always be used to provide access to the work. Any other use not mentioned above requires the authors' or copyright owners' permission.

---

**D 03837**

ULB LIBRE  
DE BRUXELLES



Université Libre de Bruxelles

Faculté des Sciences appliquées

Service d'Ingénierie Informatique et de la Décision (CoDE)

---

**Développement d'une méthodologie d'échange des métadonnées des objets numériques d'apprentissage, pour une interopérabilité entre plates-formes d'e-Learning hétérogènes : Cas de l'Université de Lubumbashi (R.D Congo) et de ses partenaires belges.**

---

Présenté par : **Blaise Mwepu FYAMA**

Promoteur : **Hugues BERSINI**

Co-Promotrice : **Françoise D'HAUTCOURT**

Thèse présentée en vue de l'obtention du titre de Docteur en Sciences de l'Ingénieur

Spécialité : Ingénierie Informatique.

Année Académique 2011-2012

Université Libre de Bruxelles



003482871

## TABLE DES MATIERES

|  |           |
|--|-----------|
| <b>TABLE DES MATIERES .....</b>  | <b>1</b>  |
| <b>Remerciements .....</b>   | <b>9</b>  |
| <b>CHAPITRE 1 : INTRODUCTION .....</b>   | <b>13</b> |
| 1.1 Enjeux de la thèse .....   | 13        |
| 1.2 Plan du travail et démarche .....  | 15        |
| 1.3 Les enjeux de l'interopérabilité des plates-formes e-Learning.....                           | 19        |
| 1.4 Développement d'ULLOXA et choix technologiques. ....   | 21        |
| 1.5 Ce que nous visons comme contribution.....   | 23        |
| <b>CHAPITRE 2 : PRESENTATION DU SUJET ET CONTEXTE DES TRAVAUX .....</b>                          | <b>26</b> |
| 2.1 Pertinence du sujet. ....  | 26        |
| 2.1.1 Contexte scientifique des travaux .....  | 26        |
| 2.1.2 Contexte du milieu de travail .....  | 28        |
| 2.2. Problématique.....  | 29        |
| 2.3. Objectifs de la thèse.....  | 32        |
| 2.4 De l'e-learning a Lubumbashi.....  | 33        |
| 2.4.1 Profil informatique ou disponibilité de l'outil informatique dans la ville de Lubumbashi.. | 34        |
| 2.4.2 Résultats détaillés d'enquêtes 2007-2009. ....   | 36        |
| 2.4.3 Analyse générale.....  | 37        |
| 2.4.3.1 Fonctionnement et nombre des ordinateurs.....  | 37        |
| 2.4.3.2 Taux de fréquentation des entités informatiques .....                                    | 39        |
| 2.4.3.3 Intérêt de la fréquentation.....   | 39        |
| 2.4.3.4 Et pour finir... ..  | 40        |
| 2.4.4 Bilans et réflexions.....  | 40        |
| 2.5 Hypothèses.....  | 42        |
| 2.6 Approche méthodologique .....  | 44        |
| <b>CHAPITRE 3. LES ENJEUX DE L'INTEROPERABILITE.....</b>   | <b>46</b> |
| 3.1 Problématique d'interopérabilité des plates-formes d'e-learning.....                         | 46        |
| 3.2 Processus et acteurs.....  | 47        |
| 3.2.1 Différents acteurs de la normalisation .....   | 50        |
| 3.2.1.1 AICC ( <i>Aviation Industry Computer-Based Training Committee</i> ).....                 | 50        |



|  |    |
|--|----|
| 3.2.1.2 Advanced Distributed Learning .....  | 50 |
| 3.2.1.3 IMS Global Learning Consortium.....  | 51 |
| 3.2.1.4 Dublin Core Metadata Initiative .....  | 52 |
| 3.2.1.5 Institute of Electrical and Electronics Engineers .....  | 52 |
| 3.2.1.6 Joint Technology Committee, Subcommittee on Standards for Learning, Education,<br>and Technology .....     | 53 |
| 3.2.1.7 Comité Européen de Normalisation .....   | 54 |
| 3.2.1.8 Bureau de la normalisation en Belgique .....   | 54 |
| 3.2.2 Conclusion de la section .....   | 55 |
| 3.3 Concepts techniques généraux d'interopérabilité en e-learning .....  | 57 |
| 3.3.1 Introduction .....   | 57 |
| 3.3.2 Définitions des différents concepts d'interopérabilité.....  | 58 |
| 3.4 Aperçu de IMS et SCORM.....  | 62 |
| 3.4.1 Historique de IMS .....  | 62 |
| 3.4.2 Les spécifications IMS .....   | 63 |
| 3.4.2.1 IMS Content Packaging.....   | 64 |
| 3.4.2.1.1 Description du IMS Content Package: .....  | 64 |
| 3.4.2.1.1.1 Le fichier manifeste .....   | 65 |
| 3.4.2.1.1.2 Les ressources physiques .....   | 66 |
| 3.4.2.1.2 Package Interchange File (PIF).....  | 66 |
| 3.4.3 Bref aperçu du SCORM .....   | 67 |
| 3.5 Expérimentations d'e-learning à l'Université de Lubumbashi face a la problématique<br>d'interopérabilité ..... | 68 |
| 3.5.1 Pourquoi une expérimentation a l'UNILU?.....   | 68 |
| 3.5.2 Présentation générale des expérimentations .....   | 69 |
| 3.5.2.1 Objectifs de l'expérimentation.....  | 70 |
| 3.5.2.2 Caractéristiques du contexte d'apprentissage .....   | 71 |
| 3.5.2.3 Public cible (acteurs et groupes) .....  | 71 |
| 3.5.2.4 Domaine d'application .....  | 72 |
| 3.5.2.5 Description de la tâche.....   | 73 |
| 3.5.2.6 Dispositif technologique.....  | 73 |
| 3.5.2.7 Aspects méthodologiques .....  | 74 |
| 3.5.3 Analyse de l'e-learning a l'Université de Lubumbashi.....  | 75 |
| 3.5.3.1 Analyse des interactions.....  | 75 |



|   |           |
|---|-----------|
| 3.5.3.2 Utilisation des outils au cours des activités .....                                   | 78        |
| 3.5.3.3 Analyse des échanges avec le tuteur par catégories des interventions.....             | 79        |
| 3.5.3.4 Bilan : Comportements émergents, usage effectif des outils et scénario narratif. .... | 82        |
| 3.6 Enjeux de l'interopérabilité a l'Université de Lubumbashi .....                           | 83        |
| <b>CHAPITRE 4. ETUDE DESCRIPTIVE DU STANDARD SCORM .....</b>                                  | <b>86</b> |
| 4.1 Introduction .....  | 86        |
| 4.2 Historique de SCORM.....  | 87        |
| 4.3 Les raisons du succès de SCORM .....  | 88        |
| 4.3.1 Positionnement par rapport aux autres acteurs .....                                     | 88        |
| 4.3.2 Avantages de SCORM .....  | 90        |
| 4.4 Objectifs de SCORM .....  | 92        |
| 4.4.1 Les laboratoires.....   | 92        |
| 4.4.2 Le modèle SCORM .....   | 93        |
| 4.5 Concepts généraux de SCORM.....   | 95        |
| 4.5.1 Modèle d'agrégation du contenu .....  | 95        |
| 4.5.1.1 Actif (Asset) .....   | 96        |
| 4.5.1.2 Sharable content Object (SCO).....  | 97        |
| 4.5.1.3 Activités .....   | 99        |
| 4.5.1.4 Organisation de contenu .....   | 99        |
| 4.5.1.5 Représentation de la structure d'un contenu .....                                     | 101       |
| 4.5.1.5.1 Les composants du contenu d'un « package ».....                                     | 102       |
| 4.5.1.5.2 Le fichier manifeste .....  | 102       |
| 4.5.1.5.3 Les fichiers physiques .....  | 104       |
| 4.5.1.6 Les métadonnées .....   | 105       |
| 4.5.2 L'environnement d'exécution SCORM.....  | 107       |
| 4.5.2.1 Lancement de ressources d'apprentissage .....   | 108       |
| 4.5.2.1.1 Actifs.....   | 109       |
| 4.5.2.1.2 SCO (Sharable Content Object) .....   | 109       |
| 4.5.2.2 API (Application Programming Interface) .....   | 109       |
| 4.5.2.2.1 Description de l'API de communication SCO-Plate-forme .....                         | 110       |
| 4.5.2.2.1.1 Fonctions de session.....   | 111       |
| 4.5.2.2.1.2 Fonctions de transfert des données.....   | 112       |
| 4.5.2.2.1.3 Fonctions auxiliaires .....   | 113       |

|   |            |
|---|------------|
| 4.5.3 Le séquençement.....  | 115        |
| 4.5.3.1 Structure de contenu et arbre d'activité.....                                 | 115        |
| 4.5.3.2 Les concepts du séquençement .....  | 115        |
| 4.5.3.2.1 Arbre d'activités .....   | 115        |
| 4.5.3.2.2 Cluster .....   | 116        |
| 4.5.3.2.3 Activité d'apprentissage .....  | 118        |
| 4.5.3.2.4 Tentatives .....  | 119        |
| 4.5.3.3 Démarrage et Arrêt d'une session de séquençement.....                         | 119        |
| 4.5.3.4 Objectifs pédagogiques .....  | 120        |
| 4.5.3.5 Le tracking .....   | 120        |
| 4.6 Utilisation de SCORM dans nos travaux.....  | 121        |
| <b>CHAPITRE 5 : ETUDE COMPARATIVE DES STRUCTURES DES CONTENUS MOODLE-DOKEOS .....</b> | <b>122</b> |
| 5.1 Justification du choix.....   | 122        |
| 5.2 Architecture et présentation des plates-formes.....                               | 124        |
| 5.2.1 Architecture des plates-formes .....  | 124        |
| 5.2.1.1 Présentation de Moodle.....   | 125        |
| 5.2.1.1.1 Introduction.....   | 125        |
| 5.2.1.1.2 Page d'accueil d'un cours Moodle et fonctionnalités offertes.....           | 126        |
| 5.2.1.1.3 Notre bilan sur Moodle .....  | 128        |
| 5.2.1.1.4 Les points faibles de la plate-forme .....                                  | 130        |
| 5.2.1.2. Présentation Dokeos .....  | 131        |
| 5.2.1.2.1 Introduction.....   | 131        |
| 5.2.1.2.2 Les fonctionnalités de Dokeos.....  | 131        |
| 5.2.1.2.3 Notre bilan de Dokeos.....  | 135        |
| 5.2.1.2.4 Les points faibles .....  | 136        |
| 5.3 Etude comparative et catégorielle des fonctionnalités Moodle vs Dokeos .....      | 137        |
| 5.3.1 Outils de communication.....  | 138        |
| 5.3.2 Outils d'organisation.....  | 138        |
| 5.3.3 Outils de partage. ....   | 138        |
| 5.3.4 Outils de contrôle .....  | 139        |
| 5.3.5 Aspects pédagogiques .....  | 140        |
| 5.3.6 Aspects techniques.....   | 140        |
| 5.3.7 Aspects financiers.....   | 141        |



|   |            |
|---|------------|
| 5.3.8 Discussion .....  | 141        |
| 5.4 Description d'un contenu Moodle .....   | 146        |
| 5.4.1 Agrégation d'un Package Moodle.....   | 146        |
| 5.4.1.1 Le manifeste moodle.xml .....   | 147        |
| 5.4.1.1.1. La section « INFO » .....  | 147        |
| 5.4.1.1.2 La section « ROLES » .....  | 149        |
| 5.4.1.1.3 La section « COURSE » .....   | 150        |
| 5.4.1.1.3.1 Les éléments traditionnels .....  | 150        |
| 5.4.1.1.3.2 Éléments conditionnels .....  | 151        |
| 5.4.1.2 Bilan de moodle.xml .....   | 152        |
| 5.5 Description d'un contenu Dokeos.....  | 152        |
| 5.5.1 Description du fichier de sauvegarde d'un cours. ....                         | 153        |
| 5.5.2 Description du fichier d'export au format SCORM de Dokeos .....               | 153        |
| 5.5.3 Constitution d'un package Dokeos : .....                                      | 154        |
| 5.6 Caractérisation des conditions de compatibilité .....                           | 156        |
| 5.7 Méthodologie pratique de mise en œuvre .....                                    | 159        |
| <b>CHAPITRE 6 : IMPLEMENTATION DE L'INTEROPERABILITE STATIQUE .....</b>             | <b>162</b> |
| 6.1. Introduction .....   | 162        |
| 6.2 Les Métadonnées des objets pédagogiques .....                                   | 163        |
| 6.2.1 Définitions et caractéristiques des métadonnées des objets pédagogiques ..... | 163        |
| 6.2.2 A quoi servent les métadonnées des objets pédagogiques .....                  | 165        |
| 6.3. L'objet pédagogique.....   | 166        |
| 6.3.1 Définition de l'objet pédagogique .....                                       | 166        |
| 6.3.2. Les caractéristiques de l'objet pédagogique .....                            | 168        |
| 6.3.2.1 L'intention pédagogique .....   | 168        |
| 6.3.2.2 La granularité.....   | 168        |
| 6.3.2.3 La réutilisabilité .....  | 169        |
| 6.3.2.4 L'agrégation.....   | 169        |
| 6.3.2.5 L'accessibilité.....  | 170        |
| 6.3.2.6 L'interopérabilité .....  | 172        |
| 6.3.3 SCORM et LOM .....  | 172        |
| 6.4 XML éléments théoriques .....   | 173        |
| 6.4.1 Le langage XML .....  | 173        |



|   |            |
|---|------------|
| 6.4.2 XML et usages du document .....   | 176        |
| 6.4.2.1 Usages du document et XML.....  | 176        |
| 6.4.2.2 Traitement DOM .....  | 176        |
| 6.5 Transformation contenu Moodle en contenu SCORM .....  | 177        |
| 6.5.1 SCORM XML Binding.....  | 178        |
| 6.5.1.1 Présentation de SCORM XML Binding et équivalences Moodle/SCORM.....                     | 178        |
| 6.5.1.1.1 L'élément <manifest>.....   | 180        |
| 6.5.1.1.2 L'élément <metadata> .....  | 181        |
| 6.5.1.1.3 L'élément <organizations> .....   | 182        |
| 6.5.1.1.4 L'élément <resources> .....   | 183        |
| 6.5.2 Concepts et exigences de conformité SCORM .....   | 185        |
| 6.5.2.1 Profils d'application des contenus .....  | 186        |
| 6.5.2.2 Règles pour un package.....   | 187        |
| 6.5.2.3 Règles pour un SCO .....  | 188        |
| 6.5.2.4 Règles pour actifs (Asset) .....  | 190        |
| 6.5.3 Transformation technique.....   | 190        |
| 6.6 Modélisation de l'architecture de l'outil de transformation des données.....                | 195        |
| 6.7 Présentation de l'outil et transfert d'un cours .....                                       | 198        |
| 6.7.1 Introduction .....  | 198        |
| 6.7.2 Tâches à accomplir par l'outil informatique.....  | 199        |
| 6.7.3 Traitement des métadonnées : agrégation et réadaptation des ressources .....              | 204        |
| 6.7.4 Illustration transformation et transfert du cours.....                                    | 206        |
| 6.7.4.1 Interface de départ dans Moodle.....  | 206        |
| 6.7.4.2 Interfaces phases de transformations.....   | 207        |
| 6.7.4.3 Interface a l'arrivée sur Dokeos .....  | 209        |
| 6.7.5 Bilan de la transformation. ....  | 209        |
| <b>CHAPITRE 7 : IMPLEMENTATION DE L'INTEROPERABILITE DYNAMIQUE. ....</b>                        | <b>212</b> |
| 7.1 Introduction .....  | 212        |
| 7.2 Présentation des outils forum de Moodle et Dokeos. ....                                     | 213        |
| 7.2.1 Présentation du forum Dokeos.....   | 213        |
| 7.2.2 Présentation du forum Moodle .....  | 214        |
| 7.3 Choix de la technique à appliquer pour une interopérabilité dynamique : les services Web .. | 216        |
| 7.3.1 Contexte technique .....  | 216        |

|   |            |
|---|------------|
| 7.3.2 Justification du choix des services Web .....                                   | 217        |
| 7.3.3 Principes et fonctionnement des services Web .....                              | 218        |
| 7.3.3.1 Définition du service Web .....   | 218        |
| 7.3.3.2 Domaines d'utilisation.....   | 220        |
| 7.3.3.3 Infrastructure des services Web.....  | 220        |
| 7.3.3.3.1 La couche de transport.....   | 221        |
| 7.3.3.3.2 La couche d'échange des messages XML .....                                  | 221        |
| 7.3.3.3.2.1 XML-RPC .....   | 221        |
| 7.3.3.3.2.2 SOAP .....  | 221        |
| 7.3.3.3.3 La couche de description: WSDL .....  | 222        |
| 7.4 Proposition de mise en œuvre du service Web des forums .....                      | 225        |
| 7.4.1 Description du processus métier .....   | 226        |
| 7.4.2 Invocation du service.....  | 229        |
| 7.5 Conclusion du chapitre .....  | 231        |
| <b>CHAPITRE 8 : CONCLUSIONS ET PERSPECTIVES .....</b>                                 | <b>233</b> |
| 8.1 Parcours des réalisations .....   | 233        |
| 8.1.1 Une étude du contexte d'application.....  | 233        |
| 8.1.2 Une étude technique .....   | 235        |
| 8.2 Notre contribution .....  | 237        |
| 8.3 Perspectives des recherches .....   | 238        |
| <b>Bibliographie.....</b>   | <b>241</b> |
| <b>ANNEXE A : PRÉSENTATION DES SITES.....</b>   | <b>253</b> |
| A.1. Campus de la Kasapa .....  | 253        |
| A.2. Quartier Golf .....  | 253        |
| A.3. Quartier Carrefour .....   | 254        |
| A.4 Kimbangu-L.D Kabila .....   | 254        |
| A.5. Révolution-Kimbangu-Lumumba-Kamanyola.....                                       | 255        |
| A.6 L.D Kabila-Likasi .....   | 256        |
| <b>ANNEXE B : TABLEAUX DE COMPARAISON DES OUTILS.....</b>                             | <b>257</b> |
| Les tableaux non repris ici sont en intégralité dans le texte du présent travail..... | 257        |
| B.1 Outils de contrôle .....  | 257        |
| B.2 Outils de partage.....  | 257        |
| B.3 Aspects pédagogiques .....  | 258        |



|  |            |
|--|------------|
| B.4 Aspects techniques .....                                     | 259        |
| <b>ANNEXE C : QUELQUES NOTIONS UTILES SUR XML .....</b>          | <b>260</b> |
| C.1 Structure d'un document XML.....                             | 260        |
| C.1.1 L'arbre d'éléments.....                                    | 260        |
| C.1.1.1 Élément racine .....                                     | 260        |
| C.1.1.2 Les éléments.....  | 260        |
| C.1.1.3 Les attributs.....                                       | 261        |
| C.1.1.4 Les entités .....  | 261        |
| C.1.1. Les sections CDATA.....                                   | 262        |
| C.1.2 Modèles de documents : DTD et schémas .....                | 262        |
| C.1.4 XPath.....   | 263        |
| C.1.5 XSL et XSLT .....  | 263        |
| <b>ANNEXE D : QUELQUES NOTIONS DE SERVICES WEB.....</b>          | <b>266</b> |
| D.1 La couche de transport.....                                  | 266        |
| D.1.1 Le protocole HTTP .....                                    | 266        |
| D.1.2 Structure des messages HTTP .....                          | 266        |
| D.1.3 La couche d'échange des messages XML.....                  | 267        |
| D.1.3.1 XML-RPC.....   | 267        |
| D.1.3.2 SOAP.....  | 268        |
| D.1.3.3 L'enveloppe SOAP .....                                   | 268        |
| D.1.3.4 Convention d'appel des procédures à distance (RPC) ..... | 270        |
| D.2 La couche de description : WSDL.....                         | 271        |
| D.2.1 L'élément "definitions" .....                              | 271        |
| D.2.2 L'élément "message" .....                                  | 271        |
| D.2.3 L'élément "portType" .....                                 | 272        |
| D.2.4 Élément "binding" .....                                    | 272        |
| D.2.5 L'élément "service" .....                                  | 273        |
| D.3 Couche de découverte des services web.....                   | 273        |
| D.4 Codes de l'implémentation du service .....                   | 274        |
| D.5 Codes de description WSDL du service.....                    | 275        |



## Remerciements

Ce mémoire de thèse représente l'achèvement de quatre années d'un long, fastidieux, mais ô combien exaltant et enrichissant travail qui n'aurait pu voir le jour sans la participation, l'aide, le soutien, les conseils, ou encore la présence de nombreuses personnes. D'avance je m'excuse pour les noms oubliés et donne ci-après les non-oubliés.

Tout commence au mois de novembre 2004, lorsqu'à l'issue d'un stage de la Coopération Universitaire au Développement (CUD), je fais la rencontre de Pascal Francq qui après avis de M. Alain Delchambre donnera son aval pour que cette étude soit menée.

De prime abord, un immense merci à Pascal Francq pour m'avoir encadré avec autant de sérieux, de rigueur scientifique, mais aussi d'humour, de gentillesse, de générosité, d'humanité et d'humilité. Merci de m'avoir toujours fait confiance, sans jamais m'avoir sous-estimé et de m'avoir toujours encouragé et même soutenu dans les moments obscurs, les moments de faiblesse, les moments de doute qui ont parsemés ces quatre années... Une litanie de mots ne suffirait pas pour exprimer l'indicible profondeur de ma reconnaissance et mon émotion, seulement ce petit mot merci, merci, et merci pour tout Pal.

Je remercie également Madame le Professeur Françoise D'Hautcourt, de m'avoir honoré en acceptant d'être ma co-promotrice de thèse. Merci d'avoir apporté de l'énergie, à la fois lumineuse et chaleureuse, à ce travail. Merci de m'avoir guidé avec autant de clairvoyance, de rigueur, de sollicitude, d'affection, et de protection. Merci de m'avoir rassuré, de m'avoir permis d'y croire. Enfin puis-je oser dire merci d'avoir été maternelle avec moi...

Je remercie le Professeur Alain Delchambre, de m'avoir fait l'honneur d'accepter de présider mon comité d'encadrement et le jury de ma soutenance de thèse. Votre appui moral et votre parrainage ont été très déterminants pour l'achèvement de cette thèse. Je vous ai connu au Congo il y a quelques années, lors de vos visitings, venant soutenir notre

enseignement dont le contexte est particulier... Je témoigne et confirme encore ici que vous êtes resté un ami du Congo.

Je remercie le Professeur Hugues Bersini, de m'avoir fait l'immense honneur d'être mon promoteur de thèse et de m'avoir accueilli chaleureusement au sein du laboratoire IRIDIA (Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle). Merci du soutien, de m'avoir toujours rassuré que tout irait bien, de la protection, et de la disponibilité. J'en profite pour souligner que je serai toujours admirateur.

Je remercie M. Philippe Bouillard, Professeur de l'Université Libre de Bruxelles, de m'avoir également fait l'honneur de faire parti de mon jury et de juger mes travaux.

Je remercie M. Pierre Manneback, Professeur de l'Université de Mons, de m'avoir également fait l'honneur de faire parti de mon jury et de juger mes travaux.

Je remercie la CTB (Coopération Technique Belge), d'avoir financé cette thèse. Merci à l'Etat belge qui par son agence au développement participe réellement à l'élimination de la pauvreté (en tous sens du terme) dans le monde. Merci à la CTB, qui, en finançant cette thèse, m'a offert l'opportunité d'apporter une contribution, pour une société africaine qui donne aux générations actuelles et futures les moyens de construire un monde durable et équitable.

Mes prochains remerciements s'adressent aux anciens collègues du bâtiment D, Faiza Gaultier, David Wartel et plus spécialement à Seth Van Hooland, dont le soutien, les conseils, l'amitié, l'humilité, l'humanisme et la grande générosité resteront à jamais gravés dans ma mémoire. Merci d'avoir été avec moi en toutes circonstances. Merci de la fraternité inconditionnelle.

Je remercie aussi ma maison d'accueil en Belgique, SETM (Solidarité Etudiants du Tiers Monde), de m'avoir offert un cadre adéquat de logement durant ces quatre années. Merci de la chaleur humaine et de l'assistance solidaire et sociale. Avec mention spéciale



pour Florence Binon, qui par sa relecture m'a permis d'améliorer l'orthographe du présent texte.

Ma dernière catégorie de remerciements se repartit des deux côtés :

Du Côté Belgique, pour avoir contribué pour peu ou prou à rendre mon séjour agréable loin des miens, je remercie du fonds du cœur et avec la plus grande gratitude : Da Pierrette, Chantal Diaki, Shannon, Ya Fifi et son mari, Natacha Tambwe, Murie Decreton, Usha Zammit, Papa Jeannot le seul représentant local des Mulongo.

Du côté de la mère patrie, vous avez été nombreux durant ce parcours à m'apporter des encouragements, un soutien moral, financier, et matériel. Grâce à vous j'ai pu persévérer, je m'en vais donc vous dire du fond de mon cœur, un grand Merci. Je pense donc à : Eric Kitanic frangin de toujours, Daddy Mbumb, Francis Tshipeng, Arthur Kabila Mutonkole fidèle des fidèles, Akilimali AK, Musole Riziki Stone, Pr Khang Mat, Vieux Perry Balloy, Laura Kimemwenze, Joseph Leonard, Omalanga Pele, Flory Kiseya, Daniel Kalufya. La ligue de petits frères, Joe Shimbi, Gauthier Mabika, Acclamer Petu, Yves Kabwika Papy, Alex Matata, Cedrick Kikunda, Yves Mulongo, Horizon Makesa.

Enfin pour ma famille, merci Maman toi qui es toujours là, reçois des honneurs mérités à travers ce travail. Merci Papa Jean, Papa Huit enfin un deuxième professeur est là. Paix à ton âme Tante Astride toi chez qui tout a commencé...



*Aussitôt qu'une pensée vraie est entrée dans notre esprit, elle jette une  
lumière qui nous fait voir une foule d'autres objets que nous  
n'apercevions pas auparavant.*

*François René de Chateaubriand*

*Vous connaîtrez la vérité... et cette dernière vous rendra libre...*

*A Pierrette, Oriane, Maeva et Lisette FYAMA je dédie ce travail.*

*Merci pour tout... Quelle que soit la durée de la nuit, le soleil fini par apparaître.*

## CHAPITRE 1 : INTRODUCTION

### 1.1 Enjeux de la thèse

Le travail initié ici s'inscrit dans le cadre de la coopération nord-sud, plus spécialement la Coopération Technique Belge (CTB) dans son volet de la promotion de l'enseignement supérieur dans les pays en développement du sud, au travers de la recherche des voies et moyens pour favoriser les conditions capables de permettre un développement durable de ces pays. Une des voies utilisées pour atteindre cet objectif est l'éducation, la lutte contre l'illettrisme, l'amélioration et l'incrémentation du niveau de l'enseignement supérieur et universitaire. Afin de garantir la formation des cadres intellectuels pouvant relever le défi d'un développement durable ainsi que la recherche scientifique pour donner des réponses locales aux divers et multiples problèmes que connaît cette partie du monde.

Pour ce qui est de la République Démocratique du Congo, plusieurs partenariats de coopération sont établis entre les universités locales congolaises et les universités belges, notamment en assurant la formation des enseignants, l'encadrement des chercheurs et doctorants ainsi que l'envoi de professeurs belges pour assurer des cours dans les universités congolaises. Ces échanges ont pour but d'améliorer la qualité de l'enseignement, d'assurer une mise à jour des contenus des cours et la bonne formation des étudiants en approchant le plus possible les standards internationaux. C'est dans cet ordre d'idées que l'Université Libre de Bruxelles (ULB) a initié une étude visant à déterminer les chances de réussite d'un projet d'apprentissage en ligne<sup>1</sup> (e-Learning) en partenariat avec l'Université de Lubumbashi ainsi que les moyens techniques et technologiques garantissant la réussite d'un tel projet.

A l'état actuel des choses, il convient de souligner que le parlement congolais, lors d'une plénière a approuvé en date du 5 janvier 2011, le rapport de la Commission

---

<sup>1</sup> Dans la suite du document nous utiliserons les termes « apprentissage en ligne » ou « e-learning » avec la même signification.

sociale et culturelle relatif à l'examen et à l'adoption du projet de loi portant sur l'organisation et le fonctionnement de l'enseignement national où il est prévu clairement, entre autres, dans un volet révélateur que : *« A la pointe de l'évolution technologique et des questions cruciales mondiales, cette loi va introduire au sein de l'enseignement national, les technologies de l'information et de la communication facilitant notamment l'enseignement ouvert et à distance d'une part, et initier les élèves et les étudiants congolais au développement durable et à la lutte contre les changements climatiques d'autre part »*.<sup>2</sup>

L'enjeu de cette thèse se situe au niveau de l'utilisation des outils que nous offrent les nouvelles technologies de l'information et de la communication dans l'enseignement afin d'une part, de mettre à jour les enseignements en Afrique subsaharienne et, d'autre part, de favoriser des échanges de contenus de formation afin de permettre aux étudiants du sud d'avoir accès à des cours de professeurs du nord surtout pour les domaines dans lesquels il n'existe pas de compétences locales. Ceci implique une connaissance des outils à utiliser, une adaptation de ces derniers au contexte local de formation, l'environnement étant un facteur déterminant dans le déroulement d'un enseignement médiatisé par les TIC (Pernin, et al., 2004), et le développement des moyens informatiques (technologiques) à mettre en œuvre pour permettre un échange quasi automatique des contenus des cours et services entre des plates-formes hétérogènes<sup>3</sup>.

Toute la démarche de ce travail sera celle qui nous mènera à la méthodologie de conception et à l'implémentation d'une application informatique qui pourra permettre les échanges automatiques des contenus et services d'apprentissage entre la plate-forme de l'Université de Lubumbashi et celles de ses partenaires belges après avoir, en amont,

<sup>2</sup> Rapport de la séance parlementaire le 05-1-2011, République Démocratique du Congo.

<sup>3</sup> Nous appelons plates-formes « hétérogènes » celles dont le modèle de conception, la structure et l'architecture sont différentes.



identifié un format ou un scénario de cours conforme (ou adapté) au contexte local d'enseignement.

## 1.2 Plan du travail et démarche

Ainsi une démarche a été conçue pour mener les travaux et atteindre les objectifs de la thèse à travers les différents chapitres qui constituent ce travail :

1. Introduction, le présent chapitre montre les enjeux d'une telle étude et sa visée.
2. Une présentation du sujet et une définition du contexte des travaux.

Nous présenterons ici le sujet, sa pertinence et son apport par rapport au milieu sur lequel il s'applique, les objectifs visés à travers cette thèse, la problématique posée par rapport au contexte, les approches conceptuelles et méthodologiques qui constituent des pistes de recherches potentielles.

Nous parlerons d'e-learning à l'Université de Lubumbashi puisque c'est finalement le milieu d'application de la présente thèse. Ceci nous amène à nous poser quelques questions auxquelles nous comptons apporter des réponses, en nous appuyant sur les résultats de deux enquêtes menées de 2007 à 2010, à savoir :

- de l'apprentissage en ligne à l'UNILU, est-ce vraiment possible ?
- quelle infrastructure informatique offre le support pour permettre à un étudiant d'effectuer de l'apprentissage en ligne ?
- quels sont les paramètres à observer pour veiller à une réussite d'un projet d'apprentissage en ligne à Lubumbashi ?

Dans ce volet du travail, nous aborderons également les aspects suivants : la pertinence du sujet, les objectifs visés de la thèse, le contexte et les chances de l'apprentissage en ligne à Lubumbashi, l'analyse et l'interprétation des résultats des enquêtes 2007-2010 qui visent à évaluer les chances d'une entreprise d'enseignement

médiatisé par l'outil informatique. Une analyse générale de la question sera ensuite abordée et sera suivie d'un bilan.

La problématique sera à son tour décortiquée pour étayer de manière très précise les aspects retenus de la question qui sera traitée dans le présent travail. Viendront ensuite les hypothèses laissant entrevoir les pistes de solutions, ou les voies à suivre pour arriver à bout de la problématique. L'approche méthodologique déterminera à son tour comment, avec quels outils et en s'appuyant sur quels concepts, nous allons atteindre les objectifs de la thèse.

### 3. Les enjeux de l'interopérabilité en e-learning

Ce chapitre aura pour objectifs d'offrir à la fois une vision panoramique et actualisée de la problématique des normes et standards ainsi que de l'interopérabilité entre les plates-formes d'apprentissage en ligne. Le sujet de notre thèse tourne en effet dans sa formulation autour de deux axes principaux : d'une part, l'interopérabilité entre les plates-formes hétérogènes dans leurs conceptions, et d'autre part, l'identification des outils permettant de dégager un format de contenu adapté à l'environnement c'est-à-dire au contexte local.

Cette partie se focalisera sur la problématique de l'interopérabilité entre plates-formes telle que posée de nos jours, les enjeux et défis à relever ainsi que sur les différents concepts et méthodologies permettant d'aborder cette question.

### 4. Étude approfondie du standard SCORM

Le standard émergeant dans le domaine de l'interopérabilité entre plates-formes d'apprentissage en ligne est le SCORM (*Sharable Content Object Reference Model*). Il nous paraît donc important de le regarder de près et d'en comprendre les rouages et les concepts. Ceci pour une utilisation efficiente dans le cas d'étude spécifique que nous allons développer comme base et fondation d'une application informatique future d'interopérabilité.



Nous l'aborderons en partant d'abord de sa naissance et du contexte de son évolution. Nous développerons ensuite les concepts de base du SCORM afin de mieux se l'approprier et l'utiliser à bon escient en dégagant notamment avec précision quels sont les aspects qui peuvent intervenir dans notre travail de thèse.

## 5. Analyse du cas d'interopérabilité Moodle-Dokeos<sup>4</sup>

La question de l'interopérabilité en e-learning est une question très vaste qui implique plusieurs philosophies d'usages, plusieurs acteurs, plusieurs écoles pédagogiques, plusieurs ontologies, plusieurs appréhensions du sujet qui donnent au problème une certaine complexité. C'est partant de là que nous nous sommes focalisés sur un cas particulier à savoir Moodle-Dokeos afin d'apporter une contribution dans la réussite de l'établissement de l'apprentissage en ligne dans l'enseignement supérieur en République Démocratique du Congo en général, et dans la ville de Lubumbashi en particulier, où Dokeos nous a servi au cours de ces cinq dernières années comme plateforme d'e-learning.

Nous passerons successivement par une identification du problème réel que pose l'interopérabilité entre les plates-formes de l'Université de Lubumbashi et celles de ses partenaires belges, tout en nous focalisant uniquement sur le cas des plates-formes Moodle et Dokeos. Pour étudier en profondeur l'interopérabilité, nous passerons par un décryptage de l'ossature des contenus de chacune des plates-formes ainsi qu'une analyse systémique et systématique de chaque élément des contenus. Nous poursuivrons aussi par une comparaison des deux contextes de contenus afin de caractériser leurs conditions de compatibilité et de poser les bases technologiques de l'implémentation d'une application d'interopérabilité assurant la transformation automatique des contenus ainsi que le transfert des contenus d'une plate-forme vers une autre.

---

<sup>4</sup> Moodle et Dokeos sont deux plates-formes open source qui sont présentées en profondeur au chapitre 5.

Ce travail a aussi pour intention d'être une prémisse à un projet de conception d'une application informatique à développer que nous nommons ULLOXA pour *University of Lubumbashi Learning Object Exchange Application*. Pour mener à bien le projet ULLOXA, nous commencerons par présenter les principes de base du fonctionnement de l'application, les objectifs que l'utilisation de l'application nous permet d'atteindre par rapport à l'interopérabilité qu'elle est appelée à assumer, la méthodologie de conception de l'application qui montrera sur quelles bases théoriques, quels concepts, quelles méthodes des sciences informatiques repose la démarche de développement de l'application, et la description de l'architecture de l'outil pour illustrer un modèle d'interopérabilité. Nous montrerons les différents modules répartis sur les différentes couches sur lesquelles ils interviennent pour un fonctionnement global de tout le système. Pour les objectifs sur le court, moyen et long terme, nous nous arrêterons à poser les bases de conception et un développement à titre démonstratif du fonctionnement futur de l'outil<sup>5</sup>. Nous tâcherons enfin d'évaluer les différentes pistes de recherche que peuvent engendrer l'application pour des problématiques futures à traiter.

## 6. Interopérabilité statique

Nous avons choisi pour notre travail de subdiviser au niveau technique la notion d'interopérabilité en deux phases l'interopérabilité statique et l'interopérabilité dynamique. L'interopérabilité statique concerne un transfert des contenus entre les plates-formes de manière quasi automatique tandis que l'interopérabilité dynamique traite des échanges de données (forum, messages, etc.) entre services des différentes plates-formes. Nous proposons une méthodologie de conception d'un outil informatique permettant les échanges des contenus entre les deux plates-formes Moodle et Dokeos. Ensuite nous passons à une mise en œuvre de l'interopérabilité statique en choisissant deux cours pilotes. Le premier cours est un cours générique obtenu sur le site officiel de

---

<sup>5</sup> Cet objectif est développé aux chapitres 6 et 7.



la plate-forme Moodle et le deuxième a été obtenu grâce à la collaboration de l'Université de Mons UMONS et du CTE<sup>6</sup> de l'Université Libre de Bruxelles.

## 7. Interopérabilité dynamique

Afin de permettre un suivi à distance des étudiants par un enseignant étranger, il nous a paru impérieux de proposer aussi une méthodologie de conception d'un outil informatique capable de permettre un échange des données entre différents services des plates-formes. Devant les multiples pistes de solutions qui se présentent pour de telles problématiques, nous avons choisi l'approche par la conception des services Web<sup>7</sup>.

## 8. Conclusions et perspectives

Respectant la tradition de la rédaction scientifique, nous atterrirons à la fin de ce travail avec une conclusion qui retracera de manière synoptique tout le parcours de la thèse, en revenant sur ce qui a pu être réalisé durant nos travaux, les objectifs atteints par le travail, les réalisations et l'apport du présent travail.

# 1.3 Les enjeux de l'interopérabilité des plates-formes e-Learning

Dans le domaine informatique de nos jours, plusieurs études et travaux portent sur la réalisation des standards, normes et labels. Ces efforts de normalisation et de standardisation aux niveaux mondial (ISO, IMS, LTSC, IEEE) et européen (ARIADNE, CEN, AFNOR)<sup>8</sup> visent à répondre aux questions relatives à l'interopérabilité des technologies et des contenus entre les systèmes informatiques. Les principaux enjeux de la standardisation et de la normalisation sont de rendre interopérables les technologies

<sup>6</sup> Centre des Technologies au Service de l'Enseignement de l'ULB.

<sup>7</sup> Technologies permettant une interopérabilité entre applications informatiques totalement différentes (voir chapitre 7)

<sup>8</sup> Tous ces acronymes des organismes de la normalisation sont définis et développés au point 3.1.2, du chapitre 3

entre elles, afin de faciliter l'usage des ressources (éducatives dans notre cas) quels que soient la plate-forme ou l'environnement technologique utilisés.

Il faut noter que l'interopérabilité est basée sur l'interconnectivité des plates-formes, qui est elle-même régie par le respect des normes et des standards informatiques de communication, d'échanges et de définitions des concepts. Les standards de communication consistent en des spécifications techniques qui se traduisent par des définitions et des règles d'ingénierie, de façon à assurer que les produits, les traitements et les services remplissent bien leurs rôles. C'est pourquoi, sachant que les plates-formes Dokeos et Moodle ont la vocation d'être conformes à la norme SCORM qui est assez générique et a le vent en poupe dans l'univers de l'e-learning. Ce travail se penchera sur SCORM pour déterminer et appréhender tous ses concepts établis. En effet, il est évident que l'application ULLOXA devra elle-même y être compatible pour arriver à interfacer Moodle et Dokeos. Nous y consacrerons tout un chapitre.

Il est donc important à ce stade de noter qu'assurer une interopérabilité entre des systèmes informatiques revient à concevoir une application capable de permettre à ces systèmes d'échanger à la fois des données et des services. Cette entreprise comprend donc à notre entendement deux aspects importants et déterminants : elle présente un aspect statique comme, par exemple, la compatibilité des types de données et un aspect dynamique comme la compatibilité des procédures ou services (Degoulet, et al., 1997).

Dans un premier temps, il sera question de concevoir une capacité d'échange syntaxique des contenus de chacune des plates-formes qui sera suivie ensuite d'une interopérabilité sémantique. Pour garantir que les différents concepts de contenus venus de contextes différents inhérents à la philosophie et l'implémentation des plates-formes sont transférés fidèlement de l'un vers l'autre système et vice versa. Nous ne perdrons pas de vue que le travail sémantique devra être conforme à une certaine ontologie de



SCORM, c'est-à-dire que les contenus des deux plates-formes passeront par une transcription en SCORM. Ceci va bien entendu mener à une interface commune de référence qui va conserver le sens des données échangées et qui va aider, en plus des échanges syntaxiques, à effectuer aussi la conformité sémantique des contenus ainsi que des services échangés. Tout cela pour s'assurer que les plates-formes en communication aient une compréhension commune de la signification des données et des services qu'elles échangent.

Nous observerons de manière particulière un axe de travail qui permettra que les données, ainsi que les descripteurs de ces données, gardent la même signification de part et d'autre de notre application, c'est-à-dire aussi bien dans Moodle que dans Dokeos, pour que leurs finalités demeurent respectées.

## 1.4 Développement d'ULLOXA et choix technologiques.

L'application d'interfaçage entre Dokeos et Moodle de notre travail va s'appeler comme indiqué plus haut *University of Lubumbashi Learning Objects Exchange Application*, en abrégé ULLOXA. Le design et la vocation de l'application ULLOXA nous pousseront à entreprendre une démarche de choix technologique approprié partant des objectifs que l'on se fixe pour notre application sachant bien quelles sont les attentes que nous avons par rapport à l'outil.

Brièvement, et d'un point de vue technique, ULLOXA devrait arriver à réaliser les fonctions suivantes :

1. Extraire du contenu d'une plate-forme A.
2. Traduire ce contenu et le présenter selon la norme SCORM.
3. Retranscrire le contenu SCORM selon les règles de la plate-forme B.
4. Injecter ensuite ce contenu dans la plate-forme B.

5. Assurer un flux d'échanges bidirectionnel, c'est-à-dire être capable d'assurer dans les deux sens une communication permanente des services entre les plates-formes.
6. Relever les mises à jour des contenus et des services de part et d'autre et les mettre à jour dans les deux plates-formes.

Ceci conduit à penser que l'application devrait avoir, au-delà de l'interface, pour les deux plates-formes, des modules d'extraction de contenus, de traduction ou mise en forme des contenus aussi bien côté Dokeos, que côté Moodle. D'autre part, l'usage des deux plates-formes nous enseigne que leurs contenus se présentent sous la forme d'un fichier zip contenant les ressources d'apprentissage, les matériaux, les outils, les liens liés à l'unité d'apprentissage. Un manifeste XML décrivant la méthode d'enseignement utilisée et tous les éléments physiques, appelés aussi ressources. Ils constituent soit un cours soit un module de cours, ou encore une unité d'apprentissage quelconque et pointent sur des liens et des services éventuels faisant partie de l'apprentissage d'une matière. La figure 1.1 Schématise le fonctionnement de l'application.

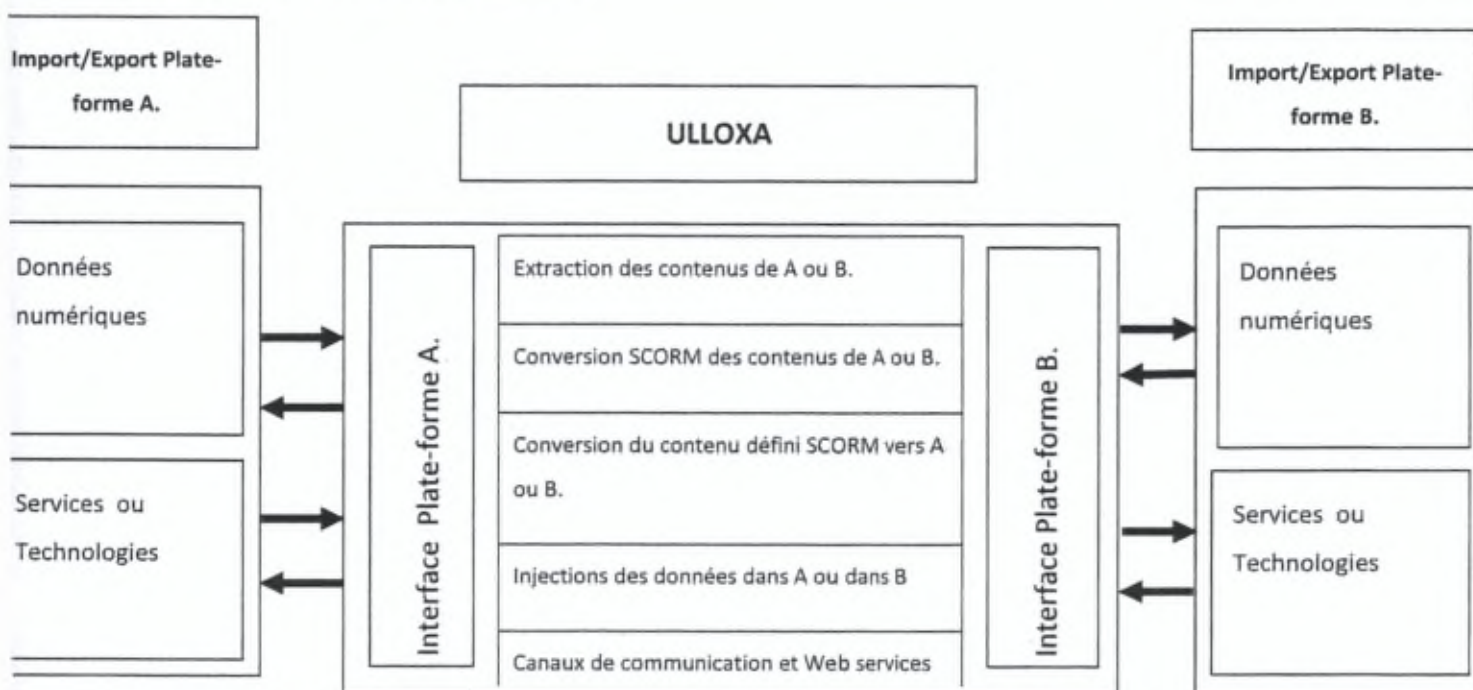


Figure n°1.1 : Schéma du rôle de l'application ULLOXA.



Nous ne manquerons pas de décortiquer la démarche de conception d'ULLOXA suivant les étapes de structuration définies comme suit :

1. La description des objectifs techniques à atteindre par la création de l'application ainsi que les stratégies de l'application.
2. La caractérisation de la nature des contenus des cours ou des enseignements et des services qui feront l'objet des échanges entre les plates-formes à travers les différents modules de l'application.
3. Au niveau computationnel, la description de la décomposition fonctionnelle de l'application et les interactions entre les interfaces des différents objets de l'application.
4. La description des moyens techniques ou d'ingénierie mis en œuvre pour que les objets de l'application interagissent.
5. La définition des technologies logicielles et matérielles à utiliser.

L'utilisation des métadonnées pour décrire des ressources numériques d'apprentissage dans les plates-formes nous pousse à trouver une technologie pouvant manipuler à la fois du XML de manière puissante et d'implémenter des modules capables de remplir les rôles d'extraction, de traduction, d'injection et d'interfaçage. Notre choix a porté tout de suite sur la technologie DOM XML et le langage Python avec la bibliothèque PyXML. DOM se définit sommairement comme : Document Object Model. Il s'agit d'une API permettant la manipulation de fichiers XML. Grâce à cette API, on peut lire un fichier XML mais également ajouter de nouveaux éléments à ce fichier ou le transformer. D'autre part, avec l'API DOM de Python, tout est élément ou nœud, même le texte. C'est donc cette technologie qui servira pour le développement des prototypes du futur outil ULLOXA.

## 1.5 Ce que nous visons comme contribution

En Afrique subsaharienne, de nombreuses études démontrent que plusieurs projets d'apprentissage en ligne ont souvent échoué pour de multiples raisons d'inadaptation et d'inadéquation de plusieurs paramètres entrant en jeu dans la conception de tels projets.

Quelques uns auront retenu notre attention pour apporter une contribution dans la recherche des solutions. Il s'agit du format ou scénario des cours conçus à la base dans un contexte d'enseignement-apprentissage du nord qui est généralement totalement différent de celui du sud, et cela à plusieurs points de vue. Une révision de leur format est parfois nécessaire pour arriver à un résultat et éviter la lassitude et les abandons de la part des enseignants et des étudiants que l'on constate fréquemment dans ce type de projet en Afrique. Un autre élément qui accentue les difficultés de l'e-learning à atteindre son objectif, de fournir au sud des enseignements mis à jour et de haut niveau du nord, c'est la difficulté du suivi de la part d'un enseignant du nord de ses étudiants du sud de manière permanente et continue ainsi que de l'évolution des enseignements qu'il dispense au sud. Cela est dû en grande partie à la différence de plates-formes utilisées par l'université de l'enseignant du Nord d'une part, et celle de l'université de ses étudiants du sud d'autre part. Il est clair que, pour encourager les deux parties, chacune devrait œuvrer avec son outil et pouvoir échanger avec l'autre dans le cadre des enseignements pour éviter le risque d'obliger chacune des parties d'utiliser autant de plates-formes qu'il n'y a des cours venus de plusieurs universités. Pire, au nord, il existe des universités possédant plus d'une plate-forme d'apprentissage en ligne. La contribution de notre travail se situe à ce niveau : il s'agit d'une part, d'imaginer le format de cours adapté au contexte local lors de l'usage d'une plate-forme car le contexte pour tenir compte des particularités liées à la discipline concernée par la formation, ainsi que des contraintes organisationnelles, spatiales, temporelles, techniques et économiques dans lesquelles s'inscrivent les activités (Pernin, et al., 2007). Pour cela, nous analyserons les usages et les comportements émergents d'un enseignement par les biais des moyens informatiques pour détecter les fonctionnalités pertinentes au bon apprentissage et les calibrer. D'autre part, nous cherchons à concevoir un outil qui fera l'interfaçage entre des plates-formes hétérogènes pour permettre les échanges des données et la communication afin que l'enseignant du nord arrive à suivre de manière régulière, synchronisée, et automatique ses étudiants du sud à partir de sa propre plate-forme d'enseignement. L'objectif est de ne pas perdre la motivation car l'apprentissage permanent de l'usage d'un outil informatique nouveau peut engendrer effectivement un



découragement de la part du futur utilisateur, et il est donc mieux de laisser chacun travailler avec l'outil auquel il est le plus habitué.

## CHAPITRE 2 : PRESENTATION DU SUJET ET CONTEXTE DES TRAVAUX

Ce chapitre a pour objectif de donner une image sommaire du milieu sur lequel notre travail s'attèle puisque c'est la première fois qu'une telle étude se fait en République Démocratique du Congo. Nous y donnerons toutes les caractéristiques « *socio-informatiques* » de la ville de Lubumbashi en vue de déterminer sa capacité à disponibiliser des moyens informatiques acceptables pour permettre la réussite d'un projet d'apprentissage en ligne. Nous donnerons aussi le contexte scientifique du travail qui répertorie toutes les disciplines scientifiques nécessaires à son aboutissement. Nous développerons enfin la problématique traitée ainsi que les hypothèses et la méthodologie que nous comptons mettre en œuvre.

### 2.1 Pertinence du sujet.

Dans cette section nous décrirons le contexte de nos travaux en deux volets. D'abord le volet scientifique et ensuite le volet en rapport avec le milieu d'application de cette étude. Car c'est la première fois que de tels travaux se mènent dans la ville de Lubumbashi en particulier et pour toute la République Démocratique du Congo en général. Ce deuxième volet donne toute son originalité à notre démarche.

#### 2.1.1 Contexte scientifique des travaux

Ce travail se situe dans le champ de l'apprentissage collaboratif en ligne ou e-learning. Nous nous intéressons à la construction d'applications, d'outils et d'artefacts informatiques capables de soutenir et d'assister les acteurs engagés dans cette forme d'enseignement. C'est dire que nous nous situons à l'intersection des sciences humaines, en ce sens qu'il faut identifier, comprendre, formaliser et modéliser les phénomènes observés, et des sciences de l'ingénieur, car nous devons puiser dans l'informatique des concepts et des paradigmes susceptibles d'aider à la résolution des problèmes identifiés.

Cette thématique se situe de manière un peu plus précise dans le domaine de l'apprentissage en ligne, qui regroupe tous les acteurs impliqués dans une pratique



d'enseignement passant par le biais d'outils informatiques. L'e-learning peut se définir comme suit : *le e-learning est l'utilisation de technologies informatiques et télématiques afin de transmettre ou de suivre à distance une formation complète ou une partie de celle-ci.* Et Valayer renchérit en prétendant que l'e-learning se décline selon différents scénarii compris entre celui de la transmission de contenu asynchrone, lié ou non à une formation présentielle, et celui d'une transmission de contenu et d'une interaction effectuées à distance entre le formateur et les apprenants (Valayer, 2004).

Notre démarche dans ce travail consiste à mener des expérimentations de situations d'apprentissage collaboratif à distance pour faire ressortir les usages et les comportements émergents au sein de l'application de ce mode d'apprentissage à l'Université de Lubumbashi afin de dégager le scénario ou le format d'apprentissage imposé par le contexte local du milieu d'apprentissage. Nous calibrerons ensuite ce format à la plate-forme adoptée localement pour construire en aval une application permettant des échanges automatiques entre les plates-formes des universités en jeu ici tout en donnant à la sortie au niveau de la plate-forme locale une unité d'apprentissage ou un contenu calibré au format de la plate-forme d'accueil. On se servira donc à ce niveau des concepts et problématiques de l'informatique capables de nous aider à atteindre les objectifs ciblés. Ce travail est donc appelé à intégrer :

- Des théories issues des sciences humaines et sociales comme la théorie de l'apprentissage collaboratif afin d'observer les apprenants en plein apprentissage.
- Des concepts développés à travers les différents travaux de normalisation et de standardisation dans le domaine plus spécifiques du e-learning.
- Des concepts et paradigmes informatiques comme l'analyse, la conception, la spécification et le développement des applications (modélisation, programmation, etc.).

## 2.1.2 Contexte du milieu de travail

Le contexte du milieu de travail étant un cas singulier, il mérite que l'on puisse s'y attarder car sa configuration contribue à l'originalité de ce travail. Nous sommes donc en Afrique subsaharienne, en R.D Congo, dans la ville de Lubumbashi où nous devons évaluer les chances actuelles et futures de la réussite d'un projet d'apprentissage en ligne. L'Université de Lubumbashi compte environ 20000 étudiants. Pour décortiquer le contexte informatique de la ville de Lubumbashi, nous allons nous appuyer sur les résultats d'une enquête initiée pour cela. Retenons que d'après un sondage en avril 2010 auprès d'un échantillon d'étudiants des facultés de Droit, Médecine, Polytechnique et de l'Ecole Supérieure des Ingénieurs Industriels, à l'Université de Lubumbashi moins de 2% d'étudiants possèdent un ordinateur soit portable soit fixe, et que la proportion des étudiants qui ont à leur domicile une connexion Internet est très insignifiante. Cependant 87% d'entre eux connaissent Internet et l'ont déjà utilisé pour travailler, 64% possèdent une adresse courriel parmi eux, près de 60% ont acquis ou envisage avoir un compte sur les réseaux sociaux, notamment « Facebook ». Si personne ne possède un ordinateur privé chez soi, c'est donc le milieu, soit la ville, qui lui offre cette opportunité, d'où la nécessité de dégager les potentialités et les caractéristiques de ce que nous appelons ici « *le profil informatique* » de la ville de Lubumbashi.

D'ores et déjà, nous pouvons prétendre qu'avec l'expansion spectaculaire de l'informatique au Congo, s'appuyant sur le fait que quasi tous les étudiants de l'Université de Lubumbashi toutes facultés confondues ont un cours d'informatique de base et qu'une grande proportion (80%) des étudiants ont déjà eu affaire à Internet, nous pouvons aisément prédire un bon départ pour un projet de session de formation en ligne à l'Université de Lubumbashi. Ces conditions vont en s'améliorant et sont nettement différentes des conditions de 2006 lors du début de nos travaux. En quatre ans tout est allé très vite et nous osons croire, comme nous le confirment les résultats de l'enquête menée sur ces trois années, que la tendance ne s'infléchira pas de si tôt.



## 2.2. Problématique

L'apprentissage en ligne à ses débuts a nourri beaucoup d'espoir dans la communauté de l'enseignement en général, mais il a vite posé de nombreux problèmes spécifiques qui pousse la communauté e-learning mondiale à imaginer des solutions en s'attaquant à chacun des problèmes posés. Pour ce qui est de l'Afrique, on dénombre de nombreux projets d'e-learning. Spécifiquement en R.D Congo, aux débuts des années 2000, quelques projets ont été initiés, mais ils ont vite fait l'objet d'abandons pour plusieurs raisons relatives à l'usage des TIC<sup>9</sup> en enseignement. Pour notre part, parmi ces raisons, nous nous intéressons particulièrement aux nombreuses lacunes dans l'accompagnement des apprenants, à la difficulté pour l'apprenant évoluant dans un contexte purement africain de s'approprier certains paradigmes d'enseignement spécifiques à des situations du nord, à la difficulté pour l'enseignant du nord d'assurer un suivi permanent à distance de ses étudiants du Congo. Ces lacunes découlent d'un manque d'outils informatiques appropriés pour assurer de manière automatique le suivi du cours et de l'évolution des étudiants, et éviter ainsi l'un des facteurs de découragement les plus constatés en e-learning. Le fait de devoir, pour chaque projet, s'approprier d'un nouvel outil informatique qui pose un biais pour l'acquisition effective du savoir qui est finalement le but suprême de l'enseignement.

Nous noterons donc pour ce travail deux problématiques :

1. Le manque d'adaptation des formats des contenus des cours au contexte local de l'environnement dans lequel se déroule l'enseignement.
2. L'absence d'un outil informatique capable de permettre des échanges automatiques et un suivi à distance permanent des étudiants et du déroulement global du cours.

Dès lors il nous incombe, pour assurer un succès à un cours en ligne et lui donner toutes ses chances, d'arriver, à terme, de mettre au point des stratégies capables d'endiguer la difficulté afin de diminuer le taux d'abandon inhérent à l'apprentissage en ligne. Pour cela il faut répondre aux questions suivantes : quels sont les indicateurs et les outils à mettre en

---

<sup>9</sup> TIC: Technologies de l'Information et de la Communication

place et à surveiller qui reflètent le déroulement réel d'une session de formation en ligne dans le contexte local ? Lesquels sont déterminants dans le scénario d'enseignement et le format adéquat du cours ? Quels dispositifs informatiques imaginer pour permettre et faciliter un suivi à distance quasi automatique et permanent des étudiants et de l'évolution de l'unité d'apprentissage par l'enseignant ? Quelles stratégies mettre en place pour s'assurer de l'adaptation au contexte du format du cours et de la maintenance de l'outil informatique afin de garantir le bon déroulement d'une formation et l'atteinte de ses objectifs ?

Ces problèmes sont évidemment inhérents aux problèmes généraux de l'e-learning. Pour notre part, nous nous baserons sur les concepts d'interopérabilité principalement et nous emprunterons quelques concepts des sciences humaines afin d'analyser l'usage d'une plate-forme en situation réelle d'apprentissage.

En définitive, la question centrale de cette thèse concerne la proposition des méthodologie de conception ainsi que la mise au point d'outils informatiques pouvant faciliter le suivi à distance automatique et permanent de l'apprentissage tout aussi bien en assurant les échanges des contenus que celui des services entre la plate-forme de l'enseignant et celle de ses étudiants dans le contexte local d'e-learning. Il devient important de connaître le format de sortie ou d'export des contenus et des services des deux plates-formes d'une part, et de connaître aussi le formalisme SCORM d'autre part, pour assurer le passage du paradigme d'une des plates-formes à celui de SCORM qui jouera ensuite l'interface avec l'autre plate-forme.

Il est aussi important de pouvoir observer et analyser une situation réelle d'apprentissage pour justement dégager à partir des comportements émergents dans l'usage des outils, le format du cours ou le scénario reflétant le plus possible le contexte local d'enseignement. On s'intéressera également dans une certaine mesure aux interactions constatées entre les étudiants suivant le même cours, sans traiter en



profondeur la question du type d'influence que ces interactions ont sur l'apprentissage et le déroulement global des enseignements.

Le souhait ici est que des systèmes informatiques soient mis à la disposition des utilisateurs des plates-formes d'apprentissage en ligne, pour leur éviter des difficultés de lassitude dues à la prise en charge et à l'appropriation de nouveaux outils informatiques qui sont souvent porteurs de plus de complexité. Ils deviennent par la même occasion des difficultés supplémentaires dans tous les domaines où l'outil informatique est devenu indispensable en général, et dans l'apprentissage en ligne en particulier. Il est dorénavant important de s'intéresser de près à l'activité de l'utilisateur et de soutenir celle-ci en proposant des systèmes informatiques disposant d'outils et d'artefacts appropriés. Cette préoccupation devrait interpeller les informaticiens et les engager à identifier des outils, des concepts et des paradigmes capables de les aider à analyser, concevoir, spécifier et développer ces systèmes informatiques (Mbala, 2003).

L'e-learning est une contribution évidente à la nouvelle vision de l'informatique intelligente qui fait de l'ordinateur non pas un outil servile et prêt à obéir aux instructions qui lui sont données par l'utilisateur, mais un partenaire de ce dernier qui l'assiste dans sa démarche de recherche des solutions (Negroponte, 1997; Leroux, 1995).

L'e-learning comme l'ensemble de tous les domaines de l'informatique, voit croître de nos jours ce qu'on appelle une démocratisation massive. L'informatique n'est plus l'apanage seulement d'une certaine catégorie d'érudits ou d'élites appelés en général informaticiens. L'ordinateur est dorénavant utilisé par des non informaticiens si bien que le programmeur n'est plus seulement appelé à mettre en œuvre des capacités d'analyse, de conception, de codage des logiciels mais aussi à appréhender toute la complexité à cerner les vrais problèmes et à les modéliser. Dès lors que des non-informaticiens sont appelés à manipuler l'ordinateur, il est établi que nombre de problèmes deviennent tellement complexes et évolutifs qu'il est quasi impossible d'envisager une analyse et une modélisation complètes débouchant sur la mise en place d'applications appropriées (Ferber, 1995).

## 2.3. Objectifs de la thèse.

Les objectifs poursuivis dans nos travaux sont de répondre à toutes les questions soulevées plus haut lors de présentation de la problématique. Cela revient premièrement à mener une formation d'e-learning à titre d'expérimentation sur le terrain, afin qu'en tant qu'acteurs et observateurs on se plonge dans la réalité du déroulement de la formation et en sorte les caractéristiques principales permettant une connaissance parfaite de tous les paramètres intervenant dans une telle formation en contexte local. Il faut aussi construire un scénario adéquat d'enseignement pour un succès de la session de formation. Enfin il faut concevoir et construire des outils informatiques, à la disposition principalement de l'enseignant, capables d'interfacer les deux plates-formes en présence, d'assurer et de faciliter le suivi à distance des étudiants. Nos travaux constituent donc une contribution aux efforts de toute la communauté e-learning en matière d'interopérabilité, de normalisation et de définition de standards pour l'apprentissage en ligne au niveau mondial par les consortium ISO, IMS, LTSC, IEEE, et au niveau européen par ARIADNE, CEN et AFNOR.

Pour bien cerner les choses :

Dans un premier temps nous allons effectuer une expérimentation en grandeur réelle d'une formation en ligne au sein de l'Université de Lubumbashi pour, comme nous l'avons signifié plus haut, dégager les faits saillants du déroulement d'une telle session. Notre souci est bien entendu de pouvoir capter les comportements et les faits émergents, surtout dans l'utilisation des outils de la plate-forme, pour en dégager un scénario de cours qui soit à la fois fidèle au déroulement réel de la formation et à l'usage réel de la plate-forme, et ceci dans les conditions qui permettent le plus un aboutissement positif de la session de formation et donc un succès de l'usage de l'e-learning. Cette expérimentation a aussi pour but de nous permettre de développer localement les stratégies qui conduisent au succès d'une formation e-learning dans les conditions et le contexte locaux d'enseignement.

Dans un deuxième temps, étant en présence pour nos travaux des plates-formes Moodle et Dokeos, nous devons décortiquer, analyser et caractériser les formats des



contenus d'une unité d'apprentissage (chapitre, leçon, module, cours, cursus, etc.) de chacune des plates-formes, comprendre leur constitution et leur empaquetage à l'export afin de mieux les manipuler et les formater conformément à une norme tampon plus générique pouvant s'interfacer avec chacune des deux plates-formes. Notons que la quasi universalité de la norme tampon SCORM a pour objectif non avoué de servir toujours d'interface pour d'autres plates-formes d'e-learning qui pourraient lui être conformes, sachant bien que la conformité des contenus des plates-formes à telle ou telle autre norme est discutable.

Dans un troisième temps, nous concevrons et construirons une méthodologie de mise au point d'une application informatique qui aura pour mission d'assurer une circulation automatique des contenus et des services d'une plate-forme à l'autre dans un flux bidirectionnel, c'est-à-dire valable dans les deux sens. Nous comptons donc arriver à une application qui pourra d'abord extraire les données d'une plate-forme, les formater suivant une norme générique choisie, les empaqueter et enfin les injecter dans l'autre. En même temps, elle devra assurer des canaux de communication entre les différents services inhérents à une session de formation et implémentés dans chacune des plates-formes pour une communication permanente mise à jour.

Dans un quatrième temps enfin, fort de l'expérimentation grandeur nature et de l'application informatique ainsi créée, en tant qu'informaticien, notre objectif sera de valoriser la contribution de notre démarche technique, des concepts et paradigmes informatiques choisis, à l'e-learning et à l'ouverture de certaines pistes de recherche dont les résultats ne viendront qu'enrichir l'édification de l'interopérabilité effective des plates-formes de l'e-learning.

## **2.4 De l'e-learning a Lubumbashi**

Dans cette section nous tenterons de présenter un peu plus le milieu sur lequel nos travaux ont porté. Nous estimons à cette étape qu'il convient de se représenter le potentiel informatique de la ville de Lubumbashi pour savoir si un étudiant de l'Université de

Lubumbashi dans sa ville a des accès suffisants à Internet. Quelle est la disponibilité des ordinateurs ? Quel type de matériel l'étudiant a-t-il à sa disposition ? Quelles sont les caractéristiques des connexions Internet disponibles ? Peut-on alors de manière sérieuse envisager pour l'étudiant de l'UNILU<sup>10</sup> une session de formation en ligne se basant sur l'infrastructure disponible dans sa ville ?

Pour arriver à cette fin nous avons diligenté une enquête de 2007 à 2010. Cela nous a permis d'avoir un synopsis de l'infrastructure informatique offert par la ville de Lubumbashi. Nous avons également, durant la même période, effectué une expérimentation en grandeur nature de l'e-learning à l'Université de Lubumbashi. Comme déjà signalé plus haut, ces expérimentations avaient pour but principal d'identifier les comportements émergents lors du déroulement des enseignements en ligne pour déterminer quels sont les outils de la plate-forme qui s'avéraient indispensables ou étaient les plus utilisés par les étudiants au cours des enseignements. Ceci devrait nous permettre de prévoir quels sont les outils majeurs à intégrer dans la forme du scénario des cours importés sur notre plate-forme en amont du développement de l'outil d'import-export. Mais nous y reviendrons avec force détails dans le chapitre prochain.

### 2.4.1 Profil informatique ou disponibilité de l'outil informatique dans la ville de Lubumbashi.

Cette enquête, menée avec la participation des étudiants de l'Université de Lubumbashi (UNILU) ainsi que ceux de l'Université Protestante de Lubumbashi (UPL), s'est opérée en divisant la ville de Lubumbashi en plusieurs sites, bien délimités par des artères principales connues et quadrillés ensuite par les patrouilles d'enquêtes constituées de neuf personnes chacune afin de n'omettre aucune entité informatique. Notons que les entités informatiques ont été identifiées en partant de la définition suivante : est appelé entité informatique tout lieu où des ordinateurs connectés ou pas à Internet sont mis à la disposition du public, soit pour des formations ou des services informatiques, soit pour la

---

<sup>10</sup> UNILU: Université de Lubumbashi.



navigation sur Internet et la recherche de l'information via le Web. Et nous catégorisons ces entités en quatre groupes à savoir les cybercafés, les bureautiques, les centres des formations et les fournisseurs d'accès Internet<sup>11</sup>:

1. Cybercafé : café disposant d'ordinateurs où l'on peut se connecter sur le réseau Internet. Un cybercafé est donc un lieu public dans lequel on propose aux personnes d'accéder à Internet. Certains auteurs prétendent d'ailleurs que CAFE est l'acronyme de *Communication Access For Everyone* (Accès à la communication pour tous).

2. Bureautique : l'ensemble des techniques informatiques désignant la mécanisation et l'automatisation des travaux de bureau (Pujolle, 2008). C'est, au sens populaire, un lieu public où on organise la conception, la création, la mise en forme et la mise au point des documents par des moyens informatiques (par exemple, les Curriculum vitae, les demande d'emploi, etc.).

3° Centre de formation : un établissement de formation informatique où se dispensent les enseignements autour des nouvelles technologies de l'information et de la communication (NTIC). Généralement au Congo, ces genres d'institutions insistent beaucoup plus sur des matières portant sur les outils de bureautique tels que le traitement de texte, les tableurs, la présentation assistée par ordinateur, etc. A côté de cela, on dénombre aussi certains centres dits spécialisés où se donnent des cours informatiques plus poussés et dans des branches et filières bien pointues telles que les réseaux, la programmation et les bases de données.

4° Fournisseur d'accès Internet (FAI) : un organisme ou une entreprise qui offre la connexion au réseau Internet à l'échelle mondiale. Les anglophones l'appelle aussi *Internet Service Provider (ISP)* ou encore *Internet Access Provider*. On distingue généralement les différents FAI par les types et la qualité des services qu'ils peuvent offrir (débits, bande passante, supports techniques, prix, etc.).

---

<sup>11</sup> Parfois ces trois catégories se retrouvent dans une même entité.

Pour étayer le profil informatique de la ville de Lubumbashi pendant ces trois ans, nous l'avons subdivisée en 10 sites partant du campus universitaire de la Kasapa (sud-ouest) jusqu'au site le plus éloigné, la commune de la Ruashi (nord). Il est aussi important de noter d'ores et déjà que nous ferons nos analyses partant du postulat que l'étudiant de l'Université de Lubumbashi ne possède pas d'ordinateurs personnels vu le faible taux (soit moins de 4%) d'étudiants en possédant. Pour chaque site, nous avons fourni un tableau détaillé qui représente par catégories les cybercafés, les bureautiques, ainsi que les centres de formation. Ces tableaux regroupent les informations suivantes pour chaque catégorie d'entité informatique : le site, le type d'entité informatique, le nom de l'entité, l'adresse physique ainsi que l'adresse e-mail, le nombre d'ordinateurs, le type de processeur, Le système d'exploitation utilisé, les nombres d'onduleurs, scanners, imprimantes, photocopieuses, graveurs, le Fournisseur d'Accès Internet, l'antivirus utilisé et les horaires d'ouverture. Pour ce qui concerne les fournisseurs d'accès Internet, étant les points qui constituent le nœud d'entrée de l'Internet dans la ville de Lubumbashi, nous avons choisi de les recenser et les regrouper dans un même tableau qui détaille leur nom, le débit offert, le prix de la connexion, le mode de connexion, les services disponibles, ainsi que l'adresse physique de ces FAI. Chaque site est présenté en détails en annexe du travail, un seul tableau (tableau 1.1) est retenu ici pour nous donner une idée de la situation du campus universitaire de la Kasapa.

## 2.4.2 Résultats détaillés d'enquêtes 2007-2009.

Les résultats détaillés des enquêtes sont consignés dans des tableaux que nous donnerons dans les annexes du présent travail. Comme nous l'avons déjà dit, ces enquêtes ont non seulement donné l'état des lieux du profil informatique de la ville de Lubumbashi mais elles ont aussi aidé à dresser un bilan global de l'évolution en moyens informatiques et Internet de la ville de Lubumbashi afin d'interpréter la tendance de cette évolution qui peut aller soit crescendo soit decrescendo et qui constitue un élément majeur dans la détermination de la réussite d'un projet e-learning en Afrique subsaharienne. Etant étalés sur trois années académiques, les résultats reprennent plusieurs tableaux et courbes



décrivant en détails les observations faites sur terrain. Nous représentons ici juste les conclusions de quelques analyses au tableau 1.2.

## 2.4.3 Analyse générale

### 2.4.3.1 Fonctionnement et nombre des ordinateurs

Il ressort des enquêtes et investigations menées dans la ville de Lubumbashi entre 2007 et 2010 que les cybercafés de cette ville utilisent en moyenne 10 ordinateurs dans des locaux excédant rarement 10 m<sup>2</sup>. Les heures d'ouverture et de fonctionnement sont de 12 heures par jour en moyenne. Les bureaux à leur tour sont encore plus exigües dans leur dimension et elles ont en moyenne six ordinateurs, pour des heures d'ouvertures moyennes plus courtes (11h30). Les centres de formation sont en plus petit nombre, soit 17 centres pour toute la ville (environ le quart du nombre des autres entités), mais quand ils existent, leurs locaux sont souvent assez spacieux et ils possèdent en général plus d'ordinateurs (en moyenne 30) que la moyenne des deux entités précédentes. Ils sont aussi dotés d'un personnel généralement plus qualifié. Notons aussi que le parc des ordinateurs de la ville de Lubumbashi, généralement vétuste et obtenu en seconde main est constitué en majeure partie de Pentium 4

Tableau n° 1.1 : Extrait du profil informatique du site de la Kasapa (Année 2010)(tiré de notre enquête)

| Site   | Type      | Nom                            | Adresse & Mail         | Nombre des PC              | OS & Processeurs           | Onduleurs & Scanner                                 | Imprimante & Photocopieuse                                       | Graveur    | Antivirus          | FAI      | Horaire   |
|--------|-----------|--------------------------------|------------------------|----------------------------|----------------------------|---|--|------------|--------------------|----------|-----------|
| CAMPUS | CYBERCAFE | SAINT ESPRIT                   | Paroisse Saint Esprit  | 8 PC<br>HDD : 80           | Win XP SP3<br>P3,P4, PM    | Pas d'onduleur<br>1 Scanner<br>CANNON               | 1 imprimante HP<br>LASER<br>1 Photocopieuse<br>CANNON            | 2 Graveurs | AVIRA sans licence | Comax    | h00-23h00 |
|        |           | CYBER DINOKONDE                | Home 9 UNILU           | 9 PC<br>HDD : 120 et 160Gb | Win XP SP2<br>P3, PM       | Pas d'onduleur<br>Pas de scanner                    | 1 imprimante HP<br>Pas de photocopieuse                          | 2 Graveurs | AVIRA sans licence | UNILU    | h00-21h00 |
|        |           | CYBER CELLULE DE CONTACT UNILU | Building Administratif | 18 PC<br>HDD : 45 Gb       | Win XP SP2<br>P4, PM       | 4 onduleurs<br>Ellipse & A°C<br>1 scanner<br>CANNON | 2 imprimantes LASER<br>2 photocopieuses<br>CANNON                | 3 Graveurs | AVIRA sans licence | UNILU    | h00-20h00 |
|        |           | SALVANET SERVICE               | Home 5                 | 8 PC<br>HDD : 45 Gb        | Win XP SP3<br>Celeron      | Pas d'onduleur<br>1 scanner<br>BENQ                 | 1 imprimante JET 1010<br>1 photocopieuse<br>CANNON               | 2 Graveurs | AVIRA sans licence | Microcom | h00-22h00 |
|        |           | ExcelSYS IT                    | Bloc K                 | 6 PC<br>HDD : 60 Gb        | Win XP SP3<br>Intel P3, P4 | Pas d'onduleur<br>2 scanner HP et<br>CANNON         | 2 imprimantes HP &<br>CANNON<br>3 Photocopieuses<br>HP et CANNON | 2 Graveurs | AVIRA sans licence | Comax    | h00-21h00 |
|        |           | CYBER LA NOUVELLE              | Home 5                 | 6 PC<br>HDD : 30 & 40 Gb   | Win XP SP3<br>Intel P3,P4  | Pas d'onduleur<br>1 scanner                         | 2 imprimantes Deskjet<br>F4180 HP<br>Pas de Photocopieuses       | 1 Graveur  | AVIRA sans licence | Vodanet  | h00-21h00 |
|        |           | CYBER SHUM                     | Faculté des Lettres    | 17 PC<br>HDD : 30 & 40 Gb  | Win XP SP3<br>Intel P3, P4 | Pas d'onduleur<br>1 scanner                         | 2 imprimantes Deskjet<br>2180 HP<br>Pas de photocopieuses        | 1 Graveur  | AVIRA sans licence | Vodanet  | h00-19h00 |



### 2.4.3.2 Taux de fréquentation des entités informatiques

Les cybercafés, surtout pour ceux qui sont encore viables, reçoivent en moyenne 100 personnes par jour. Les bureautiques arrivent en deuxième position avec un taux de fréquentation de près de 66 personnes par jour, mais cela est très variable selon les sites exploités, la période de l'année, l'emplacement géographique ainsi que le type de public. Les centres de formations viennent en dernière position avec en moyenne 28 personnes les fréquentant par jour. Notons qu'en 2007, la fréquentation des centres de formation ne représentait que 10% de la fréquentation des cybercafés.

### 2.4.3.3 Intérêt de la fréquentation

Dans les cybercafés, pour 80% d'internautes interrogés, c'est la messagerie qui de 2007 à mi-2009, est le principal usage. Ensuite, un autre usage talonne la messagerie : les réseaux sociaux du Web avec en tête Facebook, qui représente en octobre 2010 près de 50% des activités des internautes contre 45% pour la messagerie. Les calculs statistiques montrent qu'à Lubumbashi un internaute passe en moyenne 28 min dans un cybercafé, mais cela varie sensiblement avec les activités effectuées par ce dernier. La recherche documentaire, qui prend une partie importante surtout dans les milieux étudiants, pousse l'internaute à passer en moyenne une heure par jour dans un cybercafé. Les bureautiques quant à elles ont deux types d'activités qui les démarquent des autres, la photocopie (40%), et l'impression des documents (30%). Les autres activités (saisie des textes, graphiques, scannage, etc.) occupent seulement 30%. Leur taux de fréquentation est, également, très varié selon la période, l'emplacement et le type de public. Les centres de formation sont fréquentés en majorité par des travailleurs déjà dans la vie active, soucieux d'ajouter une plus-value informatique à leurs qualités et compétences professionnelles. Les centres de formation liés à des institutions d'enseignement sont logiquement fréquentés par des étudiants.

### 2.4.3.4 Et pour finir...

Le coût de fréquentation des cybercafés est en moyenne de 15 Fc<sup>12</sup>/min (environ 12 cents d'Euros). Dans une bureautique, cela varie fortement selon le type de service et l'emplacement de la bureautique, mais notons simplement que le prix d'une photocopie, une activité régulière chez les étudiants, y est de 30Fc (environ 25 cents d'Euros). Une formation coûte généralement 10 dollars américains par module choisi pour des cours de bureautique, les formations spécialisées ont des prix très variables. Nous avons compté au total 63 Cybercafés, 65 Bureautiques, 17 centres de formation pour la ville de Lubumbashi. Nous avons un total de 1169 ordinateurs disponibles pour le grand public dans toute la ville de Lubumbashi, dont 545, soit 47%, sont connectés à Internet en 2010 contre 402 en 2009 et 104 en 2007. Ces ordinateurs ont tendance à être plus vieux pour les centres de formation et un peu plus modernes pour les cybercafés. Les bureautiques dépassent en nombre à la fois les cybers et les centres de formation ce qui est normal, vu que les cybers et centres de formation ont une partie de leurs équipements consacrés également à la bureautique dans la ville de Lubumbashi. Pour la grande majorité des entités, le personnel est souvent formé sur le tas. Nous nous sommes aussi penchés sur l'âge moyen du public de la ville de Lubumbashi qui fréquente les entités informatiques lors de cette enquête, on obtient une moyenne de 22 ans pour les cybercafés, 27 ans pour les bureautiques et 27 ans aussi pour les centres de formations. Nous clôturons donc cette analyse par un tableau synthétique montrant quelques indicateurs du profil de l'infrastructure informatique de la ville de Lubumbashi. Nous proposons la synthèse des résultats dans le tableau 1.2.

### 2.4.4 Bilans et réflexions

Nous rappelons d'entrée de jeu que notre souci à ce niveau était celui de réaliser si la ville de Lubumbashi est capable, du moins par son infrastructure, d'intégrer de manière efficiente et sérieuse un projet d'e-learning dans le cadre d'un enseignement supérieur. L'une des causes des échecs récurrents de ces projets en Afrique est de toute évidence aussi un manque d'aperçu réel des caractéristiques de l'infrastructure informatique du milieu

---

<sup>12</sup> Franc Congolais (monnaie locale)



d'accueil. Il était donc intéressant de capter les indicateurs d'évolution dans le temps, de la disponibilité et de l'usage de l'informatique dans une zone donnée pour se rendre compte de la faisabilité d'un projet d'enseignement via les TIC.

Tableau n° 1.2 : Représentation synthétique du profil informatique de la ville de Lubumbashi en 2010.

| Type d'entité          | Heures<br>ouverture (h) | Coût du<br>service | Age moyen<br>du public<br>(ans) | Nombre<br>d'ordinateurs | Quantité de<br>l'entité |
|------------------------|-------------------------|--------------------|---------------------------------|-------------------------|-------------------------|
| Cybercafé              | 12                      | 15Fc /min          | 22                              | 545                     | 63                      |
| Bureautique            | 12                      | 30 Fc/copie        | 27                              | 295                     | 65                      |
| Centre de<br>formation | 7                       | 10\$/Module        | 27                              | 329                     | 17                      |

A l'issue de cette enquête, nous avons palpé le terrain quant à la réalité de la présence de l'outil informatique dans la ville de Lubumbashi ainsi que la connectivité de cette ville à Internet, évaluant ainsi son taux de désenclavement numérique par rapport au monde. L'Université de Lubumbashi, qui est concernée par cette étude, compte près de 20000 étudiants en son sein. Dès lors, si nous supposons, dans le cadre de la coopération, que tous les étudiants suivent un enseignement en ligne pour un certain nombre de cours choisis, pourront-ils le suivre au vu de la disponibilité de l'outil informatique? Ont-ils accès suffisamment à Internet? Par rapport à cela, la ville de Lubumbashi qui compte en 2010 un total de 1169 ordinateurs, dispose d'un ordinateur pour 17 étudiants. La moitié de ces ordinateurs est connectée à Internet. Avec une moyenne de 12 heures d'ouverture par jour, les cybercafés de la ville de Lubumbashi offrent environ 6804 heures de connexion par jour, ce qui donne un minimum d'à peu près 20 minutes de connexion pour chaque étudiant. Sachant que ces derniers ne suivront pas tous au même moment un projet d'apprentissage

en ligne, et en ne considérant que les étudiants ayant été concernés par l'expérimentation (soit 1100 étudiants), nous obtenons à peu près six heures par étudiant et un ordinateur connecté à Internet pour deux étudiants. L'indicateur du nombre d'ordinateurs connectés à Internet dans la ville de Lubumbashi de 2007 à 2010 passe de 104 à 545 (du simple au quintuple), et révèle de manière encourageante qu'il y a une évolution nette de l'usage de l'outil informatique et d'Internet dans la ville de Lubumbashi comme le montre la figure n° 1.2.

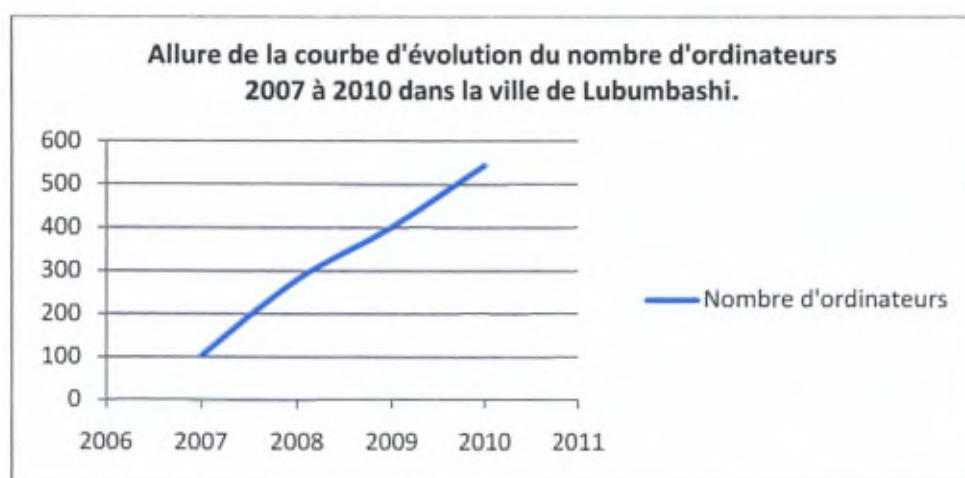


Figure n° 1.2 : Évolution du nombre d'ordinateurs de l'année 2007 à 2010

D'autre part, la jeunesse du public (22 ans en moyenne), et l'affluence majoritaire des jeunes filles (56% d'internautes !) depuis 2008 constituent à leur tour un espoir des lendemains technologiques meilleurs dans la ville de Lubumbashi. Nous estimons donc, au vu de son évolution ces trois dernières années, qu'elle est capable d'accueillir de manière tout à fait viable un projet d'apprentissage en ligne. Surfant sur l'effet de mode que constituent les nouvelles technologies à Lubumbashi (Fyama, 2006), et leur coût de plus en plus abordable pour une grande partie de la population, nous pensons que la ville Lubumbashi offre un cadre adéquat pour un projet d'apprentissage en ligne.

## 2.5 Hypothèses

La conviction que nous avons à travers ces travaux est que l'apprentissage en ligne est certainement une plus-value dans l'enseignement supérieur pour la République



Démocratique du Congo. L'apprentissage en ligne ne devenant utile que lorsqu'il porte les fruits attendus, il est nécessaire de poser les bases de la réussite d'un projet. Notre démarche repose sur les hypothèses suivantes :

1° En Afrique subsaharienne une meilleure connaissance de l'infrastructure informatique du milieu d'accueil s'avère être importante car elle permet de mesurer l'effectivité de l'e-learning dans le contexte propre à un environnement d'apprentissage donné. Il fournit les indices pouvant indiquer l'évolution de la disponibilité de l'outil informatique, est incontournable pour des projets d'enseignement véhiculés par les TIC.

2° En amont de l'importation de contenus venus des universités du nord, il est impérieux de se rendre compte localement, à travers une expérimentation d'une session de formation e-learning, des comportements émergeants qui offrent un scénario fidèle au déroulement réel de l'apprentissage permettant ainsi une appropriation locale de la plateforme, en termes d'outils effectivement utilisés et des mentalités des apprenants face à ce moyen d'apprentissage. Cette étape peut permettre en outre de partir de l'usage des outils, pour voir éventuellement quels sont les outils manquants à créer, ou les outils existants à modifier pour une réelle adaptation au contexte local.

3° L'interopérabilité est ici un facteur majeur pour une garantie du succès de l'e-learning qui permettrait, par son automatisme, des échanges de contenus et de services d'une unité d'enseignement, et d'éviter pour l'enseignant du nord à distance comme pour l'étudiant au sud de céder à la lassitude qui mène à l'abandon que l'on constate souvent. Cette lassitude est en partie causée par la lourdeur qu'engendre le fait de devoir s'accommoder à un nouvel outil autre que son outil local qui n'a généralement pas la même philosophie de fonctionnement. Concevoir sur le plan informatique des applications capables d'assurer les échanges automatiques de contenus et services des plates-formes d'e-learning sollicitant le moins possible d'efforts de la part des utilisateurs multiplierait à coup sûr par un facteur non négligeable les probabilités de succès d'une session d'apprentissage en ligne.

## 2.6 Approche méthodologique

L'approche méthodologique choisie vise à apporter une réponse aux différents éléments de la problématique dégagés plus haut et à remplir nos objectifs de recherche. Pour atteindre nos objectifs de thèse, nous avons initié une enquête pour dégager le profil informatique de la ville de Lubumbashi. De plus, dans le souci de maîtriser l'usage que peut faire un apprenant dans le contexte local des outils informatiques dans le cadre de l'apprentissage en ligne, nous avons participé à la mise en place d'une expérimentation réelle de formation en ligne avec un échantillon des étudiants des Facultés de Médecine humaine et Polytechnique de l'Université de Lubumbashi. Ceci grâce au financement de la Coopération Technique Belge (BTC/CTB). L'expérimentation a notamment permis de déterminer en amont quel est le format adéquat d'un contenu d'enseignement en local. L'approche choisie pour mener à bien nos travaux est pluridisciplinaire, pour traiter les données récoltées de l'enquête du profil, ainsi que le corpus des données d'interaction de l'expérimentation de la session de formation en ligne que nous avons suivies pendant trois ans. En tant que tuteur des ces sessions de formation, nous avons dû nous inspirer des théories de l'enseignement et de la pédagogie pour interpréter les différents comportements des acteurs impliqués, et avons eu un aperçu réel de l'apport des TIC dans l'enseignement dans le contexte local.

Au niveau informatique, nous nous sommes attelés d'abord à maîtriser les deux plates-formes en jeu dans nos travaux, à savoir Moodle et Dokeos, afin de déterminer comment les deux fonctionnent. Puis, nous avons choisi des cours de référence du site officiel de Moodle et l'Université de Mons, les avons implémentés sur Moodle pour ensuite les importer dans Dokeos qui est la plate-forme d'accueil à Lubumbashi. Nous avons aussi déterminé les formats des contenus numériques d'apprentissage de chacune des plates-formes, le listing des métadonnées de description des contenus, la caractérisation de ces métadonnées pour en avoir une signification claire que nous convertissons au format SCORM. Ensuite, nous avons conçu un outil partant de l'API DOM XML, du langage de Programmation Python (avec la classe PyXML) ou de PHP et qui s'assurera de la traduction SCORM et du transfert bidirectionnel automatique des contenus numériques d'une plate-



forme à une autre. Enfin, nous avons développé le module de transfert et de mise à jour des services accompagnant une session d'apprentissage en ligne entre les deux plates-formes.

## CHAPITRE 3. LES ENJEUX DE L'INTEROPERABILITE

### 3.1 Problématique d'interopérabilité des plates-formes d'e-learning : normalisation

Au départ, nous avons des acteurs divers et isolés qui conçoivent chacun dans leur coin des plates-formes d'e-learning pour des besoins spécifiques, dans des contextes différents avec des moyens techniques et technologiques aussi variables les uns des autres. Ceci conduira, surtout au début des années 2000, à la naissance de plusieurs plates-formes (LMS<sup>13</sup>) sur le marché de l'e-learning soumises à des cadres pédagogiques et technologiques distincts. Puis arrive le moment des constats des changements de cadre ou d'outils technologiques motivés par plusieurs raisons telles que le changement de politique des institutions, les besoins d'outils plus adaptés et performants, etc. Un autre souci qui attire l'attention est celui de la portabilité des enseignements qui aide notamment qu'un même enseignement soit donné sur plusieurs plates-formes différentes dans un cadre de partage des connaissances, des savoirs et des compétences qui sont caractéristiques au monde de la formation et de l'enseignement, qui plus est dans un cadre de coopération nord-sud. Avec l'évolution du partage des connaissances et des compétences, un besoin certain de réutilisabilité des contenus d'apprentissage s'est fait ressentir auprès des enseignants, des concepteurs et des développeurs, changeant de cadres à la fois technologiques et pédagogiques. Cette réutilisabilité des ressources et contenus d'apprentissage a engendré à son tour la problématique d'interopérabilité « *pédagogico-Informatique* » des différents systèmes de conception des contenus pédagogiques d'apprentissage ainsi que des plates-formes de formations.

L'obligation s'imposa alors de créer un cadre conceptuel d'expression commune permettant de concevoir, dans un même langage ou à partir des mêmes paradigmes, les contenus pédagogiques afin de permettre une interopérabilité entre les plates-formes. Ainsi naquirent, dans divers contextes et en plusieurs étapes, les différentes initiatives et standards d'e-learning conçus par les principaux acteurs du monde de la normalisation. Ces

---

<sup>13</sup> Learning Management System, équivalent en anglais de la plate-forme d'apprentissage en ligne.



travaux de normalisation et de standardisation ont pour objectif d'apporter des solutions aux questions relatives à l'interopérabilité des technologies, des services et des contenus entre les plates-formes d'e-learning. Les principaux enjeux de la standardisation et de la normalisation sont donc de rendre inter-opérables les technologies entre elles afin de faciliter l'usage des ressources d'apprentissage, quelle que soit la plate-forme ou l'environnement technologique utilisés. Plusieurs acteurs œuvrent dans divers domaines et secteurs, à des échelons variés, allant d'une sphère locale, nationale, ou régionale jusqu'à être internationale. Nous les recensons dans la section suivante.

### 3.2 Processus et acteurs

L'interopérabilité des systèmes est précédée en amont par des travaux de normalisation qui créent les fondements adéquats dans un domaine précis. La normalisation est un processus qui vise à établir des règles garantissant que les différents acteurs d'un domaine scientifique ou d'un secteur professionnel parlent un même langage aussi bien à la conception de leurs produits ou services, qu'à leur utilisation. En e-learning, l'enjeu central de la normalisation se résume à cinq défis fondamentaux, parmi lesquels l'interopérabilité des plates-formes nous intéresse particulièrement. Pour plusieurs chercheurs (Crepuq, 2002; AUF, 2002), ces défis peuvent être énumérés comme suit :

1. **L'accessibilité** pour permettre la recherche, l'identification, l'accès et la livraison des contenus et composantes de formation en ligne de façon distribuée.
2. **L'interopérabilité** qui permet l'utilisation de contenus et composants développés par une organisation sur une plate-forme donnée par d'autres organisations sur d'autres plates-formes.
3. **La réutilisation** des ressources pour permettre aux contenus et composants de répondre à différentes fins, de s'intégrer dans différents produits, et d'être utilisés dans différents contextes et suivant différents modes d'accès. **La durabilité** pour permettre aux contenus et composants de s'adapter aux changements technologiques sans la nécessité d'une ré-ingénierie ou d'un redéveloppement.

4. **L'adaptabilité** qui permet la modulation sur mesure des contenus et des composants.

Il est aussi important à ce stade de cerner le processus de normalisation pour in fine comprendre le monde de la normalisation, les différents acteurs qui y interviennent, les différents profils de ces acteurs là, leurs différents travaux, et à partir de quand ils produisent une norme à proprement parler. Globalement, il est important d'appréhender comment prend naissance une norme depuis sa forme embryonnaire comme spécification en passant par son établissement comme standard et enfin à sa reconnaissance comme norme. Pour cela nous nous baserons ici sur l'étude réalisée par Cyrille Simard pour le compte de l'AUF (AUF, 2002), étude au cours de laquelle il décrit le cycle qui donne naissance à une norme, en cinq phases distinctes :

1. **La phase initiale peut être qualifiée d'embryonnaire.** Il s'agit de l'instant où dans un secteur scientifique ou professionnel donné naît un besoin de normes. Les acteurs concernés, depuis les concepteurs et producteurs des produits jusqu'aux utilisateurs, réfléchissent sur les modalités de création de certaines spécifications en fonction de leur contexte. Dans cette phase, on recense et identifie les contraintes et exigences auxquelles les normes devraient répondre. C'est l'étape de l'élaboration du cahier des charges.

2. **La deuxième phase est celle de la définition ou de l'élaboration de spécifications.** Après identification des contraintes et exigences, et dans le but d'y répondre rationnellement, les différents acteurs développent et élaborent des ensembles structurés et précis de spécifications techniques. Ici les acteurs agissent de façon indépendante et par groupes isolés selon les centres d'intérêt et les différentes appréhensions et philosophies qu'ils ont du problème. Cependant, vu que les problèmes posés concernent tout le monde, des rapprochements deviennent incontournables et donnent alors naissance à des collaborations. Aussi à ce stade, devient-il nécessaire de passer au « *testing* » et à « *l'évaluation* » des produits et services élaborés à l'aune des spécifications définies. En effet, ce qui assure la validité des spécifications c'est leur stabilité et leur caractère testable.



**3. La troisième phase est la phase de testabilité.** A ce niveau, parmi les groupes impliqués dans les travaux des spécifications, quelques uns vont émerger du lot et avancer en développant des projets pilotes, des prototypes, etc. En outre, ces groupes doivent être susceptibles d'effectuer des tests de validité des spécifications dans la réalité concrète. Les tests se font de manière itérative au fur et à mesure qu'ils sont concluants, et si les spécifications y afférents sont jugées stables, elles sont consignées dans des documents écrits que l'on appelle « *lignes directrices* », « *guide d'implantation* », « *modèle de référence* », « *schéma* », etc. Lorsque ces efforts de collaboration prennent de l'ampleur et convergent, de larges consortiums se forment dans le but d'économiser à la fois les efforts consentis et les moyens financiers. Les modèles continuent d'évoluer vers un plus grand degré de stabilité par un cycle itératif : *spécifications – testing – spécifications – testing*, etc.

**4. La phase 4 est celle de la standardisation.** Ce stade correspond à celui de la consolidation et du raffinement des acquis des expériences effectuées. L'efficacité des modèles est augmentée et de plus en plus confirmée. On examine les échecs, on en détermine les causes et on les corrige éventuellement. Il se développe alors ce que l'on nomme des « *standards de fait* » qui sont en fait des modèles dominants qui prennent le pas sur l'industrie et s'imposent de facto comme des modèles à suivre. Il apparaît alors des organes d'accréditation ou de certification capables d'adjuger la conformité des produits et services aux standards devenant ainsi des « *standards de droit* » ou « *standards accrédités* ».

**5. La dernière phase est celle de la normalisation.** Les standards devenus matures, sont soumis à des discussions, validations et sanctions officielles dans le cadre d'une démarche ouverte qui vise à assurer le plus possible un haut degré de précision et de consensus parmi les acteurs. Les standards deviennent alors des normes et cette décision finale ne peut être prise que par un organisme reconnu légalement à cette fin au niveau national, régional ou enfin international. Notons par ailleurs que dans plusieurs pays, certaines normes internationales (par exemple celles de l'ISO) ont force de loi sur le plan national.

### 3.2.1 Différents acteurs de la normalisation

#### 3.2.1.1 AICC (*Aviation Industry Computer-Based Training Committee*)

L'Aviation Industry CBT (Computer-Based Training) Committee (AICC), une des pionnières dans le domaine de l'apprentissage soutenu par des moyens informatiques, est une organisation internationale. Fondée en 1988, elle regroupe tous les intervenants dans le domaine de la formation de l'industrie aéronautique (constructeurs d'avions, vendeurs de plates-formes informatiques d'enseignement, fournisseurs des contenus pédagogiques, etc.). Cette association est cependant ouverte aux personnes provenant d'autres industries.

Ces standards proposés sont par nature génériques et ont vocation de traiter de manière universelle la question de l'apprentissage par des moyens informatiques et de l'ingénierie. Ils ne se focalisent donc pas uniquement sur les thématiques liées à l'aviation et sur le domaine aéronautique. L'AICC publie des *AICC Guidelines & Recommendations (AGR)* dans différents domaines comme par exemple l'AGR-007 (AICC, 1995), qui traite de l'échange de contenus entre plates-formes, ou l'AGR-002 (AICC, 2002) qui traite de la plate-forme matérielle d'un poste client (vitesse du microprocesseur, capacité disque dur, etc). Il existe à l'heure actuelle neuf AGRs différentes. Celle qui nous intéresse le plus est l'AGR-006 (AICC, 1998) qui traite de l'interopérabilité entre *Computer Managed Instruction (CMI)*, un CMI étant défini comme un système gérant à la fois le comportement connecté (on-line) et déconnecté (off-line) des activités d'apprentissage. Il peut notamment s'appuyer sur des CBTs. L'AGR-006 décrit, notamment de manière technique, l'échange de fichiers représentant la structure d'un cours (AICC, 2001).

#### 3.2.1.2 Advanced Distributed Learning

L'initiative ADL (*Advanced Distributed Learning : Apprentissage Distribué Avancé*) a été lancée en 1997 par le Département de la Défense américain avec, entre autres, pour objectif d'en faire une créatrice des bibliothèques de savoirs, un vivier de connaissances, où les objets d'apprentissage sont accumulés et catalogués pour une distribution et un usage à grande échelle. Ces objets doivent être facilement accessibles par le Web. Le



développement de tels viviers de connaissances peut contribuer à l'établissement d'une économie des objets d'apprentissage qui récompenserait les créateurs de contenus à forte valeur ajoutée. Ces objets d'apprentissage seront accessibles, partageables et capables de s'adapter à la demande d'apprentissage des utilisateurs. Une des clés de l'initiative ADL est la possibilité de pouvoir réutiliser les composants des objets d'apprentissage dans des applications et environnements multiples, sans avoir à se soucier des outils utilisés pour les créer. Ceci implique, entre autres choses, que le contenu soit séparé des contraintes liées au contexte et aux spécificités du logiciel d'exécution de telle sorte qu'il puisse être inclus dans d'autres applications. De même, pour que son usage répété soit possible sous diverses formes, le contenu doit avoir une interface et des métadonnées communes. Le SCORM (*Sharable Content Object Reference Model*) est l'une des principales actions de ADL pour répondre à la demande d'interopérabilité des contenus d'apprentissage (Friesen, 2004).

### 3.2.1.3 IMS Global Learning Consortium

IMS est un consortium plus récent (1999) qui représente un regroupement de 250 établissements éducatifs dont le MIT (*Massachusetts Institute of Technology*) et UCM (*Université Carnegie Mellon*), d'entreprises telles que *Apple* et *IBM*, d'agences gouvernementales telles qu'*Industrie Canada* et de sociétés de développement telles que *Canvas Learning* et *Blackboard*. IMS participe au développement de standards pour la formation en ligne en proposant des spécifications relatives à la description, au repérage et à l'échange de contenu, à l'interactivité et à l'interopérabilité. Ainsi, IMS a spécifié des métadonnées qui permettent l'étiquetage des ressources d'enseignement et d'apprentissage et élaboré diverses autres spécifications relatives notamment aux contenus, au classement, au design pédagogique, aux profils des apprenants et aux besoins des entreprises. Les spécifications réalisées par IMS sont les suivantes :

1. *IMS Content Packaging* : Ce sont des spécifications qui permettent de définir un moyen pour échanger les ressources pédagogiques entre différentes plates-formes d'apprentissage en ligne. Dans ces spécifications, IMS propose

que les contenus des supports pédagogiques soient regroupés dans un paquetage.

2. *IMS Questions and Tests* : Ce sont des spécifications qui permettent de définir une structure facilitant la représentation des questions et des évaluations. L'échange de données entre différents systèmes de gestion de la formation a évidemment été pris en compte dans ces spécifications.
3. *IMS Meta-data* : Ce sont des spécifications qui concernent les ressources pédagogiques. Elles représentent un processus permettant de rechercher et utiliser plus efficacement une ressource.

### 3.2.1.4 Dublin Core Metadata Initiative

Le concept de DCMI (*Dublin Core Metadata Initiative*) n'est pas spécialement dédié au domaine de l'enseignement, c'est le monde de l'enseignement qui l'a adapté à ses objectifs, ses spécificités et ses besoins. DCMI tire son origine de la seconde *International World Wide Web Conference* de Chicago où une des discussions a porté sur le manque de marquage sémantique dans le Web. Ceci a donné naissance à une journée d'étude sur le sujet à Dublin, Ohio en 1995. La DCMI est le résultat de cette journée (DCMI, 2004). Elle est devenue une organisation tentant de promouvoir l'adoption de standards de métadonnées compatibles entre eux. Son but est de faciliter les recherches des ressources et l'interopérabilité de l'échange d'informations. La direction de Dublin Core coordonne les groupes de travail, les activités des groupes et dissémine l'information à travers le Web et d'autres publications. Les groupes de travail fonctionnent en collaboration et participent au raffinement des conventions sur les métadonnées. Parmi les participants au projet, on trouve, des agences gouvernementales, des organisations commerciales, des bibliothèques, des musées, des universités. Et tous ces participants ont créé les 15 *Dublin Core Metadata Elements*, soit le *DCM Element Set*.

### 3.2.1.5 Institute of Electrical and Electronics Engineers

L'IEEE (*Institute of Electrical and Electronics Engineers* : Institut des Ingénieurs en électricité et électronique) est une organisation centrale qui joue un rôle déterminant



comme pôle de réflexion et de proposition en matière de standards. Dans son rôle d'organisation accréditée pour développer des normes, l'IEEE soumet le plus souvent les projets de standards développés et produits au sein de son organisation, à l'Institut national américain de normalisation (ANSI : *American National Standard Institute*) qui lui-même les présente à l'ISO (section 3.2.1.6). Depuis 1998, l'IEEE pilote le comité de normalisation des technologies éducatives IEEE/LTSC (*Learning Technology Standards Committee*). Ce comité comprend 20 groupes de travail qui couvrent l'ensemble des champs à normaliser dans l'apprentissage en ligne : métadonnées, informations sur l'apprenant, gestion des contenus, de l'interactivité, etc. Cet organisme international regroupe entre autres le Canada, les États-Unis, plusieurs pays d'Europe, d'Afrique, d'Amérique latine, d'Asie et des régions du Pacifique. Le LTSC et le groupe de travail LOM s'emploient à définir des éléments pour des objets d'apprentissage. Le sous-comité LTSC de IEEE est également associé aux travaux de ISO/IEC JTC1 SC36 et d'IMS.

### 3.2.1.6 Joint Technology Committee, Subcommittee on Standards for Learning, Education, and Technology

Le Comité ISO/IEC JTC1 SC36 (*Joint Technology Committee, Subcommittee on Standards for Learning, Education, and Technology de l'ISO*) travaille à la normalisation des technologies de l'information destinées à l'apprentissage, à l'enseignement et à la formation. Plus particulièrement, il se concentre sur les systèmes d'information destinés aux apprenants, aux institutions et sur les ressources éducatives. Il se compose de plusieurs groupes qui œuvrent, entre autres, sur le vocabulaire, les technologies collaboratives, l'architecture, les métadonnées, ainsi que les adaptations culturelles, linguistiques et fonctionnelles. Plus d'une vingtaine de pays y participent et plusieurs groupes y collaborent : IEEE/LTSC, AICC, ARIADNE<sup>14</sup>, IMS, ALIC, ADL, DCMI, CEN/ISSS/WSLT. Nous notons qu'il existe un niveau de maillage important à l'intérieur de la plupart de ces consortiums. Par exemple, L'ADL, bénéficie de la collaboration d'expertises de l'ARIADNE européenne, de l'AICC américaine, de l'IEEE américaine et de l'IMS.

---

<sup>14</sup> *Alliance of Remote Instructional Authoring and Distribution for Europe.*

### 3.2.1.7 Comité Européen de Normalisation

En Europe, le Comité Européen de Normalisation/*Information Society Standardization Society* (CEN/ISSS), une association reconnue par la Communauté européenne, coordonne les différentes instances de normalisation et de standardisation, et s'occupe spécifiquement des exigences de la normalisation. Un de ses groupes se concentre sur les métadonnées pour l'information multimédia. Parmi ses principaux membres, on retrouve IEEE-LTSC, IMS et DCMI.

### 3.2.1.8 Bureau de la normalisation en Belgique

Notons en outre que au-delà de la description faite ci-dessus d'autres groupes de travail et organisations existent à l'échelle nationale dans certains pays comme l'Australie, la France, le Japon, la Belgique etc. Prenons, par exemple, la Belgique il y existe le NBN, Bureau de la normalisation en Belgique. Tout commence en 1919 avec la création de l'Association belge de standardisation (ABS) sous, le patronage du Comité Central Industriel, suivi de la création en 1945 par le législateur de l'Institut Belge de Normalisation (IBN). L'évolution des technologies imposera une nouvelle approche de la normalisation en Belgique en ce sens qu'il fallait pour plus d'efficacité décentraliser la normalisation selon les différentes spécialités et sous la coordination d'un organisme public. Ainsi le conseil d'administration du NBN procède à des choix parmi les commissions de normalisation (TC, «*Technical Committees*») actives au niveau du CEN et de l'ISO. Une fois qu'un TC est adopté au niveau belge, il est alloué à un opérateur sectoriel. Cette décentralisation ramène la normalisation vers des spécialistes d'un secteur détenteurs des connaissances spécifiques pour traiter les divers aspects de la question. Cet opérateur sectoriel anime à son tour les activités de la commission qui intervient comme commission-miroir belge pour son homologue CEN ou ISO. Pour ce qui est de l'apprentissage en ligne l'opérateur sectoriel désigné en Belgique est Agoria. Et en tant que fédération sectorielle, Agoria représente les entreprises de neuf secteurs: métaux et matériaux, produits métalliques, plastiques, mécanique et mécatronique, électrotechnique et électronique, technologies de l'information et de la communication (TIC), automobile, aérospatiale et défense et sécurité. Le TC désigné par le



NBN est le TC 353, le nom de cette commission est *Information and Communication Technologies for Learning, Education and Training*.

### 3.2.2 Conclusion de la section

Notons avant toute discussion que les acteurs de la normalisation n'agissent pas tous ni de la même façon ni au même niveau dans le processus de normalisation. Se basant sur les travaux du Sous-Comité sur les Technologies de l'Information et de la Communication (SCTIC) qui réunit les universités du Québec au Canada (Crepuq, 2002), les acteurs recensés dans ce travail peuvent être regroupés en trois catégories. La première est celle des créateurs qui pressentent et développent les spécifications susceptibles de déboucher sur des standards qui finiront par être établis comme des normes. La deuxième catégorie concerne les acteurs qui appliquent les normes en développement, leur donnent un corps et élaborent ce faisant des protocoles décrivant leur implantation dans un contexte donné. Enfin, le troisième groupe d'acteurs est celui des organismes de normalisation qui sont des institutions officiellement reconnues pour la reconnaissance, la validation et la promulgation des normes. Les travaux des uns et des autres groupes constituent les différents maillons de la chaîne dans le processus de normalisation, d'où une étroite collaboration existe entre eux pour mener à bien le processus de normalisation. Ainsi les huit acteurs recensés dans notre travail peuvent être répartis comme le montre les tableaux 3.1, 3.2, 3.3. La diversité des acteurs dans le domaine de la normalisation révèle aussi comment une interopérabilité entre divers systèmes surtout informatiques s'impose de facto. On constate, en observant de manière globale l'évolution de la normalisation, qu'on démarre avec les acteurs d'un même domaine scientifique ou professionnel qui peuvent développer des paradigmes communs pour normaliser leurs secteurs. Ils font ensuite reconnaître leurs travaux à un niveau national, ensuite régional puis international. Au niveau international, on observe aussi un souci de vouloir aller plus loin en essayant de transcender les différents domaines concernés par la normalisation afin de créer un paradigme global qui coordonnerait une normalisation plus efficace et permettrait une interopérabilité optimale entre les différents systèmes existants dans les domaines des sciences et professionnels à travers le monde.

Tableau 3.1 : créateurs des normes et standards

|  |
|--|
| IMS - IMS Global Learning Consortium (IMS) - États-Unis<br><a href="http://www.imsproject.org/aboutims.html">http://www.imsproject.org/aboutims.html</a> |
| DCMI - Dublin Core - Dublin Core Metadata Initiative - Ohio, États-Uni<br><a href="http://dublincore.org">http://dublincore.org</a>                      |
| AICC - Aviation Industry CBT Committee (EAO)<br><a href="http://www.aicc.org/index.html">http://www.aicc.org/index.html</a>                              |

Tableau 3.2 : groupes qui appliquent les normes et standards (profils et protocoles d'implantation).

|  |   |     |
|--|---|-----|
| ADL – SCORM<br><a href="http://www.adlnet.org">http://www.adlnet.org</a>                   | Projet Sharable Content Object Reference Model du<br>Advanced distributed learning -Défense américaine et<br>enseignement universitaire | IMS |
| ARIADNE<br><a href="http://ariadne.unil.ch/Metadata/">http://ariadne.unil.ch/Metadata/</a> | Alliance of Remote Instructional Authoring and<br>Distribution Networks for Europe  | IMS |

Tableau 3.3 Organismes de normalisation œuvrant dans l'e-learning.

|   |   |   |
|---|---|---|
| ISO – JTC 1- SC36<br><a href="http://jtc1sc36.org/">http://jtc1sc36.org/</a><br><a href="http://jtc1sc36.org/related_activities.html">http://jtc1sc36.org/related_activities.html</a> | International Standards<br>Organisation-Joint Technical<br>Committee no 1 –Sous-comité 36 -<br>Chantier de normalisation des<br>systèmes d'information destinés à<br>l'enseignement et la formation.<br>International   | IEEE/LTSC<br>CEN/ISSS<br>AICCC<br>ARIADNE<br>IMS<br>ALIC<br>ADL<br>DCMI |
| IEEE – LTSC<br><a href="http://ltsc.ieee.org/wg12/index.html">http://ltsc.ieee.org/wg12/index.html</a>  | Institute of Electrical and<br>Electronics Engineers, Inc. -Learning<br>Objects Metadata working group -<br>International   | ISO-JTC1-<br>SC36   |
| CEN/ISSS<br><a href="http://www.cenorm.be/iss">http://www.cenorm.be/iss</a><br><a href="http://www.cenorm.be/iss/Workshop/lt/">/Workshop/lt/</a>                                      | Comité européen de normalisation<br>-Information Society<br>Standardization System<br>PROMETEUS initiative <i>PROMoting<br/>Multimedia access to Education and<br/>Training in EUropean Society</i> ),<br>URL : <a href="http://prometeus.org">http://prometeus.org</a> | IEEE-LTSC   |
| W3C <a href="http://www.w3.org/Metadata/Activity.html">http://www.w3.org/Metadata/Activity.html</a>   | The World Wide Web Consortium   |   |



Reste maintenant à mettre tout le monde d'accord et autour d'une même table, ce qui, principalement pour des raisons techniques, technologiques ou stratégiques, n'est pas encore atteint.

Au niveau africain, et de manière très pertinente pour ce qui concerne l'Afrique subsaharienne, une interopérabilité entre plates-formes d'e-learning favoriserait des échanges intenses avec les universités du nord et multiplierait ainsi les chances de réussite de la mise à jour des enseignements et de la recherche scientifique dans les universités du sud. L'interopérabilité permettrait aussi de diminuer en Afrique les échecs et les abandons récurrents que l'on constate face aux nombreuses tentatives des projets d'e-learning suite à la lassitude engendrée par les changements d'outils technologiques que l'on doit s'approprier. Souvent, on constate en effet autant d'acteurs que d'outils en jeu, ce qui rend une telle approche très complexe. Les avantages de l'interopérabilité sont à la fois par rapport au transfert des contenus (donc des unités d'apprentissage), en permettant une mise en ligne d'un cursus de formation qu'un enseignant peut de manière permanente administrer depuis sa plate-forme locale, et aussi par rapport à la communication des services entre applications d'e-learning (le courrier, les forum, les mises à jour automatiques des cours, le suivi des apprenants, etc.) permettant ainsi un suivi des apprenants et une meilleure compréhension de la matière enseignée.

### **3.3 Concepts techniques généraux d'interopérabilité en e-learning**

#### **3.3.1 Introduction**

En définitive nous retiendrons que la normalisation est un processus de facilitation du travail dans un secteur scientifique ou professionnel donné qui part d'un besoin de normes ressenti par les acteurs concernés allant des concepteurs et producteurs des produits jusqu'aux utilisateurs. Si chacun commence de son côté à élaborer des spécifications, la concertation avec les autres acteurs devient inévitable sous forme d'une collaboration à plusieurs niveaux. Vient ensuite une phase de vérifications et tests de spécifications conçus dans des situations réelles. Lorsque les spécifications sont corroborées

par les expériences, elles deviennent des standards reconnus dans le secteur. Ceci conduit enfin à la validation officielle des standards par un organisme reconnu. Au vu de ce qui précède, la normalisation nécessite donc une description, dans un langage commun pour tous les acteurs, des concepts entrant en jeu dans une problématique et un domaine précis. Langage qui doit du reste demeurer à la fois compréhensible par l'homme et par la machine d'où la création de descripteurs (niveau syntaxique) avec une signification (niveau sémantique) commune et convenue avec tous les acteurs concernés. Les descripteurs à leur tour devraient être stockés dans un repère accessible, au fonctionnement transparent et au format connu et exploitable par tout le monde. Viendraient ensuite les outils capables d'opérer des extractions, des modifications, des ajouts, et des suppressions sur le stockage des descripteurs, afin que les échanges s'opèrent en toute tranquillité et de manière automatique. Cette réflexion conduit donc les acteurs de l'e-learning à concevoir des concepts techniques dédiés à la normalisation et à l'interopérabilité dans la communauté, permettant ainsi de matérialiser le plus possible la normalisation qui a pour finalité l'interopérabilité entre diverses applications d'e-learning. Mais, en e-learning, quels sont ces descripteurs? Ce sont les métadonnées. Comment crée-t-on et où trouve-t-on ces métadonnées? Les standards et les normes ont le rôle de les déterminer. Que décrivent ces métadonnées? Elles décrivent des objets pédagogiques<sup>15</sup>.

### 3.3.2 Définitions des différents concepts d'interopérabilité

Tentons maintenant de définir tous ces concepts en se basant sur les définitions proposées par le ministère français de l'éducation dédié l'apprentissage en ligne<sup>16</sup>. Ainsi on appelle :

1. Norme : « *Ensemble de règles fonctionnelles ou de prescriptions techniques relatives à des produits, à des activités ou à leurs résultats, établies par consensus des spécialistes et consignées dans un document produit par un*

<sup>15</sup> Voir section 6.4, pour plus de détails.

<sup>16</sup> <http://www.educnet.education.fr> (consulté en janvier 2011)



*organisme, national ou international, reconnu dans le domaine de la normalisation.»*

Nous dirons pour notre part que la norme est la dernière couche d'une démarche conduisant à permettre une interopérabilité des systèmes. Elle est le résultat d'un processus laborieux pouvant s'étaler sur plusieurs mois, aussi bien sur un niveau national, régional ou international, et peut avoir dans certains pays un statut de loi. Actuellement des organismes reconnus mondialement comme l'ISO, *International Standard Organisation*, constituent des références en matière d'assurance qualité dans des très nombreux domaines et pour plusieurs industries au niveau international.

2. Standard : « *Ensemble de recommandations développées et préconisées par un groupe représentatif d'utilisateurs ou de fournisseurs.* »

Le fait que l'on rencontre plusieurs acteurs dans l'exploitation d'un produit ou d'un service impose qu'un secteur bien déterminé puisse se doter d'un modèle commun d'expression, conduisant à des manières de faire, de concevoir, de produire, d'utiliser, etc., communes pour tout le secteur. Nous citerons ici à titre illustratif les RFC (*Request For Comments*) de l'IETF (*Internet Engineering Task Force*) ou les recommandations du W3C ou de l'IEEE. Les standards constituent évidemment l'avant dernière étape dans le processus de normalisation.

3. Spécification : « *Ensemble des règles et des prescriptions techniques établies pour une entreprise et qui servent à fixer les caractéristiques permettant de définir un élément de matériel ou de construction utilisé pour un projet donné.* »

Avec les spécifications nous sommes encore à une étape embryonnaire dans la démarche de normalisation : c'est l'agrégat de spécifications qui donnera naissance à des standards. De manière très spécifique en e-learning les standards sont bâtis sur un ensemble de spécifications qui permettent de décrire les éléments de l'apprentissage, et son organisation pour sa réutilisation et sa migration vers d'autres environnements ou plateformes d'e-learning. Citons quelques standards et spécifications des objets d'apprentissage :

- formats de paquets pour la migration de contenu (ex. *IMS Content Packaging*) ;
- description du contenu pour sa recherche ou classification (ex. *IMS Meta-Data*) ;
- spécifications de test ou examens (ex. *IMS QTI*) ;
- sharable Content Object Models (ex. *SCORM*) ;
- learning content exchange formats (ex. *IMS Learning Design*) ;
- digital content repository frameworks (ex. *IMS Repositories* ).

Pour ce travail nous attacherons une attention particulière aux spécifications IMS et SCORM pour examiner la description des contenus d'apprentissage (*IMS Content Packaging*), les métadonnées permettant leur recherche et leur classification et les outils pouvant les transporter d'une plate-forme à une autre, notamment entre la plate-forme de l'Université de Lubumbashi et ses partenaires belges.

#### 4. Les métadonnées : « *Ce sont des données qui décrivent d'autres données.* »

Elles permettent de:

- faciliter le partage d'information ;
- contribuer à la minimalisation de pertes de données ;
- rechercher des ressources dans un entrepôt de ressources d'enseignement ;
- faciliter le stockage des ressources d'enseignement.

Nous avons, en introduisant cette section, bien stipulé que la problématique centrale de la normalisation visait à créer des descripteurs d'objets bien identifiés composant un produit ou un service dans un langage commun et compréhensible pour tous. La gestion de l'enseignement en ligne est d'identifier l'unité d'apprentissage ou l'objet d'apprentissage à décrire pour une normalisation. Alors qu'est ce qu'un objet d'apprentissage ?

#### 5. Les objets d'apprentissage : en définitive une spécification e-learning a pour but la description d'un objet d'apprentissage ou d'un ensemble d'objets d'apprentissage. Dans la littérature e-learning les chercheurs ne fournissent pas une définition exacte d'un objet d'apprentissage, certains parlent même d'unité d'apprentissage ou d'objet pédagogique (Lejeune, 2006).



Dans cette thèse nous partirons de la définition suivante donnée par IEEE « *Un objet d'apprentissage est une entité digitale, numérique (ou non), qui peut être utilisée, réutilisée et référencée pendant un apprentissage par e-learning* » (IEEE, 2001). Une autre définition de l'objet d'apprentissage est celle qui le qualifie d'une brique à partir de laquelle sont construits les nouveaux cursus de formation en e-learning (Pernin, 2003).

De manière générale un objet d'apprentissage doit répondre aux caractéristiques suivantes :

- il doit être en format digital pour être utilisé sur le Web ;
- il peut être décrit par un ensemble de métadonnées ;
- il devrait être réutilisable, et doit avoir un degré de granularité et une indépendance de contexte.

Notons qu'à ce jour, pour l'e-learning, le standard IEEE LOM spécifie la syntaxe et la sémantique de l'objet qui permet de décrire un objet d'apprentissage. Un objet métadonnée (*Learning Object Metadata –LOM*) a donc pour mission de stocker l'information des attributs d'un objet d'apprentissage qui seront nécessaires pour leur description sémantique. Leurs attributs sont aussi relatifs à la catégorisation et à l'heuristique, à la gestion et à l'évaluation. Ces attributs peuvent être : type d'objet, description, format, créateur, location, style d'apprentissage, niveau, pré requis (voir section 6.4).

En conclusion, la description d'un objet d'apprentissage par des métadonnées conçues de manière méthodique pour l'apprentissage en ligne, constitue le gage d'une interopérabilité entre applications ou plates-formes. Il est par ailleurs intéressant de se rendre compte que la coupure épistémologique entre un standard et une norme n'est pas si claire que ça dans la littérature e-learning. Cela traduit un non-achèvement du processus assez lourd qu'est la normalisation en e-learning, malgré les avancées au sein du sous-comité ISO/IEC JTC1 SC36 qui est chargé d'élaborer des normes internationales sur l'e-learning. Une approche qui départage les deux concepts de manière suffisamment nourrie est celle qui stipule qu'une norme est un ensemble de règles sanctionnées par des accords

juridiques tandis qu'un standard correspond à un produit ou un service qui s'impose dans le temps sur le marché et qui par sa position dominante, amène les concurrents à rendre compatibles leurs produits et services. In fine, la norme relève du droit et le standard relève des faits (Arnaud, 2002). Et les faits en l'absence des normes d'e-learning claires nous imposent de nous intéresser de manière particulière pour nos travaux à deux standards qui, par leur nombre d'utilisateurs dans le monde académique, s'imposent de fait comme des normes. Il s'agit de l'IMS (*IMS Global Learning Consortium*) et du SCORM (*Sharable Content Object Reference Model*) que nous allons développer dans la section suivante.

Dans le cadre de cette thèse, on sait d'ores et déjà que l'interopérabilité des plates-formes d'apprentissage en ligne que nous visons recouvre deux dimensions. Une première est celle qui permet un échange des contenus que nous qualifions à ce stade de « physique » des unités d'enseignements (ressources) entre plates-formes hétérogènes. Une deuxième est celle qui devra permettre un échange entre des services (applications) de ces différentes plates formes pour échanger des éléments sur notamment la présentation des apprenants, le suivi, les échanges (interactions), la production des rapports, la gestion du contenu d'apprentissage des apprenants, la planification des séances des cours et des activités du cours, etc. Et en ce sens il sera donc nécessaire pour notre application de se référer à certaines normes, à des standards, à des spécifications établis et reconnus, ainsi qu'à des modèles de référence, tels que IMS et SCORM, pour arriver à faire dialoguer des applications hétérogènes.

## 3.4 Aperçu de IMS et SCORM

### 3.4.1 Historique de IMS

Le consortium *IMS Global Learning Consortium Inc.* développe et favorise l'adoption de spécifications techniques ouvertes garantissant l'interopérabilité des systèmes d'apprentissage en ligne. Ses spécifications et les publications associées sont accessibles librement au public et plusieurs d'entre elles sont retenues à l'échelle internationale pour l'implémentation de produits et de services éducatifs. Le nom définitif d'IMS est *IMS Global Learning Consortium*, aussi parfois mentionné comme IMS GLC. Le nom original, lorsque IMS



a commencé en 1997 était le *Instructional Management Systems (IMS) Project* (projet de systèmes de Gestion d'Instruction). Quelques temps après, il devint évident que l'appellation «*Instructional Management Systems*» créait de la confusion avec d'autres concepts utilisés pour décrire la même chose comme *Course or Learning Management System*, *Learning Server*, *CBT (Computer-Based Training) System*, *Integrated Learning System*. C'est pourquoi, ayant pour vocation d'établir une interopérabilité entre systèmes d'apprentissage en ligne, il a choisi de s'appeler *IMS Global Learning Consortium*.

Les travaux d'IMS ont successivement porté sur les différents domaines liés à l'e-learning (Lejeune, 2004): indexation des ressources d'apprentissage (*IMS Meta-Data*), profil de l'apprenant (*IMS Learner Information Package*), description des compétences (*IMS Reusable Definition of Competency or Educational Objectives*), interrelations au sein d'une organisation (*IMS Enterprise*), interopérabilité des systèmes et des environnements (*IMS Content Packaging*), évaluation des connaissances et des compétences (*IMS Question & Test Interoperability*), métamodélisation d'une unité d'apprentissage (*IMS Learning Design*), et interopérabilité des banques de ressources pédagogiques (*IMS Shareable State Persistence*). Toutes ces spécifications portant sur l'apprentissage en ligne ont tout naturellement des similitudes et des connexions étroites.

### 3.4.2 Les spécifications IMS

Les spécifications constituent les éléments de base fournies par l'IMS. Les documents des spécifications sont rendus disponibles pour l'accès public une fois approuvé par l'*IMS Technical Advisory Board*. Avant cela, les projets des spécifications sont dans un processus de raffinement final et sont seulement disponibles pour les membres d'IMS GLC et les filiales contribuant aux projets. Ainsi, à ce jour, IMS a produit les spécifications suivantes :

1. Le « *IMS Enterprise Specification* »: formats pour échanger l'information des cours et des étudiants entre les composants du système.
2. Le « *IMS Learning Resource Meta-data Specification* »: données qui décrivent un objet d'apprentissage et l'indexation des ressources d'apprentissage.

3. Le « *IMS Simple Sequencing* »: spécifications sur la manière dont les objets d'apprentissage sont triés et fournis aux étudiants.
4. Le « *IMS Content Packaging Specification* »: instructions sur le packaging et l'échange de contenus pour une interopérabilité des systèmes.
5. Le « *IMS Learner Information Package Specification* » : information à propos des capacités, les expériences et les privilèges<sup>17</sup> des étudiants.
6. Le « *IMS Question and Test Specification* »: formats pour la construction et échange de contrôles d'apprentissage.

Les spécifications IMS sont dotées d'une documentation riche avec accès libre au grand public depuis le site officiel du consortium. Pour ce mémoire de thèse nous allons nous employer en n'en décrire qu'un seul qui cadre avec nos travaux, il s'agit de *Content Packaging*.

### 3.4.2.1 IMS Content Packaging

IMS Content packaging est une spécification qui décrit les structures de données qui peuvent être utilisées pour échanger des données entre des systèmes qui veulent importer, exporter, agréger et démonter les paquets de contenus. Les *packages IMS Content* permettent l'exportation du contenu d'un système et son importation dans un autre tout en conservant des informations le décrivant dans le *package content* et comment il est structuré soit en table des matières soit pour déterminer quelle est la page Web à montrer en premier.

#### 3.4.2.1.1 Description du IMS Content Package:

Par définition «*Un paquet IMS représente une unité d'apprentissage utilisable et réutilisable*». (IMS, 2003). Il peut être une leçon, une partie ou un module d'un cours qui peut avoir une importance particulière, un cours entier, ou une collection de cours.

Un package IMS contient traditionnellement deux composants principaux :

---

<sup>17</sup> Les privilèges déterminent les droits, à accéder aux différents outils de la plate-forme. Par exemple : gérer les forums, administrer un groupe, etc.



1. Un document XML spécial et obligatoire qui décrit l'organisation intrinsèque du contenu appelé fichier manifeste *imsmanifest.xml* car il répertorie le contenu comme dans un manifeste.
2. Les ressources physiques et les fichiers référencés dans le manifeste.

La structure d'un package IMS content est illustré par le diagramme conceptuel dans la figure 3.1 :

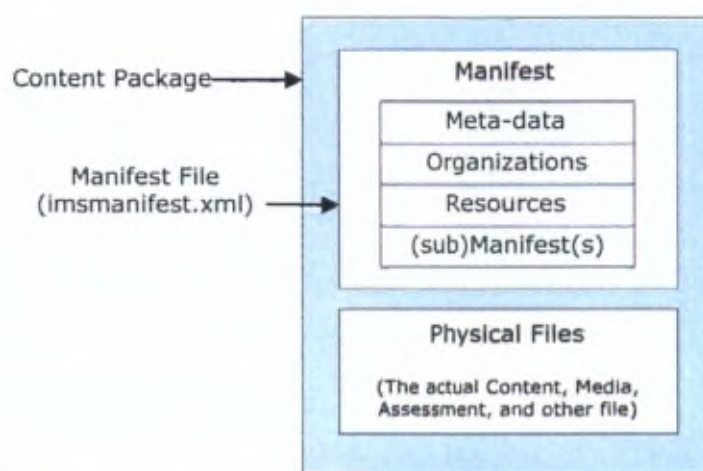


Figure 3.1 : Diagramme conceptuel d'un content packaging tiré de (IMS, 2007).

#### 3.4.2.1.1.1 Le fichier manifeste

On voit bien sur le diagramme que le *content package* doit contenir à sa racine un fichier XML qui décrit le contenu du paquet, et qui doit toujours s'appeler *imsmanifest.xml*. Ce fichier en arbre XML est composé des nœuds principaux suivants :

1. **Meta-data** : La section meta-data est utilisée pour décrire les composants du *IMS content package* à des niveaux différents, depuis la description du package comme un tout jusqu'à la description de chaque composants et sous-composants de ce dernier. Les métadonnées servent aussi à décrire l'organisation du contenu (*Organizations*) ainsi que les ressources (*Resources*).
2. **Organizations** : L'élément *Organizations* donne la structure du contenu et est typiquement donnée sous forme d'une taxonomie hiérarchique

d'apprentissage. La spécification *IMS Content Packaging* stipule par ailleurs qu'elle n'oblige pas l'utilisateur à être soumis à une structure particulière, elle lui donne par contre les moyens d'en concevoir autant que nécessaire.

3. **Resources** : L'élément *Resources* peut décrire aussi bien des fichiers externes au package que les fichiers physiquement présents au sein du package. Ces fichiers peuvent être des fichiers multimédia, des fichiers textes, des pages Web ou n'importe quel type de données sous format électronique. Cet élément peut par ailleurs représenter les groupements conceptuels et les relations qui peuvent exister entre les différents fichiers.
4. **Sub-manifest** : Cet élément décrit un ou plusieurs manifestes contenu dans le principal.

#### 3.4.2.1.1.2 Les ressources physiques

L'élément *Physical files* représente les fichiers référencés comme ressources du contenu. Ces fichiers peuvent être des fichiers locaux contenus actuellement dans le *content package*, ou bien des fichiers externes qui sont référencés par des URI (*Universal Resource Indicator*). Les fichiers peuvent ici être répartis en répertoires et sous-répertoires.

#### 3.4.2.1.2 Package Interchange File (PIF)

Le PIF est un fichier compressé sous un format d'archive (ex. \*.zip, \*.jar, \*.zip) qui doit contenir le fichier Manifeste dont le nom standard doit toujours être « *imsmanifest.xml* » et les autres fichiers physiques tels que référencés par le manifeste.

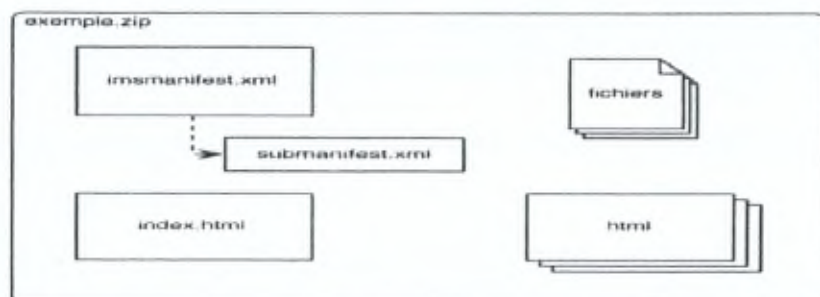


Figure 3.2 : Structure du paquet SCORM tirée de (ADL, 2001).



Un *Package Interchange File* est un format de livraison Web concis, et un moyen de transporter des informations liées entre elles et structurées. Un exemple est illustré dans la figure 3.2.

### 3.4.3 Bref aperçu du SCORM

Le SCORM (*Sharable Content Object Reference Models*, Modèle de référence de contenus objets partageables) est une émanation de l'initiative ADL (*Advanced Distributed Learning* : Apprentissage distribué avancé) du département de la défense américain (*Dod*) qui s'est donné pour mission de créer la possibilité de se servir des composants ou modules d'objets d'apprentissage dans des applications hétérogènes, indépendamment des outils utilisés pour les créer. Cette indépendance implique qu'un contenu ne devrait pas être lié au contexte et aux caractéristiques spécifiques du logiciel d'exécution, condition pour qu'il puisse être inclus dans des applications différentes. Pour être utilisé plusieurs fois, le contenu devrait avoir une interface et des métadonnées communes. Les critères du modèle SCORM sont la durabilité liée à l'indépendance du contenu par rapport aux versions successives, l'interopérabilité liée à l'usage sur divers systèmes, ainsi que l'accessibilité au contenu via des index.

C'est pourquoi dans le but d'atteindre cet objectif, l'ADL a imaginé une démarche principalement basée sur trois axes :

1. Création d'une classification, dont les différentes informations décrivent un parcours, qui se fait à l'aide de métadonnées de description des objets d'apprentissage, les *LOM metadata*. Ces métadonnées donnent, entre autres, des informations sur la langue d'enseignement, l'auteur, les pré-requis pédagogiques, et les exigences techniques.
2. Rendre simple la production des contenus grâce à une structuration prédéfinie des fichiers et indépendante des plates-formes e-learning de diffusion, que l'on appelle aussi LMS (*Learning Management System*). Ce rôle est assuré par le *packaging* ou *Content Aggregation Model (CAM)*. Concrètement, il s'agit d'une archive zip

contenant l'ensemble des fichiers du parcours, comme expliqué à la section 3.3.2.1.1.

3. Standardiser la récupération du parcours ainsi que de la progression des utilisateurs inscrits à une session de formation, quels que soient les choix technologiques pour l'hébergement. Ceci dénote une volonté d'indépendance vis-à-vis de toute plate-forme. Ainsi le SCORM passe la main au navigateur, au travers d'une interface de programmation ou API (*Application Programming Interface*) en Javascript nommée *Run Time Engine* ou *RTE*<sup>18</sup>. Cette API est chargée d'assurer la communication entre le contenu, le SCO et la plate-forme. Tous les échanges sont pilotés par le SCO suivant les volets suivants :

- initialisation de la session d'apprentissage ;
- récupération éventuelle des informations stockées lors d'une précédente session ;
- supervision des actions de l'apprenant ;
- envoi des informations issues de la session (temps passé, résultat d'un quiz, ouverture d'une page ou d'une ressource, images, PDF, vidéos, etc.) ;
- clôture de la session.

Le chapitre suivant sera consacré au SCORM, et nous reviendrons alors sur tous ces concepts en détails. Notons en attendant que le SCORM s'est imposé aujourd'hui comme le standard de facto de l'e-learning. Il est le premier qui, dans la démarche internationale de normalisation de l'e-learning, a apporté la couche informatique en fournissant les moyens techniques pour une interopérabilité concrète entre plates-formes.

## 3.5 Expérimentations d'e-learning à l'Université de Lubumbashi face à la problématique d'interopérabilité

### 3.5.1 Pourquoi une expérimentation à l'UNILU?

Plusieurs études et observations averties sont d'accord pour dire que l'enseignement universitaire et la recherche scientifique sont, de nos jours, au niveau le plus

<sup>18</sup> Le RTE est la partie de la norme SCORM qui décrit les échanges entre le contenu et la plate-forme d'accueil, nous y revenons en détails au chapitre 4.



bas en Afrique subsaharienne. L'apprentissage en ligne, surtout l'existence d'outils open source, est un concept qui peut incrémenter leurs niveaux et amener à améliorer l'enseignement supérieur africain en favorisant notamment une collaboration efficace, et à coût raisonnable, entre les universités européennes et celles d'Afrique. Cela dit, les enseignants des institutions européennes, qui n'ont pas forcément le temps d'assurer à plein temps des enseignements dans les universités africaines, peuvent, au travers des formations en ligne, avec l'aide d'un tuteur local, organiser de manière efficace des enseignements. Cependant on note que, malgré plusieurs tentatives de mise en œuvre de tels dispositifs, la plupart des projets finissent par être abandonnés pour différentes raisons. L'une des plus importantes est le manque d'adaptation de ces outils, et de leur implémentation au sein d'une plate-forme, au contexte local d'enseignement-apprentissage. Il est donc nécessaire de suivre de près des expérimentations de formation en ligne locale afin d'en tirer des comportements émergents quant à l'usage des outils pour déterminer les facteurs pouvant amener à « contextualiser » la formation en ligne à travers des observables. L'analyse du déroulement d'une expérimentation effectuée à l'Université de Lubumbashi a, par exemple, souligné un rôle très important du tuteur. Dès lors, notre démarche est celle qui consiste à créer un scénario d'apprentissage fidèle à la situation réelle sur le terrain. Nous avons caractérisé le déroulement de l'activité par un modèle, pour en tirer un prototype informatique selon un formalisme standardisé de manière à adapter l'outil de formation au contexte local.

### 3.5.2 Présentation générale des expérimentations

Les motivations de l'interopérabilité au niveau de l'Université de Lubumbashi sont, comme nous l'avons déjà susmentionné, les multiples avantages qu'une telle réalisation amènerait à la qualité de la recherche scientifique et de l'enseignement en augmentant le flux d'échanges entre l'UNILU et ces partenaires belges. Les enjeux sont à examiner à plusieurs niveaux ou plusieurs couches vu la complexité de la question. Nous évoquerons ici l'aspect technique, l'aspect technologique, l'aspect socioculturel ainsi que l'aspect pédagogique dans le volet du gain ou de la plus-value dans l'acquisition du savoir.

Nous avons réalisé à l'issue de l'enquête détaillée au premier chapitre du travail que la ville de Lubumbashi, par l'évolution de son profil informatique au cours de ces 4 dernières années, offrait une infrastructure informatique qui permettait à un étudiant de l'Université de Lubumbashi de gérer, pendant son cursus, quelques cours en ligne, en particulier pour les cours n'ayant pas des spécialistes locaux. Encouragé par l'enquête, et dans le but de nous rendre compte par nous même du déroulement effectif d'une session e-learning dans le contexte actuel de l'Université de Lubumbashi, nous avons initié pendant trois ans une expérimentation d'outil d'e-learning sur le terrain. Notre objectif était de déterminer le scénario d'apprentissage à l'import à partir d'une autre plate-forme. L'expérimentation de l'outil à l'Université de Lubumbashi a permis de définir un scénario qui donnerait l'ossature principale que devrait avoir un cours importé pour être conforme au contexte local d'apprentissage. Ce scénario voulait également, en amont d'une opération d'import d'un cours, déterminer l'ordre d'utilisation et la hiérarchie des priorités à accorder aux différents composants et services accompagnant ainsi le cours. Le développement du code de l'application d'import-export des cours dépendra en effet en partie de l'usage des outils dans le cadre d'un apprentissage et donc du scénario adapté au contexte local. Ceci suppose en outre que les cours importés n'auront pas nécessairement le même profil que dans l'application d'origine afin de se conformer au déroulement réel des cours dans le contexte e-learning de l'UNILU.

### 3.5.2.1 Objectifs de l'expérimentation

Nous partons de la conviction que, pour bien juger un outil dans un contexte précis, il faut le mettre à l'épreuve. Notre expérimentation grandeur nature visait principalement deux objectifs majeurs :

1. Une expérimentation du dispositif pour identifier des observables et des comportements émergents qui peuvent nous aider à cerner *les facteurs de contextualisation* dans l'usage des outils et services auxquels il faut veiller pour une adaptation de la plate-forme au contexte local d'enseignement-apprentissage.



2. La description à l'aide d'un scénario narratif ou d'une modélisation de ces *facteurs de contextualisation* des activités d'apprentissage et de tutorat en formation en ligne pour favoriser le développement d'outils adaptés et réellement utiles à la formation concernée.

### 3.5.2.2 Caractéristiques du contexte d'apprentissage

Nous avons choisi pour notre expérimentation deux facultés, à savoir la faculté polytechnique et la faculté de médecine humaine, et une école supérieure de l'Université de Lubumbashi, où deux cours<sup>19</sup> ont été donnés sous-forme d'enseignement en ligne. Nous avons au total près de 1000 étudiants pour la promotion de premier doctorat en médecine humaine, ainsi que près de 130 étudiants pour la faculté polytechnique et l'Ecole Supérieure des Ingénieurs Industriels (ESI), respectivement pour les promotions de deuxième graduat Electromécanique et deuxième graduat Informatique. Le groupe de médecine évoluait à part, tandis que ceux de polytechnique et ESI évoluaient dans une même classe. Les raisons éventuelles d'organisation, dues au nombre élevés d'étudiants en médecine nous pousserons plus tard à abandonner les travaux côté médecine<sup>20</sup> pour nous concentrer que sur Polytechnique et ESI. L'expérimentation s'est faite à la faculté Polytechnique et s'est inscrite dans le cadre du cursus académique normal de la formation d'ingénieur d'une période allant chaque fois d'Avril à Juillet 2007, 2008 et 2009, soit au second semestre pour environ quatre mois. Nous disposions à cette fin de deux salles d'informatique et un parc de 10 et cinq ordinateurs et deux Kits complets de matériels de projection.

### 3.5.2.3 Public cible (acteurs et groupes)

Partant de l'expérimentation de 2008, qui est celle qui s'est déroulée de manière plus acceptable (sans interruptions administratives ou techniques), le public était constitué de 68 étudiants de G2 ELM et 13 étudiants G2 Info/ESI que nous devions constituer en

<sup>19</sup> Deux cours car le cours d'Algorithmique 1 et Base de données, a été donné à la fois en Polytechnique et à l'Ecole Supérieure des Ingénieurs Industriels.

<sup>20</sup> En effet 1000 étudiants environ pour 15 ordinateurs seulement disponibles, cela rendait la tâche quasi impossible, car la plate-forme n'était accessible que par intranet de l'UNILU.

groupes pour favoriser les échanges (Johnson, et al., 1998). Nous avons ainsi constitué des groupes de huit à neuf étudiants pour G2 ELM et quatre à cinq étudiants pour G2 Info.

Dans le concept de formation en ligne on distingue généralement deux approches de qualification des types d'acteurs impliqués dans une formation (Paquette, 1998). La première stipule que dans une formation on a trois types d'acteurs (l'apprenant, le concepteur, et le tuteur). La deuxième stipule qu'il y en a plutôt cinq (l'apprenant, l'informateur, le concepteur, le formateur et le gestionnaire). Pour ce qui est de notre expérimentation, vu la singularité des conditions d'exécution du programme e-learning à l'UNILU et le fait qu'il s'agisse d'une nouveauté, nous avons distingué deux types d'acteurs : l'apprenant d'une part et le tuteur d'autre part, ce dernier jouant ici à la fois un rôle de concepteur et de tuteur.

#### 3.5.2.4 Domaine d'application

Dans les conditions actuelles de travail, le choix du domaine d'application était motivé par:

1. Le contexte local et la facilité d'accès à l'outil informatique.
2. La nécessité de s'inscrire dans le cursus académique normal des enseignements à l'Université de Lubumbashi pour bien s'imprégner de ses réalités et rendre effective l'adaptation des outils à implémenter tout en ne perturbant pas l'horaire normal des cours.

Les cours ainsi choisis furent les cours d'Algorithmique et Programmation pour G2 ELM et de Base de Données pour G2 Info. Notons par ailleurs que l'objectif pédagogique d'un tel choix de cours est de favoriser auprès de l'apprenant une acquisition des compétences en programmation, conception et administration des systèmes de gestion des bases de données relationnelles. La durée des expérimentations s'est étalée sur les années académiques 2007-2008 et 2008-2009 et nous a permis de relever, à partir des comportements émergents et de l'usage des outils, le design que nous pourrions donner au scénario d'importation des cours qu'il fallait adapter à notre plate-forme ainsi qu'au



contexte d'utilisation de cette dernière localement. Pour relever les résultats des expérimentations, nous avons cherché d'abord à relever les traces des activités des étudiants à partir d'outils de communication synchrones et asynchrones. Un questionnaire a ensuite été préparé pour avoir une juste appréhension de l'outil informatique qu'ont les étudiants avant et après son utilisation. Le tutorat que nous avons assuré nous a permis de relever les outils les plus utilisés en fin de chaque journée au cours de la session

### 3.5.2.5 Description de la tâche

La description de tâche que nous allons faire ci-dessous concerne principalement le cours de Base de données I, que nous avons choisi pour illustrer notre démarche. Dans le but d'acquérir les compétences préliminaires dans la conception et l'administration des systèmes de gestion des bases des données, nous avons, en plus des activités en présentiel, prévu les activités suivantes réparties en quatre phases:

- phase 1 : Définir la modélisation (travail de groupe) ;
- phase 2 : Identifier et étudier les concepts clés du langage SQL (travail de groupe) ;
- phase 3 : Créer sous EasyPhp/PhPMyAdmin une base de données "Musique Congolaise" (travail de groupe) ;
- phase 4 : Rédaction d'un rapport final sur le processus de création de la base de données "Musique Congolaise" (travail individuel).

Pour cette première expérience dans l'institution, on notera que les différentes étapes des phases de l'activité ont été présentées au fur et à mesure de l'évolution des apprenants, suivant le calendrier académique et le temps imparti aux apprenants pour accéder à la salle informatique (la plate-forme ne pouvant fonctionner à l'époque qu'en Intranet).

### 3.5.2.6 Dispositif technologique

Pour notre expérimentation, nous avons choisi la plate-forme Dokeos ([www.dokeos.com](http://www.dokeos.com)) dont l'adéquation à l'Université de Lubumbashi a été établie au cours

d'une étude par analyse multicritère basée sur trois axes : l'ergonomie, la technicité, la Pédagogie. Trois plates-formes open source candidates ont été comparées (Fyama, 2006). Son interface conviviale offre les outils d'apprentissage répertoriés dans le tableau 3.4.

Tableau 3.4 : Types d'outils offerts par Dokeos

| a) Outils de production | b) Outils d'interaction |
|-------------------------|-------------------------|
| Description             | Agenda                  |
| Parcours                | Forums                  |
| Tests                   | Utilisateurs            |
| Documents               | Discuter                |
| Liens                   | Groupes                 |
|                         | Travaux                 |
|                         | Partage des fichiers    |

### 3.5.2.7 Aspects méthodologiques

Nous avons par ailleurs au cours de cette expérimentation adopté un tutorat à la fois proactif et réactif (De Lièvre, et al., 2001). On a combiné des interventions du tuteur qui faisaient suite à une sollicitation d'un ou plusieurs apprenants (réactif), avec des interventions initiées par le tuteur lui-même sans aucune sollicitation de la part des apprenants (proactif). On a mélangé des séances de cours en présentiel et des séances médiatisées par la plate-forme (*blended learning*). La question posée était donc : comment au cours d'une activité d'apprentissage relever des indices observables, quantifiables ? Et



que pouvions-nous interpréter ensuite pour une modélisation fidèle au contexte réel d'enseignement-apprentissage ?

Pour décortiquer l'activité d'apprentissage, nous nous sommes appuyés sur les expériences des travaux de plusieurs chercheurs (Betbeder, 2003; Gounon, et al., 2004; Gounon, et al., 2005; Vantroys, 2003; Depover, et al., 2005; De Lièvre, et al., 2006). Les théories d'activité de Kuuti nous ont amené à comprendre que les échanges entre les trois acteurs principaux d'une formation en ligne, à savoir l'enseignant, le tuteur et les étudiants. Ils constituent des observables qui réifient le déroulement des activités d'apprentissage dont les analyses quantitative et qualitative (Paquette, 1998) peuvent dévoiler des comportements émergents dans la manière d'utiliser les outils médiatisant les échanges et amener à décrire enfin ce que nous appelons des "facteurs de contextualisation" par lesquels on va modéliser l'activité. Schématiquement cela donne:

1. La récolte et la sauvegarde quotidienne de tous les échanges entre acteurs via les outils et en dehors des outils.
2. La classification des messages selon leurs types détaillée à la section 3.4.3.1.
3. L'analyse quantitative et qualitative des messages.
4. La déduction des comportements émergents et de l'utilisation effective des outils.
5. L'énoncé des règles favorisant l'adaptation des outils au contexte local (*facteurs de contextualisation*).

### 3.5.3 Analyse de l'e-learning a l'Université de Lubumbashi

#### 3.5.3.1 Analyse des interactions

Le déroulement des enseignements s'est fait par des séances présentielle combinées aux séances en ligne, suivant un horaire prédéterminé par l'institution. Nous avons rencontré quelques difficultés concernant l'accès très limité en temps à la salle informatique de la faculté, ainsi que la connexion payante au sein de l'intranet de l'université. De plus, la plate-forme n'étant pas encore accessible via le Web nous devons

faire face à l'impossibilité de se replier sur des salles hors du domaine de l'UNILU. Mais, au delà de ces difficultés, la remarquable volonté des étudiants de s'approprier un outil visiblement moderne et à la mode a fait que les cours se sont déroulés dans des conditions acceptables que nous allons à présent analyser.

Pour analyser les interventions, on a commencé par une qualification de ces interventions en se basant sur les méthodes des chercheurs cités au point précédent pour les classer en messages d'organisation de l'activité, messages directement liés à l'activité et messages divers. Nous en faisons une illustration au tableau 3.5. Par ailleurs comme le recommande Glickman, il faut analyser les thèmes qui font l'objet des discussions afin que, à partir des contenus des interventions, on puisse les catégoriser (Glickman, 1999). Nous avons alors adopté la taxinomie suivante :

1° La catégorie « Organisation » : les interventions d'organisation concernent toutes les interventions relatives à l'organisation des tâches au sein d'un groupe, à la répartition des rôles au sein du groupe, à la gestion des différents rendez-vous et agenda du cursus.

2° La catégorie « Activité » : cette catégorie concerne les interventions centrées sur l'activité proprement dite. Elle comprend les questionnements, l'éveil du sens critique, le développement de l'argumentation, la construction du savoir.

3° La catégorie « Autres » : cette catégorie regroupe toutes les interventions relevant d'un aspect tout à fait social entre les membres d'un groupe et comprend la vie affective du groupe, la détente de l'atmosphère ainsi que le respect des règles de bonne conduite. Nous y avons aussi inséré tous les messages qui concernent les problèmes techniques, les coupures d'électricité, l'utilisation de certains outils, etc.

Ainsi après catégorisation des interventions, nous nous sommes aussi basés sur le recueil des observables tels que les questionnaires, productions des étudiants, traces informatiques (Betbeder, et al., 2003) pour nous aider à l'élaboration définitive des grilles de classification des interventions, dont nous donnons un exemple sur le tableau 3.5, ainsi qu'à l'analyse des interventions.



Tableau 3.5 : Exemples des messages relevés lors des expérimentations

| Organisation |                           | Exemples des messages  |
|--------------|---------------------------|--|
|              | Organisation sur le fond  | Jonathan tu t'occuperas de la table « Musiciens »                |
|              | Organisation sur la forme | Chacun crée sa table puis nous les mettons ensemble.             |
|              | Gérer le temps            | Le travail est à remettre avant vendredi.                        |
|              | Proposer une séance       | N'oublie pas de rappeler Dan et Elie pour qu'on travaille demain |
|              | Structure du groupe       | Elie tu dirige les débats aujourd'hui                            |
|              | Disponibilités            | Je ne serai pas là demain, je vais à Kipushi                     |
|              | D'accord ou pas           | C'est pas ça une modélisation.                                   |
| Activité     |                           | Exemples des messages  |
|              | Objet de l'activité       | Définir d'abord le concept de modélisation                       |
|              | Synthèse                  | Finalement on adopte toutes les listes.                          |
|              | Précision et Ajout        | Ouvre le cours BDD1, puis va sur « annonce ».                    |
| Autres       |                           | Exemples des messages  |
|              | Salutations               | Bonjour Jonathan   |
|              | Remerciements             | Merci tout le monde et à demain                                  |
|              | Félicitations             | Le test a été bien fait.   |
|              | Encouragements            | C'est un système génial!   |
|              | Privés                    | J'ai pas assez dormi cette nuit.                                 |

### 3.5.3.2 Utilisation des outils au cours des activités

Au vu des scores d'utilisation des outils, nous remarquons que c'est l'outil *liens* qui a le plus été utilisé au cours de l'apprentissage (527), vu la propension de nos étudiants à aller souvent sur Internet pour se documenter, suivi du *chat* (485). Le moins utilisé était *agenda* (36). Nous savons que les outils de communication synchrones et asynchrones sont déterminants pour l'apprentissage, en contribuant à structurer la pensée et à soutenir le développement de certaines démarches cognitives (Salomon, 1993). On observe en fait une préférence pour les chats et les forums, par rapport aux autres outils de communication qu'offre la plate-forme. Le chat a été cependant plus utilisé que le forum car le caractère synchrone de la communication par chat donnait, selon eux, un sentiment d'être en présence de la personne avec qui on échange. Ceci traduit le souci de la part des étudiants de se sentir toujours en communauté et d'éviter l'isolement, pourtant souvent constaté au cours d'un apprentissage en ligne. Ceci abonde dans le même sens que Daft et Lengel (Daft, et al., 1984) cités par (Charlier, 1998) qui reconnaissent, dans leur établissement des facteurs de *l'information richness*, que le chat est riche en *interactivité* et en *indicateurs socio-émotionnels*. Celui-ci est de plus porteur d'un aspect *social-motivationnel*, tel que l'indique pour sa part Candace (Candace Chou, 2002), qui donne à l'apprenant le sentiment d'être impliqué et accepté dans le groupe de travail. Les étudiants ont eu aussi recours, de temps en temps, pour communiquer, aux e-mails qui constituent une fonction non implémentée dans la plate-forme.

Nous avons noté par ailleurs qu'il existait un lien évident entre la nature de l'outil de communication utilisé et la forme de travail:

1. Le chat est lié à la collaboration directe, favorise le consensus et entretient une certaine convivialité au sein du groupe. Il a aussi vocation à véhiculer des échanges directs d'avis ainsi que la répartition de rôle et la distribution des tâches, ce que constate également Charlier<sup>21</sup> qui insiste pour sa part sur

---

<sup>21</sup> Dans le même ouvrage cité quelques lignes plus haut (Charlier, 1998).



la richesse de cet outil qui est globalement basée sur sa convivialité et son interactivité.

2. Le forum est lié à des échanges de documents entre apprenants et les aide à coopérer. Il a aussi été largement utilisé en remplacement de l'e-mail, et nous sommes à ce niveau sur la même longueur d'ondes que Henri et Lungren-Cayrol (Henri, et al., 2001) qui stipulent que dans le cadre des stratégies d'apprentissage l'intérêt principal du forum réside dans son rôle de distributeurs des contenus.

Tableau 3.6 : Score d'utilisation des outils.

| Usages des Outils    |         |         |         |       |
|----------------------|---------|---------|---------|-------|
| Outil                | Groupe1 | Groupe2 | Groupe3 | Total |
| Forum                | 23      | 24      | 59      | 106   |
| Discuter(Chat)       | 90      | 75      | 320     | 485   |
| Agenda               | 12      | 12      | 12      | 36    |
| Annonces             | 53      | 51      | 56      | 160   |
| Partage des fichiers | 87      | 133     | 203     | 423   |
| Documents            | 37      | 89      | 153     | 279   |
| Liens                | 172     | 169     | 186     | 527   |
| Tests                | 16      | 16      | 16      | 48    |

### 3.5.3.3 Analyse des échanges avec le tuteur par catégories des interventions

On notera pour commencer que le rôle du tuteur au cours de cette expérimentation était, comme on pouvait s'y attendre, essentiel à la bonne marche des enseignements. Le recours à l'assistance du tuteur de la part des étudiants était fréquent et il représente 76,4%

des échanges par forum et 47,8% par chat. La communication asynchrone révèle ici le recours constant au formateur pour toutes sortes de difficultés rencontrées, ce qui à coup sûr renseigne que, pour les étudiants de l'UNILU, l'usage des outils tout à fait nouveaux a nécessité de l'assistance. Ceci confirme que, en e-learning, l'accès à un tuteur humain conduit à un usage plus important des outils d'aide que son absence (De Lièvre, et al., 2001). Cette expérimentation corrobore Mason qui argue que la proactivité donnerait à l'apprenant le sentiment d'être suivi en le stimulant à rester en état de veille cognitive et à plus exploiter les aides mises à disposition (Mason, 1992). Au cours de notre expérimentation, le tuteur était parfois désigné par les apprenants affectueusement sous le sobriquet de « coach ».

Pour ce qui est des types d'échanges menés avec le tuteur, comme illustrés dans les tableaux 3.6, 3.7 et 3.8, les pourcentages très élevés touchent l'organisation (44,1%), alors que seulement 38,6% concernent l'activité et 17,3% les divers. Les échanges sur l'organisation qui sont majoritaires nous font découvrir un recours constant à l'autorité de l'enseignant en apprentissage académique classique et traduisent très peu d'indépendance vis-à-vis de celui-ci. Ceci est évidemment compréhensible car l'apprentissage en ligne est un fait nouveau pour l'étudiant de l'UNILU chez qui, comme le dit Rodet, l'enseignant joue le rôle principal en étant considéré comme possesseur de la connaissance qu'il doit transmettre à l'apprenant (Rodet, 2000). Les échanges sur la formation ont aussi été nombreux car, tout naturellement, les groupes sont guidés par l'enseignant-tuteur. Nous remarquerons aussi, en nous focalisant sur les performances des notes des étudiants obtenues à l'issue de la formation, que le groupe 3 s'est distingué des autres, notamment parce qu'il avait la meilleure note, mais aussi par ses pourcentages élevés d'échanges basés sur l'activité (47,6% par forum et 60,2% par chat) c'est-à-dire sur le contenu. Il a visiblement passé plus de temps que les autres au travail. Il est par ailleurs celui qui recours le moins à l'assistance du tuteur sur son organisation (vu ses pourcentages les plus faibles des trois), exprimant par là une certaine autonomie affichée par rapport aux autres groupes. Ses échanges de groupe dans la catégorie divers sont parmi les plus nombreux et atteignent leur maximum en mode synchrone ce qui nous laisse soupçonner une plus grande convivialité dans le groupe et beaucoup plus de socialisation en son sein. Nous rejoignons en cela la



littérature où les chercheurs qualifient la communication synchrone comme un bon outil pour soutenir les processus sociaux des groupes qui les apprécient pour apprendre à se connaître et à socialiser (Henri, et al., 2001).

Tableau 3.7 : Echanges avec le tuteur par catégorie via le forum, en pourcentages.

| Catégorie des interventions | % Groupe 1 | % Groupe 2 | % Groupe 3 |
|-----------------------------|------------|------------|------------|
| Organisation                | 47,4       | 55,0       | 40,5       |
| Activité                    | 42,1       | 35,0       | 47,6       |
| Autre                       | 10,5       | 10,0       | 11,9       |

Tableau 3.8 : Echanges avec le tuteur par catégorie via le Chat en pourcentages.

| Catégorie des interventions | % Groupe 1 | % Groupe 2 | % Groupe 3 |
|-----------------------------|------------|------------|------------|
| Organisation                | 35,3       | 25,9       | 24,0       |
| Activité                    | 50,0       | 55,6       | 60,2       |
| Autre                       | 14,7       | 18,5       | 15,8       |

Au cours de cette expérimentation, une attention particulière a été portée également sur les échanges hors activités, qualifiés de catégories autres (divers). Ils représentent entre 12 et 38 % des échanges en général et sont pour près de 70% dans les chats. Ceci prouve notre allégation qui stipule que les outils synchrones sont liés à la collaboration directe et favorisent le consensus en développant plus les liens sociaux

(convivialité, remerciements, complicité, félicitations, etc.). Les messages hors activité ont aussi exprimés des temps de flottement, de fatigue, d'attente, et de difficultés techniques.

Un fait marquant aussi de notre expérimentation est que, au vu des résultats des étudiants, il est apparu que la quantité des échanges synchrones était directement proportionnelle au rendement du groupe, le meilleur des groupes étant le 3 suivi du 2 puis du 1 qui était le dernier, tel que l'illustre la figure ci-dessous.

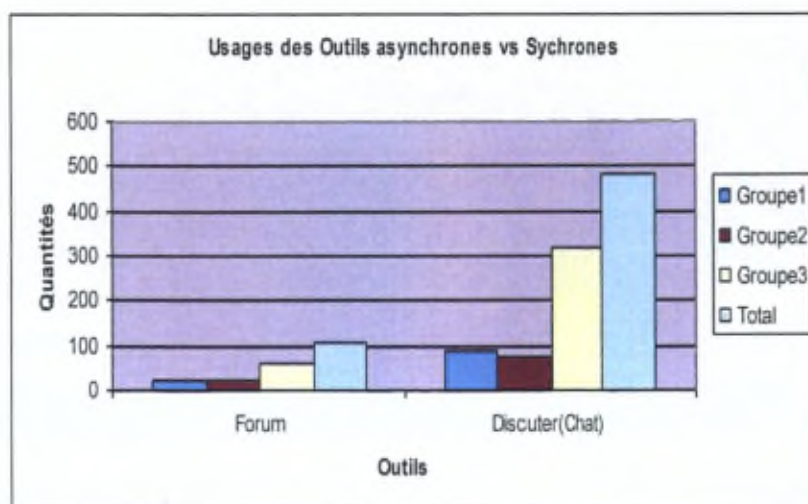


Figure 3.3 : Usages des outils asynchrones vs synchrones par les différents groupes.

Notons qu'une bonne partie des échanges n'était pas médiatisée par la plate-forme, cela était dû à l'usage de la même salle par des étudiants qui du reste était familier entre eux.

### 3.5.3.4 Bilan : Comportements émergents, usage effectif des outils et scénario narratif.

Après analyse des interventions au cours de nos expérimentations à l'Université de Lubumbashi, le scénario à mettre en œuvre afin de garantir un bon déroulement d'un apprentissage en e-learning devra être basé sur :

1. La répartition des étudiants en groupes afin de favoriser une organisation et une répartition des tâches dans une ambiance conviviale et social-motivationnelle.



2. Prévoir pour chaque groupe un forum et un chat thématiques qui, par leurs modes de communication synchrone et asynchrone, sont le gage d'un apprentissage collaboratif assuré.
3. Un outil des dépôts et de partage de fichiers et documents sous forme de collecticiel<sup>22</sup>.
4. Un outil d'annonces et un agenda pour des rappels des dates et des rendez-vous importants.
5. Un outil permettant d'accéder à des liens hypertextes utiles pour l'apprentissage.
6. Un outil de suivi des performances et du parcours de l'étudiant.
7. Inclure dans la plate-forme un service d'e-mail interne favorisant aussi certains types d'échanges privés auxquels nos étudiants avaient aussi recours.
8. Le souci des étudiants de voir physiquement l'enseignant qui assure le cours à distance, souci constaté par rapport aux nombreuses questions reçues de leur part, nous pousse à prévoir un outil de vidéoconférence qui rapprochera un peu plus, à notre avis, l'apprenant de son enseignant à distance.
9. Nous noterons aussi qu'une unité d'apprentissage devrait être subdivisée en plusieurs phases successives, chaque phase étant accompagnée des outils de communication synchrone et asynchrone.

### 3.6 Enjeux de l'interopérabilité a l'Université de Lubumbashi

Au cours de ce chapitre, nous avons réalisé un synopsis de tous les aspects de l'apprentissage en ligne à Lubumbashi en commençant par étudier au niveau international l'évolution des travaux de normalisation. Ensuite, nous avons réalisé un aperçu des concepts et standards émergents dans la démarche de normalisation en e-learning, et identifié dans la communauté les divers acteurs et chercheurs, leurs apports et leurs rôles. Nous nous sommes aussi penchés sur des expérimentations menées pendant trois années académiques des sessions des cours en ligne. Il s'agissait pour nous de nous imprégner du contexte, de palper du doigt la réalité de l'e-learning in situ et d'en dégager des règles ou de bonnes

---

<sup>22</sup> Un collecticiel est un logiciel qui permet un travail collectif, collaboratif et à distance afin de rassembler ainsi des groupes de personnes éloignées sur un projet commun.

pratiques pour favoriser la réussite de telles entreprises localement. Ceci a permis de dégager les grandes lignes d'un cahier des charges devant figurer dans un scénario pédagogique à posteriori pour prévoir, après importation des cours sur notre plate-forme, quel reformatage est nécessaire pour les adapter au contexte de l'UNILU et les dispenser de la manière la plus optimale possible. Ces deux axes d'études du chapitre présent nous ont conduits à mieux appréhender et à dégager les enjeux locaux de l'interopérabilité entre plates-formes, tant il est vrai que l'e-learning est d'un apport indéniable dans la recherche des solutions aux multiples problèmes auxquels sont confrontés l'enseignement supérieur et la recherche scientifique en République Démocratique du Congo en général, et à Lubumbashi en particulier. Les principaux enjeux sont :

1. Le désenclavement de l'UNILU, en permettant à un enseignant d'une institution partenaire belge d'assurer son cours et de suivre ses étudiants à distance est une plus-value dans la coopération entre l'UNILU et ces partenaires belges. En effet, la difficulté de temps et de distance se verra réduite lorsque le suivi se fera de manière automatique à distance et les enseignants du nord pourront alors assurer des enseignements en ligne pour l'UNILU qui en a besoin.
2. L'opportunité à la fois pour les enseignants et les étudiants de maîtriser encore plus les outils informatiques, gage d'une plus-value dans les compétences et la compétitivité sur les marchés à la fois du savoir et du travail.
3. L'incrémentation du niveau et la mise à jour des enseignements donnés à l'UNILU, permettant ainsi l'émergence locale des nouvelles compétences compétitives dans un cadre de la globalisation actuelle.
4. La mise au point de la recherche scientifique afin qu'avec des connaissances scientifiques mises à jour, l'université soit capable d'apporter à la société des solutions efficaces aux problèmes auxquels elle est confrontée avec des outils et des concepts nouveaux, puissants et modernes.
5. Notre étude vise aussi dans un des aspects à essayer au maximum d'adapter les nouveaux outils informatiques au contexte local de l'enseignement, afin de tenir compte tant que faire se peut de la culture locale dans l'apprentissage. Comme le dit



si bien Arnaud : « *Sans occulter les disparités géo-économiques d'accès aux réseaux, le Web, pour devenir un espace commun partagé par le plus grand nombre, doit intégrer l'avis de représentants d'autres régions du monde tout en prenant en compte des critères de plurilinguisme.* » (Arnaud, 2002).

6. L'UNILU devrait pour sa part réfléchir sur les moyens de mise en place au sein du service informatique des compétences nécessaires afin de répondre à des demandes urgentes de mise en ligne des contenus.

## CHAPITRE 4. ETUDE DESCRIPTIVE DU STANDARD SCORM

### 4.1 Introduction

L'aperçu à travers la littérature et les différentes recherches en matière de normalisation en e-learning, dont l'un des objectifs affichés est d'atteindre l'interopérabilité des contenus d'apprentissage, laissent apparaître qu'une position dominante occupée par des spécifications ou des standards peut leur donner de facto un statut de norme établie avant même la ratification officielle par les organismes internationaux attitrés. Ainsi nous reconnaissons la prédominance du standard SCORM qui s'impose au niveau de la normalisation, et nous nous proposons de ce fait, de mener, au cours de ce chapitre, une étude approfondie du standard SCORM afin d'en comprendre la philosophie, les concepts, les objectifs, ainsi que les outils techniques d'implémentation. Notre intérêt pour SCORM s'aiguise encore dans la mesure où dans la pratique de l'e-learning à l'Université de Lubumbashi, les outils techniques dont nous disposons, à savoir les plates-formes open source Dokeos et Moodle, implémentent SCORM au moins en partie (comme nous le verrons au chapitre prochain). Nous sommes convaincus de surcroît que le cas d'étude que nous traiterons dans la présente thèse devra passer par SCORM pour que l'application à développer soit au maximum réutilisable. En outre, les travaux menés actuellement au sein du consortium montrent des perspectives de convergences encourageantes vers SCORM puisqu'il est notamment compatible avec différentes normes et standards promus par l'IMS *Global Learning Consortium*.

Au cours de ce chapitre nous noterons qu'il existe très peu d'études analytiques qui concernent le standard SCORM, au-delà du site officiel de l'ADL qui établit la norme. La plupart de nos éléments de descriptions s'inspirent donc de la documentation disponible sur le site officiel d'ADL<sup>23</sup>. Nous nous appuierons aussi sur un exemple d'un cours fictif, d'un cours de Physique avec ses modules, pour illustrer les trois concepts du modèle SCORM. Le cours est schématiquement représenté sur la figure 4.1.

---

<sup>23</sup> [www.adlnet.org](http://www.adlnet.org) (consulté en mai 2011)



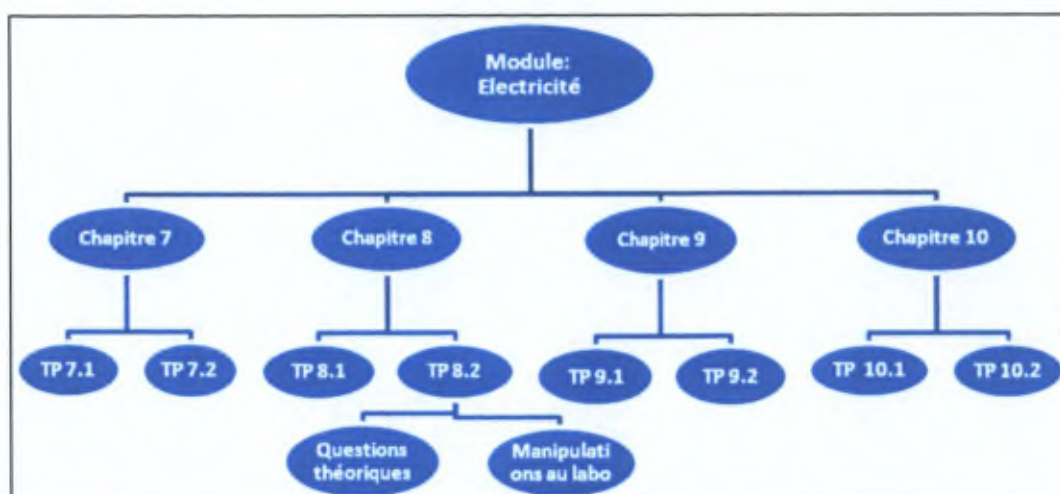


Figure 4.1 : Exemple d'un extrait de cours de Physique divisé en 10 modules.

## 4.2 Historique de SCORM

Nous sommes en novembre 1997 lorsque le bureau de la politique des sciences et techniques de la Maison Blanche et le département de la Défense (DoD) des Etats-Unis réunissent des industriels et des universités pour cogiter autour d'un projet de description des contenus des ressources d'apprentissage. L'initiative ADL (*Advanced Distributed Learning* : Apprentissage Distribué Avancé) est ainsi née et prévoit la création de bibliothèques de savoirs ou viviers de connaissances, où les objets d'apprentissage sont accumulés et catalogués pour une distribution et un usage à grande échelle. Ces objets doivent être facilement accessibles par le Web. Ces objets d'apprentissage seront donc accessibles, partageables et capables de s'adapter à la demande d'apprentissage des utilisateurs. Il s'agissait aussi, à l'échelle du DoD, d'inventer une stratégie d'utilisation des technologies de l'apprentissage et de l'information afin de moderniser les études et la formation, de promouvoir la coopération entre l'Administration, le monde universitaire et les entreprises dans le but d'assurer la normalisation de l'e-learning. L'initiative ADL a contribué à ce que le contenu soit séparé des contraintes liées au contexte et aux spécificités du logiciel d'exécution, de telle sorte qu'il puisse être inclus dans d'autres applications. De même, pour que son usage répété soit possible sous diverses formes, le contenu doit avoir une interface et des métadonnées communes à tous les développeurs et concepteurs. Le SCORM (*Sharable Content Object Reference Model*) est l'une des principales actions de ADL

pour répondre à la demande d'interopérabilité des contenus d'apprentissage (Friesen, 2004).

SCORM est l'une des normes qui constituent un premier pas important vers la libéralisation des objets (contenus) pédagogiques à l'égard des réalisations locales et des contextes hétérogènes. Elle a pour but de fournir les moyens techniques permettant aux objets de contenu d'être facilement partagés dans des environnements de prestation d'apprentissage multiples (Fage, 2005). Le modèle se veut aussi pédagogiquement neutre dans le sens où il permet une description d'un contenu indépendamment de son contexte. Ceci dit, le débat est ouvert car une bonne partie des pédagogues lui conteste à ce stade ladite neutralité pédagogique.

## 4.3 Les raisons du succès de SCORM

### 4.3.1 Positionnement par rapport aux autres acteurs

La normalisation en e-learning a pour objectif principal de permettre que n'importe quel contenu d'apprentissage (cours et exercices) soit capable de communiquer avec n'importe quelle plate-forme ou tout autre système de gestion des contenus et des apprenants. Les normes tentent d'apporter une réponse à cette problématique afin de créer un mode commun de communication des contenus et contenants d'apprentissage. Comme nous l'avons souligné dans les chapitres précédents, SCORM n'est pas le seul acteur dans la normalisation. Nous y trouvons également des acteurs majeurs comme l'AICC, l'IMS et le LOM pour ne s'arrêter qu'à ceux là. Que font ces autres acteurs? Et pourquoi SCORM tend à les supplanter sur le marché?

1. AICC est plus vieux que SCORM, et dans AICC, le contenu est considéré comme monobloc et sans granularité. Les liens entre les éléments ne sont pas clairement déterminés. « *Le modèle pédagogique est basé sur l'apprentissage procédural, consistant à présenter le cours, les questions à choix multiple avec validation, selon une approche behaviouriste et linéaire* »<sup>24</sup>.

<sup>24</sup> <http://www.educnet.education.fr>: site du ministère français de l'éducation (consulté en janvier 2011)



2. IMS s'intéresse à l'interopérabilité des applications et services en ligne notamment grâce aux métadonnées. Il implique, comme nous l'avons déjà souligné plus haut dans ce travail, plusieurs spécifications traitant de l'agrégation de contenus et de leur description. Il s'intéresse aussi au profil de l'apprenant afin d'individualiser le plus possible l'apprentissage. Il s'intéresse également à la problématique du questionnement ainsi que de la modélisation des scénarios pédagogiques.
3. LOM est la norme la plus complète en matière de métadonnées, et la plus connue dans les milieux universitaires et administratifs gouvernementaux. Elle est proche des problématiques de gestion documentaire puisqu'elle résulte des travaux du Dublin Core, mais elle s'est cependant focalisée sur la description des concepts de l'enseignement.
4. SCORM pour sa part est plutôt un modèle qu'une norme, dans le sens où il n'impose rien. Le défi que tente de relever SCORM est de fédérer en un seul modèle les concepts de tous les autres acteurs de la normalisation en essayant de synthétiser les différents systèmes passés (AICC) ou présents (IMS, LOM). Et nous pensons, au vu de l'évolution des choses dans la normalisation en e-learning, que l'une des raisons principales du succès de SCORM est la mission que se donne l'ADL non pas d'obliger une certaine conformité à certains paradigmes rigides et prédéfinis, mais de créer un modèle qui essaye de rassembler tous les travaux existants en matière de normalisation en e-learning de sorte à libérer les contenus des contextes à la fois logiciels et pédagogiques dans lesquels ils ont été conçus. Friesen précise qu'un des axes majeurs du succès de SCORM est aussi de prendre en compte l'aspect pédagogique dans la création de contenus. Cette intégration simultanée, au-delà de l'aspect informatique, de la pertinence pédagogique et de la neutralité pédagogique lui ont donné un succès grandissant dans le monde de l'enseignement (Friesen, 2004). En définitive, SCORM rallie donc des propositions techniques à des aspects pédagogiques, et apporte la couche informatique aux travaux de description d'objets d'apprentissage entrepris avec le LOM. Ceci explique la tendance de plusieurs logiciels à lui être conformes.

### 4.3.2 Avantages de SCORM

Au-delà du fait que SCORM fait plutôt office de « successeur » d'AICC dans le sens qu'il est né après lui, les deux standards demeurent les plus répandus en matière d'apprentissage en ligne. Il va de soi que pour donner les avantages de SCORM, nous ne pouvons les mesurer qu'à l'aune de son rival principal sur le marché à savoir AICC. Ainsi Fage se penche sur la question en détectant quatre avantages fonctionnels majeurs de SCORM sur AICC, notamment (Fage, 2005):

1. *« SCORM s'appuie sur le langage XML, largement utilisé sur le Web et conforme aux standards du W3C qui définissent le Web ; alors que AICC a été au départ développé pour des contenus sur CD ROM. XML est largement utilisé au niveau mondial, en dehors du secteur enseignement/formation, ce qui garantit sa pérennité et son évolutivité.*
2. *SCORM se base sur un système de métadonnées. Les ressources SCORM sont identifiées par des mots clés, une description, une structure interne, etc., qui serviront à faire le lien vers des LMS<sup>25</sup> et, plus largement, vers tout système ayant des fonctions de gestion documentaire (recherche, indexation, ...). Ce système de métadonnées est particulièrement intéressant et est utilisé par plusieurs projets de recherche et normes (LOM, IMS, Dublin Core, etc.), que ce soit en éducation/formation comme en gestion documentaire.*
3. *SCORM propose de travailler sur des « SCO ». Les SCO échangent des informations avec la plate-forme de suivi si cette dernière est également SCORM. A noter que chaque grain ou SCO est lui-même identifié par ses métadonnées. Ces grains peuvent être composés eux mêmes de plusieurs exercices et pages de cours. On travaille donc sur un cours comme un séquençement d'unités et non comme un bloc, ce qui améliore nettement le suivi et les possibilités d'individualisation.*
4. *SCORM permet également de définir la structuration des grains en un cours. Pour être plus précis, un fichier précis (imsmanifest.xml) détaille les métadonnées transmises à la plate-forme, l'organisation du cours (table des matières du cours et le mode de*

<sup>25</sup> Learning Management System appellation anglaise des plates-formes d'apprentissage en ligne.



*navigation) et la banque de ressources utilisées (ressources internes type sons et images, comme les liens vers les ressources externes). »*

En conclusion, le fait que SCORM pense à subdiviser le contenu selon une granularité des SCO capables d'échanger entre eux et d'être dotés d'une description riche et détaillée, il procède ainsi par systèmes modulaires et non monobloc comme c'est le cas chez AICC. Il possède l'avantage de permettre la création des formations en identifiant précisément les ressources et leur objectif, selon des champs d'information définis et communs. Ce qui tout naturellement favorise la réutilisation des ressources et leur administration. En comparant AICC et SCORM, les données fournies par SCORM sont donc plus riches et plus intéressantes, autant pour l'apprenant que pour le tuteur et le gestionnaire. Aussi un contenu AICC est plus orienté CD ROM tandis qu'un contenu SCORM utilise les standards les plus récents, entre autres XML et les métadonnées, et cela avec une granularité qui permet une utilisation du suivi des apprenants plus fine et plus personnalisable. SCORM va encore plus loin en intégrant la majorité des informations de description associées par AICC à un contenu comme le titre de la ressource, un score et la durée de consultation.

Le site francophone consacré à SCORM <sup>26</sup> révèle quant à lui que SCORM est plus moderne, dans le sens qu'il est adapté au Web, et que son identification des cours et des grains des contenus SCO par des métadonnées les rend réutilisables, partageables et d'une exploitabilité riche en informations. Le même site abonde dans le même sens en établissant un distinguo entre les avantages offerts par SCORM d'une part, à l'apprenant et son enseignant, et d'autre part à l'informaticien. En effet, l'informaticien voit son travail devenir plus complexe pour satisfaire aux exigences de SCORM. En contre partie l'enseignant jouit du fait que SCORM assure un suivi très détaillé de l'apprenant et l'apprenant se voit offrir une plate-forme nettement plus ergonomique. Par ailleurs, l'observation des sites des consortiums travaillant sur les normes laisse transparaître clairement qu'AICC n'évolue presque plus et est restée essentiellement technique. À l'inverse les acteurs de la norme

---

<sup>26</sup> <http://www.scorm.fr>: site francophone dédié au SCORM (consulté en janvier 2011)

SCORM sont très actifs: c'est un monde qui bouge et qui réfléchit beaucoup aux aspects pédagogiques.

Bref, SCORM englobe l'AICC en reprenant la totalité de ses informations tout en y ajoutant d'autres. Il est proche du LOM sur lequel il est fondé et qui vient du monde de la gestion documentaire tout en étant plus pragmatique. Enfin SCORM, en prenant en compte les aspects pédagogiques, se veut proche d'IMS dont il intègre certaines spécifications. Nous nous permettons de conclure, au vu de tout ce qui précède, qu'une interopérabilité basée sur SCORM est plus stable, plus évolutive, plus pérenne tout en étant pleine d'avenir.

## 4.4 Objectifs de SCORM

A partir de la documentation officielle de l'initiative ADL, on peut se rendre effectivement compte de l'hyperactivité de ses membres et réseaux avec un souci affiché d'atteindre une normalisation à l'échelle des Etats-Unis, puis à l'échelle continentale américaine et enfin à l'échelle internationale. L'organisation des activités au sein d'ADL nous a laissé percevoir qu'ils sont motivés par un souci permanent de faire que SCORM soit adopté, compris et utilisé par la plus grande variété d'intervenants ainsi que le plus grand nombre possible. Nous avons donc cerné les objectifs d'ADL sur deux axes principaux. Le premier axe est constitué par ce qu'ils appellent les « laboratoires » qui constituent un réseau d'agences pour répandre et expliquer le plus possible l'initiative. Le deuxième axe est SCORM lui-même qui se veut être un modèle de référence qui intègre le plus possible tous les autres modèles existant en e-learning et aboutir ainsi à un modèle unique.

### 4.4.1 Les laboratoires

Suite à une politique qui visait, à l'échelle des Etats-Unis, à élaborer des normes et des spécifications communes liées à l'apprentissage basé sur les nouvelles technologies, le Département de la défense a créé en 1999 à Alexandria, le Laboratoire *Advanced Distributed Learning(ADL) Alexandria* qui avait pour rôle de soutien technique et de lieu d'échange de plusieurs acteurs pour l'évaluation de l'état d'avancement des travaux destinés au projet ADL. Il constituait un lieu d'accueil pour les organismes fédéraux parrains et les gestionnaires de projet.



Pour stimuler les travaux et bien cibler les acteurs à atteindre par l'ADL, trois nœuds du Colaboratoire ont vu le jour ensuite. Premièrement le *Colaboratoire ADL conjoint d'Orlando* qui visait les militaires et divers services de l'armée, deuxièmement le *Colaboratoire ADL universitaire de Madison* qui avait pour cible les milieux universitaires, enfin troisièmement le *Colaboratoire ADL des travailleurs à Memphis* qui concerne les entreprises et les industries.

Ces colaboratoires ont pour mission principale, chacun dans son contexte, de promouvoir et de faire connaître les technologies d'ADL et de collaborer dans le but de partager les résultats des recherches, l'expertise, les outils communs et les contenus d'apprentissage via leur réseau. Les Colaboratoires sont chargés en outre d'évaluer la conformité des divers produit au modèle SCORM. Ces évaluations concernent :

- La capacité de transférer le contenu Web d'un environnement d'apprentissage à un autre ;
- La réutilisation des contenus d'apprentissage d'une plate-forme à une autre ;
- La création des contenus d'apprentissage dans lesquels on peut effectuer des recherches et qui sont repérables quel que soit l'environnement dans lequel ils se trouvent ;
- La création d'outils servant à la production et à l'utilisation de contenus d'apprentissage conformes au SCORM.

## 4.4.2 Le modèle SCORM

Le modèle SCORM est un modèle de référence et, en tant que tel, il a pour objectif de fournir à l'e-learning un ensemble de pratiques normalisées qui peuvent être acceptées généralement et leurs mises en œuvre faites sur une vaste échelle. Pour cela, il subdivise sa mission en trois tâches. Premièrement au niveau technique, il conçoit des lignes directrices de sorte qu'elles soient comprises et mises en œuvre par les développeurs de contenus. Deuxièmement, il veut ratisser large en se faisant adopter, comprendre et utiliser par le plus grand nombre possible d'intervenants. Troisièmement, il se veut fédérateur car il établit une correspondance entre lui-même et n'importe quel autre modèle venu d'un autre

concepteur. Pour atteindre ces visées, SCORM se dote des exigences de haut niveau que l'ADL nomme les « capacités » à appliquer aux contenus d'apprentissage. Et ces capacités sont :

- abordabilité : capacité à augmenter l'effcience et la productivité en réduisant le temps et les coûts nécessaires pour dispenser la formation ;
- accessibilité : capacité à repérer des composants d'enseignement à partir d'un site distant, d'y accéder et de les distribuer à plusieurs autres sites ;
- adaptabilité : capacité à personnaliser la formation en fonction des besoins des personnes et organisations ;
- durabilité : capacité à résister à l'évolution et aux changements de la technologie sans avoir recours de nouveau à la conception, à la configuration, et au codage qui sont coûteux ;
- interopérabilité : capacité à utiliser, dans un autre emplacement et avec un autre ensemble d'outils ou sur une autre plate-forme, des composants d'enseignement développés dans un site ;
- réutilisabilité : souplesse permettant d'intégrer des composants d'enseignement dans des contextes et des applications multiples.

Et pour finir, outre ces capacités, SCORM définit un autre concept général qui est « *l'hypothèse sur le Web* », qui affirme que le Web constitue la meilleure occasion de maximiser l'accès au contenu d'apprentissage et sa réutilisation.

Ainsi se basant à la fois sur les capacités et sur l'hypothèse sur le Web, SCORM compte offrir à ceux qui l'adoptent les assurances suivantes :

1. La capacité d'un système basé sur le Web à lancer un contenu développé au moyen d'outils provenant de différents fournisseurs et à échanger des données avec ce contenu.
2. La capacité des produits et des systèmes basés sur le Web provenant de différents fournisseurs à lancer le même contenu et à échanger des données avec ce contenu lors de l'exécution.



3. La capacité de plusieurs produits ou environnements basés sur le Web à accéder à un référentiel commun de contenus exécutables et à lancer de tels contenus.

## 4.5 Concepts généraux de SCORM

Les concepts principaux du modèle SCORM reposent sur une sorte de trilogie qui est la conséquence des différentes missions qu'il s'est donné pour former des contenus qui peuvent s'intégrer à toutes sortes de système. Il faudrait que ces contenus aient un assemblage et une forme de présentation qui répond aux mêmes principes de regroupements de ces unités d'apprentissage, et qu'ils soient capables d'établir un dialogue avec le système hôte via une interface dans un environnement précis. Enfin, étant question des contenus d'apprentissage, chaque grain devrait pouvoir donner des informations précises sur la succession des étapes à suivre (activités, tâches, rôles, etc.) durant l'apprentissage pour acquérir la connaissance ou la compétence visée. Ainsi, pour illustrer les travaux effectués à travers les spécifications et standards qui le constituent, SCORM prévoit une série de documents techniques, qui sont regroupés en trois principaux sujets conformément à ses ambitions de normaliser le secteur du e-learning :

- le « modèle d'agrégation du contenu » (*Content Aggregation Model*) ;
- l'« environnement d'exécution » (*Run-Time Environment*) ;
- le « séquençement et la navigation » (*Sequencing and Navigation*).

La description qui va suivre durant cette section est totalement inspirée du site officiel de l'ADL qui met à disposition une documentation unique sur le SCORM<sup>27</sup>, l'étude s'appuiera aussi sur un exemple de cours pour illustrer les concepts.

### 4.5.1 Modèle d'agrégation du contenu

Le modèle d'agrégation (*Content Aggregation Model, CAM*) du contenu SCORM constitue un moyen, neutre sur le plan pédagogique, qui permet aux responsables de la conception et de la mise en œuvre de la formation de regrouper les ressources appropriées

---

<sup>27</sup> [www.adlnet.org](http://www.adlnet.org): site officiel de l'initiative ADL (consulté en mai 2011)

dans le but d'offrir un parcours individualisé de formation. Une ressource d'apprentissage consiste en toute représentation de l'information utilisée dans le cadre d'un parcours.

Les parcours de formation sont constitués d'activités supportées par des ressources d'apprentissage électroniques et non électroniques.

Le modèle de contenu SCORM décrit les composants du modèle de référence SCORM utilisés pour créer un parcours d'apprentissage à partir de ressources d'apprentissage réutilisables et partageables. Le modèle de contenu précise également comment ces ressources d'apprentissage partageables et réutilisables, de niveau inférieur, sont agrégés et organisés dans les unités d'apprentissage de niveau élevé d'instruction. Le modèle de contenu de SCORM est composé de *Sharable Content Objects* (SCOs), d'activités, de l'organisation et de l'agrégation des contenus.

#### 4.5.1.1 Actif (Asset)

Sous sa forme la plus élémentaire, le contenu d'apprentissage se compose d'actifs (Assets), c'est-à-dire de représentations électroniques de médias, de textes, d'images, de séquences sonores, de pages Web, d'objets d'évaluation ou d'autres éléments d'information qui peuvent être envoyés à un poste client et exploités par un apprenant, comme illustré à la figure 4.2. Si nous prenons un exemple d'un cours de « Physique générale » de première année qui a 11 chapitres (fichiers PDF) pour sa partie théorique, rythmée de 22 Travaux Pratiques (fichiers Word) à raison de deux travaux par chapitre théorique (voir figure 4.1 et 4.10). Nous supposons que l'enseignant diffuse son cours sur base d'un résumé dans une présentation Powerpoint et de quelques vidéos d'illustration. Chaque fichier PDF, Word, Powerpoint et vidéo constitue un actif (asset) du cours, c'est le grain le plus petit et indivisible qui participe au cours.



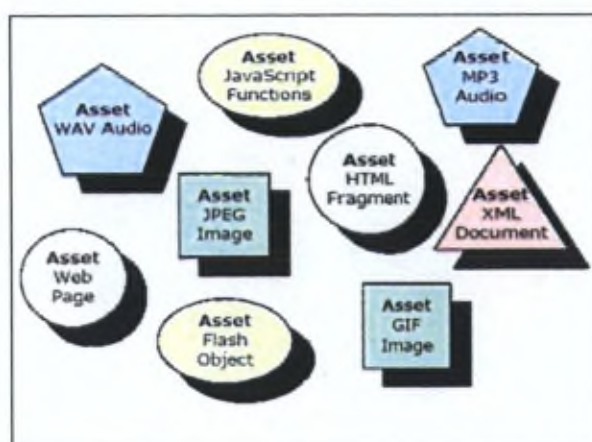


Figure 4.2 : Exemple d'un Asset extrait de (ADL, 2001).

#### 4.5.1.2 Sharable content Object (SCO)

Un objet de contenu partageable, *Sharable Content Object*, ou SCO, est un ensemble comprenant un ou plusieurs actifs (Assets), y compris un actif spécifique<sup>28</sup> qui utilise l'environnement d'exécution du SCORM pour communiquer avec des logiciels de gestion pédagogique ou Learning Management System (LMS) c'est-à-dire plate-forme d'apprentissage (figure 4.3). Le SCO peut être suivi par une plate-forme dans l'environnement d'exécution du SCORM.

Pour pouvoir être réutilisé, le SCO lui-même doit être indépendant du contexte d'apprentissage. Les SCO sont conçus pour être des unités subjectivement petites, de façon à pouvoir être réutilisées dans un grand nombre d'objectifs d'apprentissage.

Revenons à l'exemple du cours de la section 4.5.1.1 et supposons maintenant que le cours est constitué de trois modules. Le premier module est « Mécanique » constitué des chapitres 1 jusqu'au sixième chapitre, Le deuxième module s'appelle « Electricité » constitué des chapitres 7 jusqu'au dixième, enfin le troisième est « Optique » et ne comporte que le onzième chapitre. Prenons maintenant le premier module et supposons qu'il soit à son tour divisé en trois parties dans l'ordre « Statique », « Cinématique », « Dynamique ». Chacune

<sup>28</sup> Un actif est toujours prévu, comme nous le verrons plus loin, pour entamer le dialogue avec la plate-forme d'accueil du contenu.

de ces parties comporte deux chapitres accompagnés de deux travaux pratiques donc l'ensemble est constitué de 18 fichiers.

Pour que ce module constitue un SCO au sens de SCORM, il faut lui ajouter un dix-neuvième fichier au format JavaScript qui sera chargé de fournir à la plate-forme les informations suivantes : que les six chapitres constituent un module (ou un cursus), que le premier chapitre est le chapitre 1, que chaque chapitre a pour pré-requis le chapitre précédent, que chaque chapitre est suivi d'une évaluation et un apprenant n'accède au chapitre suivant que s'il a réussi au test, etc.

Un SCO peut être décrit au moyen de « métadonnées de SCO » pour faciliter la recherche et le repérage dans les dépôts de données en ligne, ce qui a pour conséquence d'accroître les possibilités de réutilisation. La nécessité pour un SCO de participer à l'environnement d'exécution du SCORM offre les avantages suivants :

- toute plate-forme supportant l'environnement d'exécution du SCORM peut lancer des SCO et en assurer le suivi, peu importe qui les a générés ;
- tout plate-forme supportant l'environnement d'exécution du SCORM peut assurer le suivi d'un SCO et en connaître les dates de début et de fin ;
- toute plate-forme supportant l'environnement d'exécution du SCORM peut lancer n'importe quel SCO de la même façon.

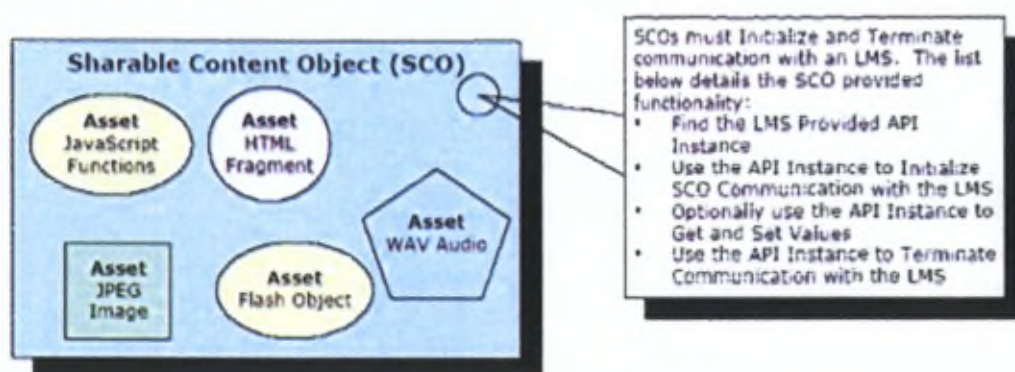


Figure 4.3 : SCO constitué de plusieurs actifs extrait de (ADL, 2001)



### 4.5.1.3 Activités

Une activité d'apprentissage peut être décrite comme une unité significative d'instruction. Elle est conceptuellement quelque chose que l'apprenant fait tout au long de son évolution lors de l'apprentissage (par exemple, les séances de cours ex cathedra, les séances des TP, etc.). Une activité d'apprentissage peut fournir une ressource d'apprentissage (SCO ou Actif) à l'apprenant et elle peut en outre être composée de plusieurs sous-activités.

Les activités représentées dans l'organisation d'un contenu peuvent consister en d'autres activités ou des sous-activités qui peuvent à leur tour être constituées d'autres activités. Il n'y a aucune limite du nombre des niveaux d'emboîtement pour les activités. On peut cependant associer une taxonomie d'étude spécifique aux niveaux hiérarchiques d'activités, par exemple, cours, chapitre, module, etc., sans que cela soit obligatoire. Les activités qui ne sont pas composées d'autres activités auront simplement une ressource d'apprentissage (SCO ou Asset) à utiliser pour les exécuter. Partant de l'exemple du cours à la section 4.5.1.2 « lire le chapitre 1 » est une activité qui n'utilise que le seul fichier PDF dédié au chapitre 1. Une activité composée d'autres activités se nomme cluster comme nous le verrons un peu plus loin (voir section 4.5.3.2).

### 4.5.1.4 Organisation de contenu

Une organisation de contenu montre comment le package est organisé à l'utilisation, ce qui permet de représenter la structure des contenus. Elle permet d'établir une hiérarchie ainsi qu'une correspondance qui peut être utilisée pour regrouper les ressources d'apprentissage en une unité d'enseignement cohérente (par exemple : une unité modulaire de formation, un module, un sous-module, etc.).

L'organisation du contenu ne devrait pas être confondue avec la structure physique du content package, ou avec la structure du manifeste lui-même (voir figure 4.4). Les fichiers dans le *content package* sont, par exemple, souvent organisés dans une hiérarchie de dossiers, mais cette structure en soi ne peut pas révéler à l'utilisateur d'un contenu

comment l'utiliser. Les informations de séquençement sont donc associées à chaque activité. La plate-forme est responsable de l'interprétation de ces informations décrites dans l'organisation du contenu ainsi que du contrôle de l'exécution du séquençement des ressources d'apprentissage.

Cette approche marque un changement important par rapport au développement traditionnel des didacticiels. Dans presque tous les cas, les systèmes ou les logiciels auteurs définissaient ou appliquaient des méthodes de séquençement propriétaires et souvent exclusives. En conséquence, il est encore aujourd'hui difficile, ou même impossible, de partager des contenus créés au moyen de systèmes auteurs différents. Il est également difficile de réutiliser ces contenus dans d'autres contextes avec des exigences de séquençement spécifiques.

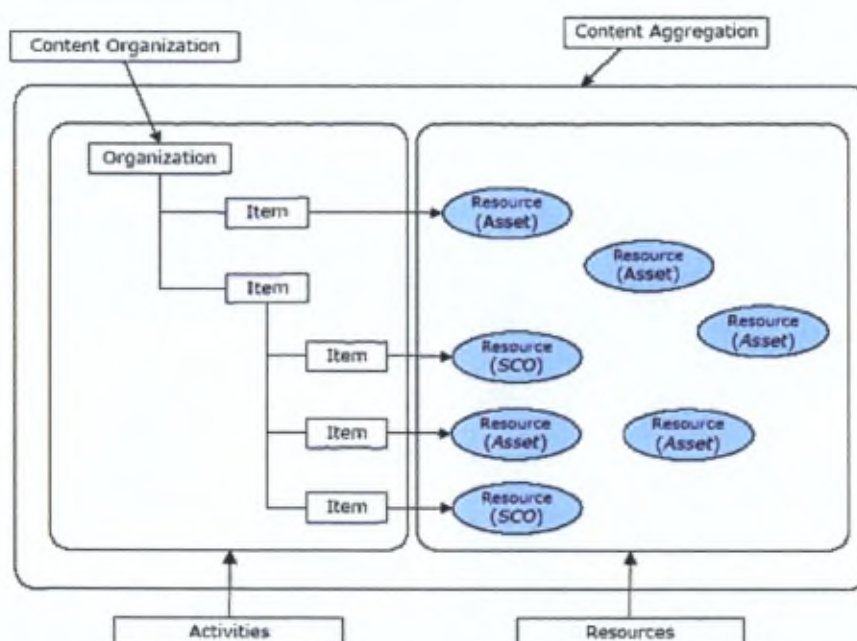


Figure 4.4 : Organisation de contenu extrait de (ADL, 2001).

Dans SCORM, le séquençement des ressources d'apprentissage est défini dans la structure du contenu et donc indépendamment de la ressource d'apprentissage elle-même. C'est à la plate-forme qu'il appartient de lancer les ressources d'apprentissage selon la séquence appropriée définie. Cette approche est fondamentale sur le plan conceptuel



puisque la réutilisation des ressources d'apprentissage est effectivement irréalisable si ces ressources comportent de l'information intégrée spécifique au contexte de la formation.

#### 4.5.1.5 Représentation de la structure d'un contenu

Une organisation d'un contenu SCORM inclut tous les composants qui ont pour mission de définir différents aspects de la structure d'un contenu. Ils s'articulent sur la hiérarchie au sein du contenu, sur les métadonnées de description, et enfin sur le séquencement dans la navigation au sein du contenu. On retrouve :

1. La hiérarchie du contenu : c'est une représentation en forme d'arbre, semblable à une table de contenu qui représente une organisation logique pour les ressources d'apprentissage ou les activités qui utilisent ces ressources d'apprentissages. Dans de nombreux cas, mais pas tous, l'arbre hiérarchique peut être parcouru dans un ordre spécifique qui représente l'ordre par défaut de parcours du contenu que l'auteur a prévu pour que l'apprenant progresse avec la matière à apprendre.
2. Les métadonnées : ce sont des données descriptives facultatives, qui sont spécifiques au contexte de la structure ou de l'organisation du contenu. De telles métadonnées peuvent être utilisées pour décrire comment une ressource d'apprentissage particulière peut être utilisée dans une organisation précise du contenu.
3. Le séquencement et navigation : ce sont des prescriptions facultatives qui peuvent être incorporées dans l'organisation du contenu surtout lorsque le développeur veut contrôler et déterminer le rythme auquel les différentes ressources sont présentées à l'apprenant au fur et à mesure qu'il navigue au sein du contenu.

SCORM établit en plus une distinction entre une ressource et un contenu. Une ressource est représentative, comme le décrit le manifeste, d'une ressource aussi bien externe qu'interne au *content package*. Il peut s'agir de fichiers multimédia, de fichiers textes, d'objets de questionnements ou toute autre donnée sous forme électronique. Le

contenu quant à lui représente conceptuellement les fichiers référencés dans les composants des ressources. Cela peut être des fichiers contenus physiquement dans le package tout comme des ressources externes référencés par une URI (*Universal Resource Identifier*).

#### 4.5.1.5.1 Les composants du contenu d'un « package »

Un « *package content* » se compose d'un fichier descriptif de l'organisation des contenus et des fichiers physiques proprement dits. La structure est définie par un document XML descriptif de l'organisation logique répertoriant également les ressources physiques associées. Le fichier XML est utilisé conformément à la spécification *XML binding*<sup>29</sup> de l'*IMS Content Packaging* (ADL). Le *XML binding* a été spécialement coopté pour deux raisons principales : premièrement il adhère à la spécification XML 1.0 du consortium W3C (*Word Wide Web Consortium*) et deuxièmement le il doit maintenir la structure de définition *IMS Content Packaging Information Model*.

Ce fichier XML est appelé « fichier manifeste » (anglicisme dérivé de son libellé : *imsmanifest.xml*). Le manifeste doit se situer à la racine du package comme illustré sur figure 3.1 de la section 3.3.2.1 au chapitre précédent.

Un package représente ainsi une unité logique de formation dite « unité modulaire de formation ». En ce qui nous concerne, Une telle unité logique peut être vue comme une portion de cours ou comme une formation complète contenant plusieurs cours. Une unité modulaire est autonome, c'est-à-dire qu'elle doit contenir toutes les informations nécessaires à l'utilisation des contenus, pour une séquence d'apprentissage, quand elle est importée par la plate-forme. Un package doit contenir un unique manifeste à sa racine, mais celui-ci peut contenir un ou plusieurs « sous-manifestes ».

#### 4.5.1.5.2 Le fichier manifeste

Le manifeste est un inventaire structuré du contenu du package (voir figure 4.5). Si le package est destiné à être livré à un utilisateur final, alors le manifeste contient aussi des

---

<sup>29</sup> Technique informatique qui consiste à décrire un objet à l'aide des ces métadonnées au sein d'un fichier XML. Nous développons cette notion au chapitre prochain.



informations sur l'organisation de ce contenu. L'*imsmanifest.xml* est, comme le nom l'indique, un fichier XML. Le fichier manifeste représente donc les informations nécessaires à la description des contenus du package. Il comporte quatre grandes sections que nous présenterons :

1. Métadonnées ou l'élément `<metadata>` : cet élément contient des métadonnées décrivant le manifeste. Il contient les informations appropriées qui décrivent le *package content* comme un tout. L'élément `<metadata>` est souvent considéré comme le nœud racine pour des métadonnées définis dans un package content. Il possède des nœuds descendants que nous allons illustrer dans un tableau à la fin de cette section.
2. Organisation ou l'élément `<organizations>` : cet élément contient la structure du contenu ou l'organisation des ressources d'apprentissage composant une unité autonome.
3. Ressources : cet élément définit les ressources d'apprentissage empaquetées. l'élément `<resources>` est une collection de références aux ressources. Cependant il n'y a pas d'ordre supposé ou de hiérarchie entre les éléments `<resource>` contenus dans l'élément `<resources>`.

```
<manifest identifier="SAMPLE1" version="1.3" xml:base="mycontent"
xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1
imscp_v1p1.xsd
http://www.adlnet.org/xsd/adlcp_v1p3
adlcp_v1p3.xsd">

  <metadata>
    <!-- les éléments descendants de metadata -->
  </metadata>
  <organizations>
    <!-- les éléments descendants de organizations -->
  </organizations>
  <resources>
    <!-- les éléments descendants de ressources -->
  </resources>
</manifest>
```

Figure 4.5 : Exemple d'un extrait de fichier *imsmanifest.xml*.

### 4.5.1.5.3 Les fichiers physiques

Les fichiers physiques représentent des fichiers référencés dans le composant ressources.

Tableau 4.1 : Extrait du profil d'application des exigences des éléments du manifeste d'un SCORM Content Package (ADL, 2001).

| No.       | Elements                        | Resource Content Package | Content Aggregation Content Package |
|-----------|---------------------------------|--------------------------|-------------------------------------|
| 1         | <manifest>                      | M                        | M                                   |
| 1.1       | identifier                      | M                        | M                                   |
| 1.2       | version                         | O                        | O                                   |
| 1.4       | <metadata>                      | M                        | M                                   |
| 1.4.1     | <schema>                        | M                        | M                                   |
| 1.4.2     | <schemaversion>                 | M                        | M                                   |
| 1.5       | <organizations>                 | M                        | M                                   |
| 1.5.1     | default                         | NP                       | M                                   |
| 1.5.2     | <organization>                  | NP                       | M                                   |
| 1.5.2.1   | identifier                      | NP                       | M                                   |
| 1.5.2.2   | structure                       | NP                       | O                                   |
| 1.5.2.3   | adlseq:objectivesGlobalToSystem | NP                       | O                                   |
| 1.5.2.4   | adlseq:sharedDataGlobalToSystem | NP                       | O                                   |
| 1.5.2.5   | <title>                         | NP                       | M                                   |
| 1.5.2.6   | <item>                          | NP                       | M                                   |
| 1.5.2.6.1 | identifier                      | NP                       | M                                   |
| 1.5.2.6.2 | identifierréf                   | NP                       | O                                   |



Ces fichiers peuvent être des fichiers locaux au package, ou externes et référencés par une URI (*Universal Resource Identifier*).

Notons de manière pertinente que les éléments du fichier manifeste ont chacun une assez longue liste des descendants dont nous regrouperons un extrait au sein du tableau 4.1. Ce tableau révèle aussi les exigences auxquels il faut répondre pour que le contenu que l'on conçoit ainsi que sa structure répondent au profil d'un *SCORM Content Package*<sup>30</sup>. Ainsi en partant du *XML Binding* on donne quelques contraintes sur les éléments et les attributs (voir tableau 4.1) du manifeste ("M" obligatoire, "O" optionnel, "NP" non permis).

#### 4.5.1.6 Les métadonnées

Le modèle SCORM a aussi décrit comment emballer les composantes dans les *Content Aggregations* pour une distribution d'un système à un autre. Une fois que les composants du *SCORM Content Model* ont été construits, il peut être utile et avantageux de décrire ces composants d'une façon cohérente à l'aide de descripteurs ou de métadonnées.

La description des composants avec des métadonnées facilite la recherche et la découverte de ces composants à travers les différents systèmes. Une plate-forme pourrait utiliser les métadonnées pour donner à l'apprenant des informations sur l'organisation du contenu (cours, leçon, module, etc.). Les métadonnées peuvent aussi être utilisées en temps d'exécution pour aider le système à décider de quel composant de contenu livrer à l'apprenant. Les métadonnées utilisées ici sont une émanation du *LOM Learning Content Metadata* qui est un standard de l'IEEE.

Ces métadonnées sont divisées en neuf catégories décrites dans le standard LOM. Nous les illustrons à travers le schéma XML de la figure 4.6, qui est le schéma officiel du LOM.

---

<sup>30</sup> Car l'agrégation du contenu se fait avec des éléments obligatoires et non obligatoires

```

<lom xmlns="http://ltsc.ieee.org/xsd/LOM">
  <general/>
  <classification/>
  <annotation/>
  <lifeCycle/>
  <technical/>
  <metaMetadata/>
  <educational/>
  <relation/>
  <rights/>
</lom>

```

Figure 4.6 : Fichier XML des métadonnées LOM

L'élément parent ici est l'élément `<lom>` et contient les éléments enfants: `<general>`, `<lifeCycle>`, `<metaMetadata>`, `<technical>`, `<educational>`, `<rights>`, `<relation>`, `<annotation>`, et `<classification>`. Le rôle de chaque élément se décrit comme suit :

1. La catégorie *General* est utilisée pour décrire des informations générales sur le contenu du SCORM *Content Model Component* comme un tout.
2. La catégorie *Life Cycle* est utilisée pour décrire les caractéristiques liées à l'histoire et l'état actuel du contenu SCORM ainsi que celles qui ont affectées le contenu durant son évolution.
3. La catégorie *Meta-metadata* est utilisée pour décrire des informations en rapport avec les métadonnées elles mêmes, ce sont des métadonnées décrivant des métadonnées.
4. La catégorie *Technical* est utilisée pour décrire les exigences et les pré-requis techniques ainsi que les caractéristiques du contenu SCORM.
5. La catégorie *Educational* est utilisée pour décrire les caractéristiques pédagogiques et d'éducation du contenu SCORM.
6. La catégorie *Rights* est utilisée pour décrire les éléments de propriété intellectuelle ainsi que les conditions d'utilisation du contenu SCORM.
7. La catégorie *Relation* est utilisée pour décrire les caractéristiques qui régissent les relations entre le contenu SCORM et les autres composants ciblés.



8. La catégorie *Annotation* est utilisée pour fournir des commentaires sur l'usage dans l'enseignement des composants du contenu SCORM, ainsi que des informations sur l'auteur et la date des commentaires émis.
9. La catégorie *Classification* est utilisée pour dire dans quelle catégorie particulière du système on peut classer le contenu.

Toutes ces métadonnées s'intègrent dans un contenu au niveau de l'élément <metadata> dans le fichier manifeste tel que décrit par le *Content Aggregation Model (CAM)*. Cet élément <metadata> a pour vocation de décrire le package comme un tout. Nous donnons une illustration ici de la façon dont les métadonnées du LOM s'intègrent pour décrire le package au niveau de cet élément dans le manifeste.

Cette section de description tout à fait technique du *Content Aggregation Model (CAM)* nous renseigne bien sur le souci de SCORM d'être un modèle fédérateur des autres standards œuvrant dans la normalisation en e-learning, car ici l'empaquetage du contenu hérite des spécifications et standards du *IMS Content Packaging* et la description du contenu elle est basée sur le standard LOM du IEEE.

## 4.5.2 L'environnement d'exécution SCORM

Il s'agit avec l'environnement d'exécution (*SCORM Run-Time Environment, RTE*) d'établir principalement une communication entre la plate-forme et le SCO. Dans le contexte de SCORM, lors de l'exécution d'un contenu, c'est le SCO seul qui communique tandis que l'actif ne communique pas.

L'environnement d'exécution doit permettre de lancer les ressources d'apprentissage et consiste en un mécanisme standardisé permettant aux ressources d'apprentissage de communiquer avec la plate-forme. Il s'appuie sur un langage commun (un vocabulaire prédéfini) constituant la base de la communication.

Comme on peut le voir en examinant la figure 4.7, les trois aspects caractéristiques de l'environnement d'exécution concernent respectivement le lancement, l'interface de programmation d'applications (API) et le modèle de données.

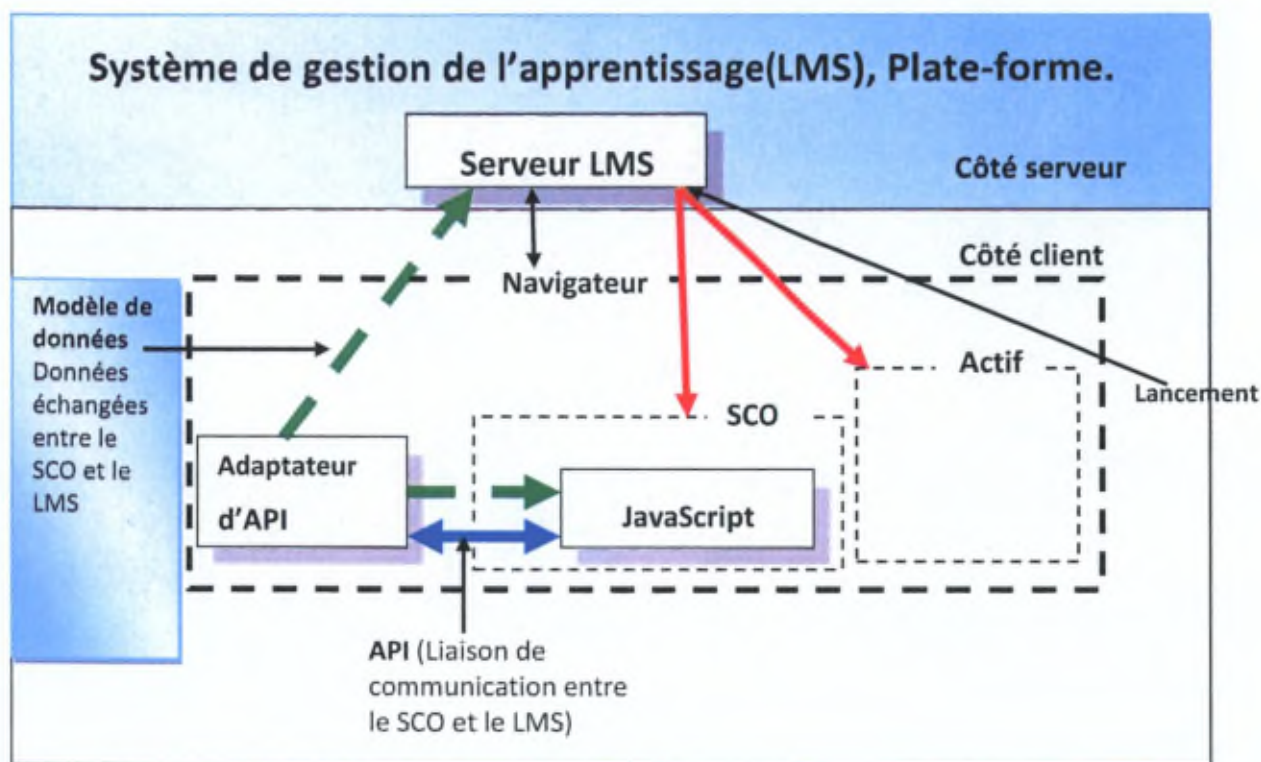


Figure 4.7 : Aspects caractéristiques de l'environnement d'exécution (ADL, 2009).

#### 4.5.2.1 Lancement de ressources d'apprentissage

Le mécanisme de *Lancement* définit un moyen commun (standardisé) permettant à la plate-forme de démarrer des ressources d'apprentissage basées sur le Web. Ce mécanisme définit les procédures et les responsabilités pour l'établissement des communications entre la ressource d'apprentissage et la plate-forme. Les protocoles de communication sont normalisés grâce à l'utilisation d'une API. Les actifs et les SCO sont les deux composants du modèle de contenu SCORM qui peuvent être lancés par une plate-forme. Les procédures et les responsabilités pour l'établissement de la communication entre la ressource d'apprentissage et la plate-forme varient selon le type de ressource d'apprentissage SCORM lancée.

C'est la plate-forme qui est chargée de gérer le séquençement et la navigation entre les ressources d'apprentissage. Elle peut déterminer de façon adaptative le séquençement en fonction de la réalisation de conditions préalables définies associées aux diverses



ressources d'apprentissage. La progression à travers les ressources d'apprentissage, qui constituent un parcours d'apprentissage particulier, peut être séquentielle, non séquentielle, dirigée par l'utilisateur ou adaptative, selon les capacités de la plate-forme.

Le lancement doit être effectué au moyen du protocole HTTP. À la fin du traitement, la ressource d'apprentissage identifiée par l'adresse de lancement dans un contenu conditionné est initialisée et présentée au navigateur du poste client.

#### 4.5.2.1.1 Actifs

Dans le cas des ressources d'apprentissage qui représentent des actifs (*Assets*), le modèle de lancement de SCORM exige uniquement que la plate-forme lance l'actif au moyen du protocole HTTP. Comme un actif n'a pas besoin de re-communiquer, avec la plate-forme, il n'est pas nécessaire qu'il cherche l'adaptateur d'API fourni par ce dernier.

#### 4.5.2.1.2 SCO (Sharable Content Object)

Le modèle de lancement de SCORM exige que la plate-forme lance un seul SCO à la fois et qu'un seul SCO soit actif à un moment donné. Le modèle de lancement exige également que les SCO puissent être lancés uniquement par une plate-forme. Un SCO ne peut pas lancer un autre SCO. La plate-forme doit lancer le SCO dans une fenêtre du navigateur qui est une fenêtre enfant ou un cadre enfant de sa fenêtre. L'adaptateur d'API<sup>31</sup> doit être fourni par la plate-forme. Le SCO doit explorer de façon récursive la hiérarchie des fenêtres parentes ou d'ouverture jusqu'à ce qu'il trouve l'adaptateur d'API. Une fois l'adaptateur d'API trouvé, le SCO peut amorcer la communication avec la plate-forme.

#### 4.5.2.2 API (Application Programming Interface)

L'API constitue le mécanisme de communication qui permet de renseigner la plate-forme sur l'état de la ressource d'apprentissage (par exemple est-elle initialisée, terminée ou dans une situation d'erreur?). Ce mécanisme est utilisé pour lire et fixer les données (par exemple notes, limite de temps, etc.) entre la plate-forme et le SCO. Au sein d'une API on distingue trois concepts principaux : *l'API*, *l'API Implementation*, *l'API Instance*. L'API est

---

<sup>31</sup> C'est un déclencheur des fonctions de l'API qui est expliqué à la section 4.5.2.2.1

simplement un ensemble de fonctions définies que le SCO peut rendre disponibles, l'*API implementation* est un bout de code qui implémente et expose les fonctions de l'API, et l'*API instance* représente pour sa part un contexte d'exécution particulier et un état de l'*API Implementation*.

#### 4.5.2.2.1 Description de l'API de communication SCO-PLate-forme

L'utilisation d'une API fournit aux SCO une façon normalisée de communiquer avec les plates-formes, tout en masquant au développeur de contenu les détails de la réalisation de l'interface de communication. De façon simplifiée, une API est un ensemble de fonctions prédéfinies pouvant être exploitées par les SCO. Une API masque aux SCO les détails de la réalisation, favorisant ainsi la réutilisation et l'interopérabilité.

Un adaptateur d'API est un logiciel fonctionnel qui met en œuvre et présente les fonctions de l'API (voir figure 4.8). Les développeurs de contenu n'ont pas à se préoccuper des mécanismes internes de l'adaptateur d'API, pourvu qu'ils utilisent la même interface publique. Il suffit que la plate-forme fournisse un adaptateur d'API mettant en œuvre les fonctionnalités de l'API et qu'il présente son interface au SCO client.

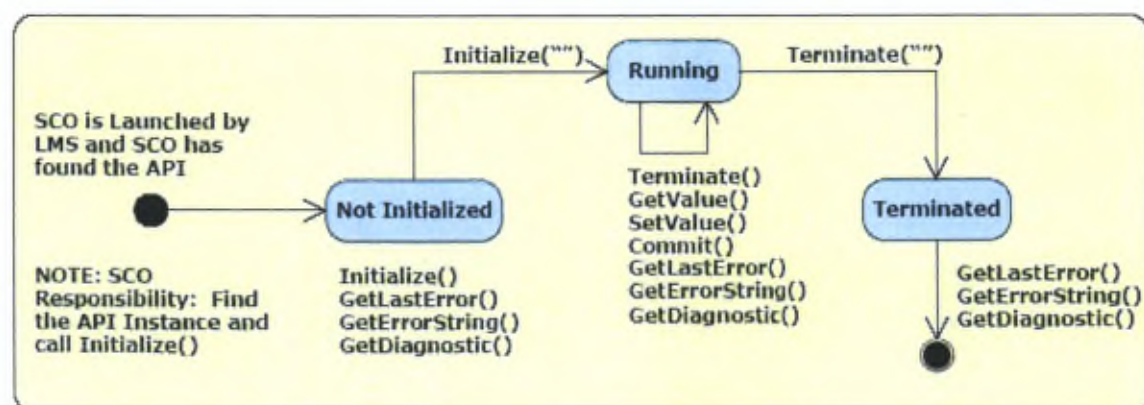


Figure 4.8 : Transition d'état de l'adaptateur d'API de communication SCO-Plate-forme, tiré de (ADL, 2009).

L'adaptateur de communication SCO-Plate-forme passe par plusieurs états, pour une instance donnée d'un SCO, lors de l'exécution. Les états de l'adaptateur d'API correspondent aux réponses de ce dernier à des événements d'entrée particuliers. Lors de



chacun de ces états, le SCO peut entreprendre diverses activités. Voici quels sont les états possibles de l'API : Not Initialized, running et Terminated.

L'API possède huit fonctions : LMSInitialize(), LMSFinish(), LMSGetValue(), LMSSetValue(), LMSCommit(), LMSGetLastError(), LMSGetErrorString(), LMSGetDiagnostic(). Le SCO peut utiliser ces fonctions pour échanger des données avec la plate-forme. Elles sont divisées en trois catégories, que nous déclinons dans le tableau 4.2.

Tableau 4.2 : Les trois catégories des fonctions de l'API

| Fonctions (Methods)  | Description  |
|--|--|
| <i>Session Methods</i> ou fonctions de session                     | Elles marquent le début et la fin d'une session de communication entre un SCO et une plate-forme via l'API instance. |
| <i>Data-transfer Methods</i> ou fonctions de transfert des données | Elles assurent les échanges des valeurs du modèle des données entre un SCO et une plate-forme via l'API instance.    |
| <i>Support Methods</i> ou fonctions auxiliaires                    | Elles s'occupent des communications auxiliaires (le cas des erreurs par exemple)                                     |

#### 4.5.2.2.1.1 Fonctions de session

##### 1. Initialise

- syntaxe de la fonction: `return_value = Initialise (parameter);`
- description : la fonction est utilisée pour amorcer la session de communication. Elle permet à la plate-forme de manipuler (traiter) des questions d'initialisation spécifiques au début de la session.

##### 2. Terminate

- syntaxe de la fonction: `return_value = Terminate (parameter);`

- description : elle met fin à la session de communication, lorsque le SCO réalise qu'il n'a plus besoin de communication avec le LMS ou la plate-forme.

#### 4.5.2.2.1.2 Fonctions de transfert des données

Un SCO utilise généralement ce type de fonctions pour diriger le stockage et la récupération des données à utiliser pendant la communication courante. Le SCO utilise aussi ces fonctions pour transférer les données d'exécution à partir de la plate-forme ou vers elle.

##### **1. GetValue**

- syntaxe de la fonction: `return_value = GetValue (parameter) ;`
- description: cette fonction demande des informations à la plate-forme, cela permet au SCO de déterminer entre autres :
  - les valeurs des éléments du modèle des données supportées par la plate-forme ;
  - la version du modèle des données supportée par la plate-forme ;
  - vérifier si oui ou non certains éléments spécifiques du modèle des données sont supportés.

##### **2. SetValue**

- syntaxe de la fonction: `return_value = SetValue(parameter_1, parameter_2) ;`
- description: cette méthode est utilisée pour demander le transfert à la plate-forme de la valeur de l'argument `parameter_2`, à l'élément spécifié comme `parameter_1`. C'est une méthode qui permet au SCO d'envoyer les informations à la plate-forme pour le stockage. L'instance de l'API peut être configurée pour agir autant sur les données côté serveur que sur les données stockées localement sur le cache côté client.

##### **3. Commit**

- syntaxe de la fonction: `return_value = Commit (parameter) ;`



- description: cette fonction demande l'expédition vers les données actuellement stockées de n'importe quelle donnée du SCO, qui peut avoir été mise en cache par l'instance API depuis le dernier appel à `Initialize ('')` ou `Commit ('')` ou n'importe lequel des deux qui s'est effectué le plus récemment. La plate-forme est supposée mettre le code d'erreur à 0 (pas d'erreur rencontrée) et retourner la valeur 'true'. Si par contre l'instance de l'API n'a pas mis en cache les valeurs des données, la fonction `Commit ('')` retourne la valeur 'true' et met le code d'erreur à 0 (pas d'erreur rencontrée) et ne fait plus aucun traitement. L'appel à la fonction `Commit` ne modifie pas les données en cache.

#### 4.5.2.2.1.3 Fonctions auxiliaires

Ces fonctions existent dans l'API pour simplement permettre au SCO de déterminer le traitement et le diagnostic des erreurs. Avec les fonctions de l'API décrites précédemment les conditions d'erreur peuvent arriver. Quand ces conditions d'erreur sont rencontrées, le code d'erreur est changé pour indiquer l'erreur rencontrée. Les appels aux fonctions auxiliaires n'affectent pas l'état d'erreur. Autrement dit, en les appelant le code courant d'erreur ne change pas. Ces fonctions auxiliaires permettent au SCO de déterminer si une erreur est arrivée et comment traiter les conditions d'erreur rencontrées.

##### 1. GetLastError

- syntaxe de la fonction: `return_value = GetLastError()` ;
- description : cette fonction demande le code de l'état actuel d'erreur de l'instance API. Si le SCO appelle cette fonction, l'instance API ne peut plus changer l'état d'erreur courant. Il ne peut que simplement retourner l'information recherchée.

##### 2. GetErrorString

- syntaxe de la fonction: `return_value = GetErrorString(parameter)`;
- description: cette fonction est utilisée pour obtenir une description textuelle de l'état d'erreur courant. Le SCO l'utilise pour demander la description textuelle de code d'erreur spécifié dans la valeur de l'argument `parameter`. L'instance API sera

responsable de la reconnaissance des codes d'erreurs identifiés. Cet appel n'a aucun effet sur l'état d'erreur courant, il retourne simplement l'information recherchée.

### 3° GetDiagnostic

- syntaxe de la fonction: `return_value = GetDiagnostic(parameter)` ;
- description: cette fonction est créée pour un usage spécifique de la plate-forme. Elle lui permet de définir des informations supplémentaires de diagnostic via l'instance API. Cet appel aussi n'a aucun effet sur l'état d'erreur courant, il restitue simplement l'information demandée.

Selon le modèle de la session de communication, les états de l'instance API traduisent les transitions de ces états vers des événements spécifiques de l'instance API et ces états prédéterminent les fonctions que le SCO peut invoquer. Il s'agit de `Initialized`, `Running` et `terminated`. `Initialized` signale le début de la session de communication avant l'invocation de la fonction `Initialize` et relève de la responsabilité du SCO seul car c'est à lui de trouver l'instance API de la plate-forme. Lorsque l'appel d'`Initialize` a été un succès, on passe à l'état `Running`. Cet état représente l'exécution et comprend toutes les fonctions comprises entre `Initialize` et `Terminate`. Une fois que l'appel à la fonction `Terminate` est un succès, le SCO est autorisé à invoquer maintenant les fonctions auxiliaires de traitement d'erreurs et l'état ainsi obtenu s'appelle `Terminated` tel qu'illustré à la figure 4.7.

SCORM fait appel à un *API Adapter* qui est en fait le modèle de la session de communication. L'*API Adapter* doit résider dans une fenêtre qui est une fenêtre appelante ou un cadre parent de la fenêtre qui contient la leçon. Ce qui signifie que la plate-forme peut lancer le contenu soit dans une nouvelle fenêtre soit dans un ensemble de fenêtre cadre (*frameset*). L'*API Adapter* doit être un objet JavaScript nommé « API ». Cet adaptateur doit implémenter les huit fonctions énumérées précédemment.

Toutes les communications entre le contenu et la plate-forme sont gérées par cet adaptateur. Ainsi le développeur du contenu n'a pas à se soucier de la communication avec le serveur, il doit seulement trouver l'API et appeler les fonctions adéquates en JavaScript.



Cette séparation du client et du serveur est essentielle avec SCORM et assure ainsi la portabilité des contenus. Il est important de bien noter que le contenu peut communiquer avec la plate-forme uniquement en JavaScript avec cette API. En revanche il n'y a pas de méthode obligatoire et le contenu peut utiliser un service Web ou une requête HTTP.

## 4.5.3 Le séquençement

### 4.5.3.1 Structure de contenu et arbre d'activité

Le but poursuivi ici est d'activer pour chaque apprenant « un arbre d'activité » à partir du « package de contenu ». Le séquençement de SCORM (*SCORM Sequencing and Navigation*) est basé sur les travaux des spécifications de l'*IMS Simple Sequencing (IMS SS)* qui ont pour but de prévoir pour un apprentissage le comportement de la plate-forme par rapport au séquençement des activités discrètes d'apprentissage. Dans le séquençement, IMS SS ne tient compte que de l'apprenant lui-même sans s'occuper des apports issus ou dépendants des autres acteurs tels que l'enseignant, le tuteur ou ses pairs.

En définitive, le séquençement SCORM dépend d'une structure définie d'activités d'apprentissage, structure appelée l'arbre d'activité. Une stratégie de séquençement définit le modèle de définition de séquençement. L'application du comportement définit des événements déclenchés par le système ou externes au système, les comportements du séquençement de SCORM. La navigation assume l'existence de dispositifs d'interface utilisateur pour déclencher des événements de navigation. Ces dispositifs peuvent être fournis par la plate-forme ou incorporés dans des objets d'apprentissage. Quand l'apprenant déclenche un tel dispositif, la plate-forme traduit l'événement dans sa requête de navigation correspondante, traite la requête et indique ensuite l'activité d'étude suivante à livrer.

### 4.5.3.2 Les concepts du séquençement

#### 4.5.3.2.1 Arbre d'activités

C'est dans la structure du contenu que les liens hiérarchiques des différents objets d'apprentissage sont définis, le plus souvent dans le manifeste ou les sous-manifestes s'il y en a. Un arbre d'activité est la représentation logique, dans la mémoire de la plate-forme, de

la hiérarchie de contenu décrit dans le manifeste. Il permet de gérer une activité d'apprentissage et de l'individualiser, et permet également au *SCORM Sequencing Model* de décrire les conditions de traitements tels que les algorithmes de séquencement et les comportements lors d'une implémentation réalisée de manière indépendante. Aussi un arbre d'activité est un terme général qui représente une instance d'activités d'apprentissage hiérarchiques et les informations de séquencement correspondantes pour une application interopérable des comportements de séquencement indiqués. C'est l'élément *<organization>* du manifeste qui constitue la racine à partir de laquelle on peut extraire un arbre d'activité, et chacun de ses éléments *<item>* correspond à une activité d'apprentissage comme illustré à la figure 4.8. Cette correspondance du contenu et l'arbre d'activité s'articule en trois points :

1. Un arbre d'activité représente la structure conceptuelle du contenu qui dérive des processus de conception et d'agrégation.
2. Une plate-forme conforme SCORM traduit une organisation des contenus en un arbre d'activités. Un arbre d'activité représente la structure de données qu'une plate-forme implémente pour refléter la représentation hiérarchique interne des activités d'apprentissage définies, y compris le dépistage d'informations de statut pour chaque activité dans la hiérarchie, basée sur l'apprenant.
3. Quand un apprenant choisi d'interagir avec du contenu représenté par un arbre d'activité, la plate-forme évalue les informations de séquencement et de dépistage afin de déterminer la séquence liée à une activité d'apprentissage, tout comme l'éligibilité d'une activité d'apprentissage à être essayée par un apprenant sur une base conditionnelle.

#### 4.5.3.2.2 Cluster

Un «*cluster*» est une activité qui intègre une ou plusieurs sous-activités. Il comprend un parent et ses fils immédiats mais pas les descendants de ses fils. Le schéma de la figure 4.9 montre cinq exemples de *cluster*. Le *cluster* est considéré comme le bloc de base de l'arbre d'activité. Beaucoup d'éléments du séquencement s'appliquent spécifiquement aux *clusters*. La partie parente du *cluster* contient les informations globales du séquencement.



Un *cluster* a pour enfant aussi bien d'autres *clusters* que des activités feuilles, cependant une activité feuille n'est pas un *cluster*.

Pour revenir à l'exemple de notre cours à la section 4.5.1.2, prenons le module « Electricité » comme étant un arbre d'activités composé des chapitres 7, 8, 9 et 10. Chaque chapitre a deux travaux pratiques que nous nommons TP  $i,j$  ( $i$  : rang du cours,  $j$  : rang du TP au sein du cours), par exemple, TP 9.1 pour premier TP du chapitre 9. L'arbre d'activités se présentera comme à la figure 4.10.

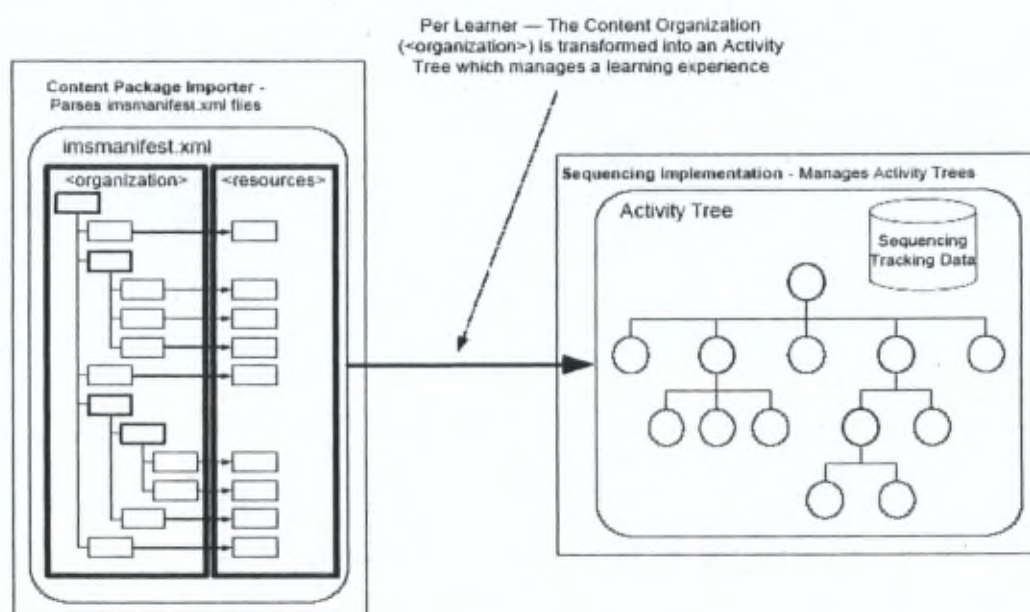


Figure 4.9 : Correspondance entre le contenu et l'arbre d'activité (ADL, 2009).

Nous avons dans notre exemple sept *clusters* au total qui sont composés de la manière suivante, sachant que le *cluster* rassemble uniquement le nœud et ses enfants :

- *cluster* 1 : l'arbre d'activité et le nœud « Module Electricité » ;
- *cluster* 2 : module Electricité plus chapitre 7, chapitre 8, chapitre 9, chapitre 10 ;
- *cluster* 3 : chapitre 7, TP 7.1, TP 7.2 ;
- *cluster* 4 : chapitre 8, TP 8.1, TP 8.2 ;
- *cluster* 5 : chapitre 9, TP 9.1, TP 9.2 ;
- *cluster* 6 : chapitre 10, TP 10.1, TP 10.2 ;
- *cluster* 7 : TP 8.2, questions théoriques, manipulations labo.

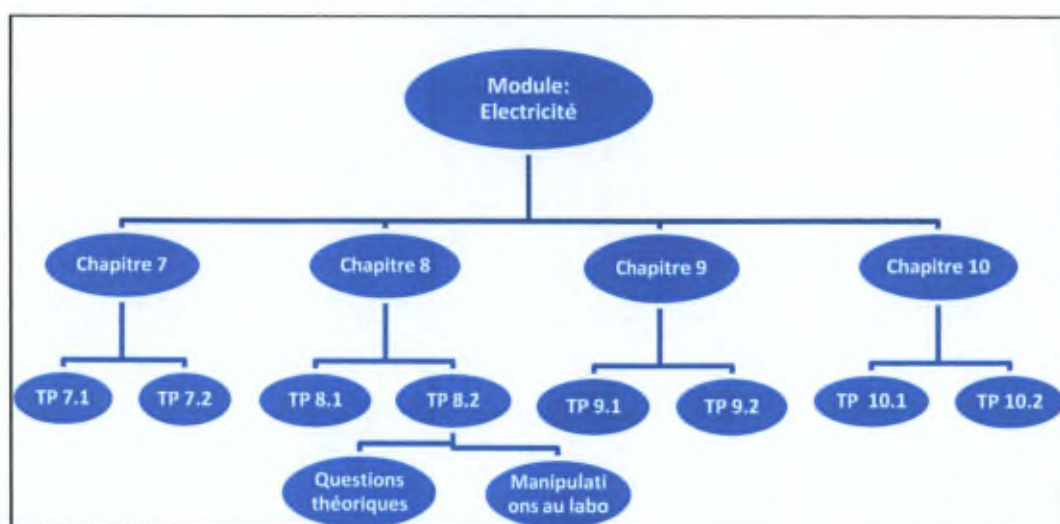


Figure 4.10 : Exemple d'arbre d'activité.

#### 4.5.3.2.3 Activité d'apprentissage

Une activité d'apprentissage peut être définie comme une unité significative d'instructions (voir figure 4.11). Le terme activité d'apprentissage est le synonyme du terme activité. Une activité peut soit référencer une ressource soit référencer des sous-activités. Les sous-activités peuvent être supplémentaires à n'importe quel niveau d'emboîtement. Celles qui ne sont pas supplémentaires sont considérées comme des activités feuilles. Les activités feuilles sont associées à un objet.

La plate-forme identifiera l'activité d'apprentissage à livrer dans une séquence d'apprentissage déterminée pendant un temps d'exécution basée sur les progrès réalisés par l'apprenant dans la séance précédente d'activité d'apprentissage.

La série d'objets utilisés par l'apprenant au cours des activités d'apprentissage constituent son parcours pédagogique. Une activité présente les caractéristiques suivantes :

- elle possède un début et une fin discrète ;
- elle définit parfaitement les conditions nécessaires à la « complétude de l'activité » et la maîtrise ;
- elle peut consister en sous-activités emboîtées à n'importe quelle profondeur ;
- elle s'exécute dans le contexte de son parent s'il en existe un.



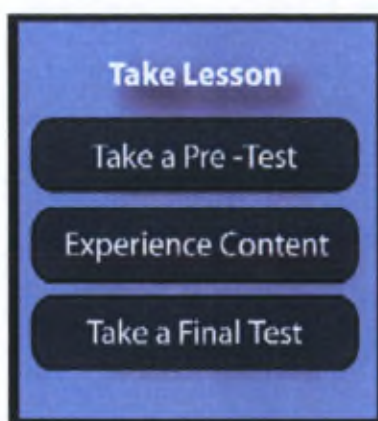


Figure 4.11 : Une activité d'apprentissage extrait de (Fage, 2005)

#### 4.5.3.2.4 Tentatives

Une tentative est définie comme un effort d'arriver au bout d'une activité, des objectifs pédagogiques pouvant être satisfaits ou pas. Les tentatives sur une activité sont liées au contexte des activités parents. Il est important de noter ici que pour un arbre d'activités donné, une et une seule activité feuille peut être essayée à un moment donné et toutes les tentatives sur ses ancêtres via la racine vont progresser en même temps. Quand une activité est essayée, cela va de soi que l'objet de contenu correspondant est lancé. Le statut de dépistage pour une activité peut changer, par le résultat d'une tentative sur une activité ou par une action administrative extérieure, et lorsque son statut change celui de ses ancêtres est aussi affecté, on appelle ça un *Roll-Up*.

#### 4.5.3.3 Démarrage et Arrêt d'une session de séquencement

Une session de séquencement indique le temps entre le commencement d'une tentative sur l'activité racine d'un arbre d'activité jusqu'à son arrêt. En dehors du contexte de la session de séquencement, l'activité courante est considérée comme non définie. Les comportements du séquencement de SCORM déterminent quelles requêtes de navigation peuvent commencer une session de séquencement, mais ils ne spécifient pas quand ou comment ces requêtes de navigation sont déclenchées.

#### 4.5.3.4 Objectifs pédagogiques

Cette notion n'est pas encore définie ni développée suffisamment par SCORM. Car bien que les objectifs pédagogiques soient bien distincts des activités d'apprentissage, SCORM n'exige et n'explicite rien quant à comment associer les objectifs pédagogiques aux activités d'apprentissage ni comment les contenus utilisent les objectifs pédagogiques. Un objectif peut être local à une activité ou global à plusieurs activités. Il y a deux restrictions majeures sur la manière dont une activité peut référencer des objectifs globaux partagés :

1. Un objectif local peut lire les composants d'informations des traces de suivi d'un et d'un seul objectif global partagé.
2. Pour l'ensemble des objectifs locaux définis pour une activité donnée, deux objectifs ne peuvent lire les traces de suivi sur le même objectif global partagé.

#### 4.5.3.5 Le tracking

Le *tracking* consiste à pouvoir suivre (littéralement poursuivre) le cheminement et l'activité de l'apprenant dans son parcours de formation. Cet accompagnement est organisé à partir des données significatives mémorisées au cours de la formation. L'état du *tracking* peut changer suite à un comportement particulier au sein d'une activité. Si l'état du *tracking* d'un objet change, cela peut également affecter l'état du *tracking* de ses ancêtres. Ce mécanisme s'appelle le *Roll-up*.

Pour permettre le séquençage conditionnel des activités, les informations sur les interactions de l'apprenant avec les objets de contenus associés doivent être mises à jour et gérées. Le comportement du séquenceur SCORM est lié aux valeurs courantes du modèle d'état du *tracking*. A chaque activité déroulée par un apprenant sont associées des données représentatives de l'état du *tracking* (par exemple, la durée de la connexion, le nombre des clics, le nombre de pages parcourues, etc.). Les interactions de l'apprenant avec un objet de contenu peuvent naturellement affecter l'état du *tracking* de l'activité associée à l'objet. Pendant le parcours d'apprentissage, les éléments du modèle de *tracking* mis à jour reflètent les interactions de l'apprenant avec l'objet de contenu lancé.



## 4.6 Utilisation de SCORM dans nos travaux

Après avoir scruté les trois concepts principaux du modèle SCORM à savoir le CAM, le RTE, et le SSN développés à la section 4.5, nous avons identifié les éléments nécessaires à l'implémentation de notre outil d'interopérabilité. Il est en effet prévu que nous prendrons SCORM comme modèle de référence pour les différents contenus conçus et produits par deux plates-formes hétérogènes. En assurant la compatibilité SCORM, nous nous assurons que notre outil couvrira un large spectre de contenus d'apprentissage produits par différentes plates-formes. Il était donc justifié de nous focaliser sur SCORM. Les études démontrent d'ailleurs que le pourcentage des plates-formes qui sont conformes ou certifiées SCORM ne cesse d'augmenter. A titre d'exemple, l'étude effectuée par un site français spécialisé en technologies e-learning<sup>32</sup> qui, comme le montre la figure 4.12, affirme que toutes les plates-formes généralistes sont conformes et certifiées SCORM 1.2. Les plates-formes open source, qui nous intéressent particulièrement ici, sont en grande majorité réputées conformes au SCORM (85%), un quart d'entre elles étant certifiées conformes par le ministère américain de la défense via l'ADL. Notons cependant que par rapport à SCORM 1.2 qui a été très vite adopté dans le secteur e-learning, SCORM 2004 a eu des difficultés à s'imposer mais, de nos jours, l'évolution de son adoption est devenue comparable à celle de SCORM 1.2.

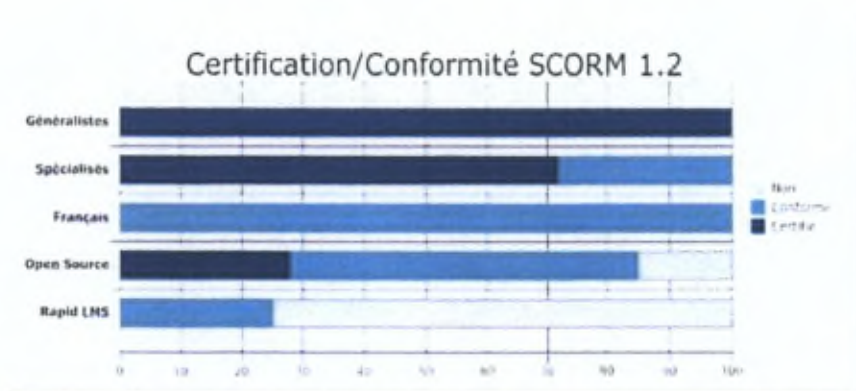


Figure 4.12 : certifications et conformités des plates-formes open source<sup>33</sup>

<sup>32</sup> [www.lms-selection.fr](http://www.lms-selection.fr): site de l'entreprise *KNOWLEDGE DECISION* expert indépendant en technologies e-Learning (Consulté en mai 2011).

<sup>33</sup> Extrait du même site.

## CHAPITRE 5 : ETUDE COMPARATIVE DES STRUCTURES DES CONTENUS MOODLE-DOKEOS

### 5.1 Justification du choix

Rappelons que l'étude que nous menons à travers ce travail a un aspect bicéphale qui concerne d'une part l'Université de Lubumbashi et d'autre part ses partenaires belges. Tenant compte du nombre grandissant des plates-formes d'e-learning recensées à ce jour dans le monde (un peu plus de 200<sup>34</sup>). Devant la complexité du sujet de l'interopérabilité des plates-formes en e-learning, nous nous sommes trouvés devant un choix à faire pour matérialiser notre apport dans ce travail. Du côté UNILU, en se basant sur une étude que nous avons précédemment effectuée (Fyama, 2006) qui consistait, partant d'une étude comparative de trois plates-formes open source (Claroline, Moodle, et Dokeos), à effectuer un choix sur la mieux adaptée après une analyse multicritères basée sur des critères ergonomiques, pédagogiques et technologiques. Le choix de la plate-forme s'est porté sur Dokeos. Du côté partenaires belges, pour effectuer le choix le plus représentatif, nous sommes partis de la tendance des usages au niveau mondial, comme au niveau européen en matière des plates-formes. Une étude menée par Mark Smithers<sup>35</sup>, propose une comparaison d'usage et des plates-formes, notamment Blackboard (ex WebCT), Moodle, Angel 7.2, Desire2Learn, BlackBoard Vista 4.2, Sakai, etc., dans 11 universités américaines et australiennes. Son étude comparative basée sur des critères pédagogiques, technologiques, ergonomiques et financiers montre que sept universités sur 11 ont préféré Moodle à près d'une dizaine d'autres plates-formes. De même en 2007 notamment, le répertoire de THOT<sup>36</sup> comptait près de 233 plates-formes dans le monde dont 47 plates-formes open-source, 11 plates-formes publiques<sup>37</sup>, 175 plates-formes commerciales. Les plus utilisés sont dans l'ordre<sup>38</sup> :

<sup>34</sup> <http://www.siloinspiroche.com>: blog de Stéphane Wattier expert français en e-learning (consulté mai 2011)

<sup>35</sup> <http://www.masmithers.com>: expert australien en e-learning (consulté mai 2011)

<sup>36</sup> <http://www.cursus.edu>: le portail e-learning de référence dans la francophonie (consulté en mai 2011)

<sup>37</sup> Les plates-formes dites publiques sont celles utilisées par les organismes publics ou étatiques

<sup>38</sup> Résultats tirés de : <http://www.cursus.edu> (consulté en mai 2011)



1. Moodle. Plus de 50000 sites dans plus de 208 pays ont enregistré leur implantation. L'implantation la plus importante comportant 19000 cours avec 41000 utilisateurs.
2. WBT Manager. Deuxième plate-forme au monde.
3. WebCT. Plate-forme utilisée dans plus de 80 pays par plus de 3500 institutions.
4. Claroline. Deuxième plate-forme européenne après Moodle.
5. Dokéos. Plate-forme ayant deux millions d'utilisateurs, 7155 portails, 150794 cours.
6. Atutor. Plate-forme très utilisée dans l'enseignement supérieur aux États-Unis.
7. SAKAI. Plate-forme utilisée dans des universités de renommée mondiale comme Cambridge, Yale, Cape Breton University, Stanford, the Australian National University et Massachusetts Institute of Technology.
8. GANESHA. Plus de 3656 téléchargements de la version 1.2.1, plus de 3395 téléchargements de la version 1.1.0, 124 abonnés dans la mailing-list Ganesha (développement).

Vu l'incontestable succès de Moodle, aussi bien à l'échelle mondiale qu'à l'échelle européenne, nous l'avons choisi comme étant plus représentative de la partie belge. De plus, de l'avis de plusieurs observateurs, 90% des implémentations de Moodle se concentre dans les milieux universitaires, en remplacement de WebCT. Pour la Belgique, le site officiel de Moodle<sup>39</sup> dénombre un total de 191 sites enregistrés sans compter les 44 sites privés n'apparaissant pas sur leur liste. Parmi les partenaires déclarés de l'Université de Lubumbashi, on note l'Université de Mons, l'Université Catholique de Louvain, la Faculté Universitaire des Sciences Agronomiques de Gembloux ainsi qu'à l'Université de Liège. Le projet actuel de campus virtuel entre l'UNILU et l'AUF (Association des Universités Francophones) ayant choisi comme plate-forme Moodle est un élément encourageant dans notre choix.

Nous rappelons, pour finir cette introduction, que notre travail se concentrera, au vu des considérations précédentes, sur l'interopérabilité entre Moodle et Dokeos. Ceci d'abord parce que les deux plates-formes, comme nous l'avons montré, sont suffisamment représentatives des deux milieux d'application du travail (UNILU et universités belges). Aussi

---

<sup>39</sup> [http : //moodle.org](http://moodle.org), site officiel de la plate-forme Moodle (consulté en Août 2011).

le cas Moodle-Dokeos constitue un substrat à partir duquel d'autres développements sur l'interopérabilité peuvent être menés.

## 5.2 Architecture et présentation des plates-formes

### 5.2.1 Architecture des plates-formes

Chaque concepteur ou éditeur de plate-forme définit son propre formalisme et les concepts utilisés ne sont pas homogènes entre plates-formes. Cela oblige les utilisateurs à adopter le modèle pédagogique intrinsèque, souvent implicite, de la plate-forme utilisée. C'est la pratique, l'usage et l'habitude qui aident les différents acteurs à décortiquer les modèles sous-jacents des plates-formes (Leclercq, et al., 2003). Au niveau informatique, il est aussi important d'appréhender l'architecture d'une plate-forme à travers ses fonctionnalités, ainsi que leur mise en œuvre interne, pour déterminer la correspondance des concepts entre plates-formes hétérogènes dans un projet d'interopérabilité.

La structure générale des plates-formes d'e-learning se base le plus souvent sur une architecture client/serveur avec un serveur Web, une base de données ou un système de fichiers pour le stockage des ressources supplémentaires (par exemple des fichiers PDF). Cependant, l'évolution et le progrès technologique en matière d'architecture Web conduisent actuellement les plates-formes à proposer des architectures trois-tiers. Ce type d'architecture permet notamment de séparer la présentation, le traitement et le stockage des données, elle est découpée de la manière qui suit :

- un client, généralement le navigateur de l'utilisateur, assure la partie présentation et fournit l'interface de commande ;
- un intergiciel<sup>40</sup> garantit, par l'intermédiaire d'un serveur Web, la génération et la mise en ligne des pages correspondantes aux réponses des applications spécifiques ;
- un serveur de base de données stocke les documents avec les métadonnées associées.

---

<sup>40</sup> En anglais « *Middleware* » c'est un logiciel intermédiaire traduisant les données échangées entre plusieurs applicatifs afin de garantir l'interopérabilité des applications



Les deux plates-formes Moodle et Dokeos reposent, pour leur déploiement, sur une architecture trois-tiers. A l'Université de Lubumbashi, le serveur Web est Apache, le système de gestion des bases de données MySQL, l'intergiciel est PHP et nous avons utilisé Mozilla Firefox comme client. La présentation que nous ferons de ces deux plates-formes dans les sections suivantes résulte de la compilation des observations relevées durant leur utilisation pendant ces cinq dernières années à l'Université de Lubumbashi. Elle sera aussi l'émanation de la documentation officielle tirée des sites des ces plates-formes et enfin nous nous sommes aussi basés sur des études d'évaluation faites par d'autres chercheurs dans le cadre de l'enseignement supérieur et universitaire.

## 5.2.1.1 Présentation de Moodle

### 5.2.1.1.1 Introduction

La plate-forme Moodle a été créée en 2002, par un ancien administrateur de la plate-forme WebCT (maintenant Blackboard) à l'Université de Curtin en Australie, du nom de Martin Dougiamas. Ce dernier, au cours de ses recherches doctorales, a étudié les apports du constructivisme social dans l'apprentissage en ligne. Ses travaux ont fortement influencé la conception de la plate-forme. Depuis 2002, à sa création, Moodle reste une plate-forme d'avant-garde, aujourd'hui, il existe plus de 200 modules additionnels développés depuis des années par la communauté.

Moodle est une plate-forme e-learning servant à créer des communautés d'apprenants autour de contenus et d'activités pédagogiques, il est téléchargeable gratuitement sur le site officiel. Le terme Moodle est l'acronyme de *Modular Object-Oriented Dynamic Learning Environment* en français Environnement d'Apprentissage Dynamique Modulaire et Orienté-Objet, mais il veut aussi dire « flâner » en anglais. Dotée d'un système de gestion de contenu (SGC) performant, Moodle implémente des fonctions pédagogiques et communicatives lui permettant de créer un environnement d'apprentissage en ligne. Ces fonctionnalités permettent de créer des interactions entre enseignants, apprenants et ressources pédagogiques, formant ainsi un réseau et parfois même une véritable communauté autour d'un thème choisi par les membres de la plate-forme (par

exemple, l'apprentissage d'un logiciel comme l'utilisation de la plate-forme). C'est une particularité que l'on retrouve peu dans les autres plates-formes d'e-learning.

Moodle fait partie des systèmes d'e-learning qui sont appelés dispositifs de «Formation Ouverte et à Distance» (FOAD), pour favoriser un cadre de formation socioconstructiviste. Ce courant de pensée affirme que les gens construisent activement leurs nouvelles connaissances en interagissant avec leur voisinage. Tout ce que nous lisons, voyons, entendons, ressentons et touchons est comparé à nos connaissances antérieures et si cela est viable dans notre monde mental, il pourra former une nouvelle connaissance qui nous appartiendra. L'interface de Moodle se présente sous forme de différents blocs : le bloc central présente les documents et les activités du cours (voir figure 5.1). Ils peuvent être classés selon plusieurs formats :

- Thématique. Classement en fonction de thèmes ou de sujets du cours.
- Hebdomadaire. Classement suivant un agenda ou un calendrier.
- Informel. Classement en fonction de sujets de discussion et de forum.

#### 5.2.1.1.2 Page d'accueil d'un cours Moodle et fonctionnalités offertes.

Sur une page d'accueil d'un cours Moodle, pour présenter les ressources, l'enseignant a plusieurs choix :

- composer une page de cours avec un texte court sans mise en forme ;
- créer une page Web au format HTML qui peut être éditée à l'aide de l'éditeur intégré à la plate-forme ;
- mettre un lien vers un site Internet ou un fichier déposé sur le serveur ;
- afficher le contenu d'un dossier par un lien vers une liste de fichiers déposés sur le serveur ;
- ajouter un fichier *IMS Content Package* par un lien vers un fichier de format SCORM ou AICC ;
- insérer une étiquette permettant de commenter la ressource.



Les blocs latéraux affichés sur les pages Web donnent accès aux différents outils et liens du cours, par exemple :

- « *Personnes* » : la liste des inscrits au cours et celle des différents sous-groupes ainsi que l'accès aux profils ;
- « *Cours* » : la liste des cours auxquels est inscrit l'utilisateur ;
- « *Recherche* » : l'outil de recherche dans les forums du cours ;
- « *Administration* » : le relevé des notes de l'usager et autres éléments de son profil ;
- « *Dernières nouvelles* » : les dernières nouvelles brèves publiées sur le forum ;
- « *Prochains événements* » : les activités inscrites au calendrier de son cours ;
- « *Calendrier* » : les activités classées en fonction du calendrier ;
- « *Utilisateurs en ligne* » : la liste des personnes, enseignants et usagers, connectés au cours ;
- « *Fils RSS* » : pour la mise à jour des nouvelles venues des sites externes.

Les membres d'un cours ont accès aux activités suivantes si l'enseignant les a sélectionnées :

- « *Messagerie électronique* » : messagerie instantanée ou salon de discussion qui offre la possibilité de planifier par heure, par jour ou de manière hebdomadaire, etc. ;
- « *Forum* » : différents types de forums où sont discutés des sujets divers : soit des sujets imposés par l'enseignant, soit des sujets proposés par les étudiants, avec des évaluations, avec des commentaires possibles, etc. Notons que l'étudiant ne peut avoir le droit de poster un sujet que si cela est au préalable permis dans la configuration du cours par l'enseignant ;
- « *Devoir* » : remise de travaux avec évaluation de l'enseignant ;
- « *Test* » : suite de QCM (Questionnaire à Choix Multiples), de questions vrai/faux, appariement, etc. ;
- « *Leçon* » : document comprenant des questions et plusieurs parcours possibles en fonction des réponses avec une possible évaluation ;
- « *Atelier* » : remise de travaux avec évaluation par les étudiants ;

- « *Glossaire* » : production collective d'un document organisé alphabétiquement accompagné parfois de commentaires et d'évaluations ;
- « *Wiki* » : production collective d'un document hypertexte où l'enseignant peut ajouter des commentaires s'il le souhaite ;
- « *Journal* » : rédaction d'un journal personnel où d'autres peuvent ajouter des commentaires ;
- « *Dialogues* » : messagerie interne entre membres du cours.

Retenons aussi que toutes les activités qui concernent un cours sont paramétrables par l'enseignant titulaire ou concepteur du cours. L'enseignant peut en outre diviser son groupe de classe en plusieurs sous-groupes de manière à faciliter la communication entre les personnes. Les groupes ont des outils dédiés : forum, sondage, chat, etc. Moodle a été créée de manière modulaire : elle permet de répondre aux besoins d'un formateur isolé comme d'une institution académique.

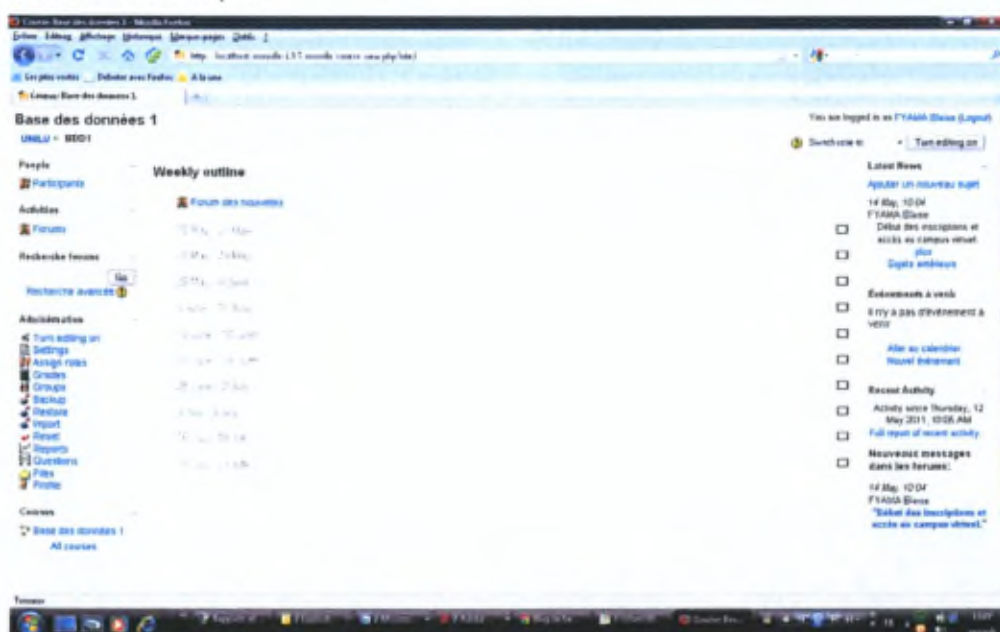


Figure 5.1 : Illustration de la page d'accueil du cours de Base de données 1.

### 5.2.1.1.3 Notre bilan sur Moodle

Aujourd'hui, le développement de Moodle est fortement influencé par les demandes de la communauté d'administrateurs et d'utilisateurs (enseignants, pédagogues, informaticiens etc.). Ce projet bénéficie d'un développement très actif à l'échelle mondiale.



Moodle est très novateur dans l'exploitation des nombreuses spécifications en e-learning et tente d'implémenter notamment, SCORM pour partager les contenus, *Dublin Core Metadata* pour les métadonnées, *IMS Simple Sequencing* pour les parcours d'apprentissage et *IMS Content Packaging* pour la gestion des contenus. Elle fait un effort palpable et va notamment au-delà des autres plates-formes en essayant de manière active d'intégrer IMS-LD, une des dernières spécifications parues, visant à incorporer la flexibilité pédagogique et elle contribue à compléter certains aspects traités par les autres normes avec des aspects pédagogiques. Elle sera l'une des premières plates-formes à l'intégrer vers mi-2008. IMS-LD n'impose pas de modèle pédagogique particulier, mais peut être utilisée avec un grand nombre de scénarii et de modèles pédagogiques.

Les forums de discussions de Moodle sont très actifs. Ces groupes de discussions participent à une critique positive sur les caractéristiques, l'utilisabilité, les fonctionnalités des spécifications, leur contexte théorique et les applications qui leurs sont liées. Très au fait des nouvelles tendances pédagogiques comme technologiques, la communauté de Moodle s'intéresse aussi au *Portfolio*, au standard RSS, au *Podcasting* ou « *balladodiffusion* ». L'enseignant a la possibilité de faire une sauvegarde de ses cours avec ou non les données et les productions estudiantines. La restauration d'une sauvegarde permet de créer ou compléter un cours de manière extrêmement rapide. On peut aussi réinitialiser le cours afin de garder sa structure sans les ressources, les utilisateurs et les échanges d'informations. Ceci permet à un enseignant d'utiliser la même base pour tous ses cours. Un cours peut être défini comme « méta-cours » d'un cours principal, chaque étudiant qui s'inscrit dans le cours principal est automatiquement inscrit dans les méta-cours connexes prédéfinis. Moodle peut accueillir un grand nombre d'utilisateurs, et l'une des plus grandes implémentations compte plus de 41000 utilisateurs et environ 19000 cours, comme indiqué en début de ce chapitre à la section 5.1.

Le projet Moodle dispose d'une communauté mondiale d'utilisateurs dont il reste très à l'écoute grâce à des conférences annuelles. Très axée sur les interactions entre utilisateurs, formateurs et tuteurs, la plate-forme favorise, par sa structure et ses outils, l'échange entre apprenants. Sa modularité, la richesse des possibilités qu'offre le

paramétrage de chaque outil, les différentes interfaces proposées pour chaque cours font de Moodle une plate-forme extrêmement novatrice, originale, et aujourd'hui incontournable.

#### 5.2.1.1.4 Les points faibles de la plate-forme

Au-delà de tout le discours laudatif, ventant les mérites et avantages de Moodle, la plate-forme présente quelques faiblesses que nous recensons dans cette section.

Il existe une possibilité de test de positionnement<sup>41</sup> mais sa gestion est manuelle, et il n'y a pas d'affectation automatique de parcours.

Dans l'agenda du cours, on peut effectuer une visualisation des nouveautés, une prévision d'événements pour tout type d'utilisateur. Les travaux à rendre apparaissent automatiquement dans l'agenda, mais il n'existe pas d'agenda personnel.

La communication en mode synchrone n'existe qu'avec le chat ou clavardage. Il n'y a pas de mode vidéoconférence (du reste très réclamée à l'UNILU) sans rajouter un module qui parfois est payant.

Moodle est une plate-forme très riche en fonctionnalités, et sa prise en main par les apprenants peut nécessiter un temps d'adaptation, car les pages peuvent être très chargées en informations ce qui constitue une lourdeur ergonomique.

Pour les enseignants, la diversité et la spécificité de tous les paramétrages des outils peuvent paraître trop complexes aux yeux d'utilisateurs peu familiers à l'apprentissage en ligne et à l'informatique. Cependant, malgré les nombreux tutoriels qui existent en ligne, il est aussi important que l'administrateur de la plate-forme ou le coordinateur du projet puisse être disponible pour aider les enseignants à appréhender la totalité des fonctionnalités de Moodle.

---

<sup>41</sup> Le positionnement révèle une évolution quantifiée (par exemple sur un total de 10) de l'évolution d'un apprenant par rapport aux objectifs pédagogiques du cours.



## 5.2.1.2. Présentation Dokeos

### 5.2.1.2.1 Introduction

Classée actuellement deuxième après Moodle à l'échelle européenne (voir section 5.1), Dokeos est une plate-forme open source (sous licence GPL<sup>42</sup>). Créé par Thomas Depraeter, qui a participé à la création de Claroline, ce logiciel est un fork<sup>43</sup> de Claroline. Il s'appuie sur une architecture multilingue qui supporte 34 langues. Le logiciel est écrit en PHP et utilise le Système de Gestion de Bases de données MySQL. Il est réputé efficace, complet, modulaire, évolutif, et supporté par une importante communauté d'utilisateurs. Le fait d'être open source, disponible et fonctionnel milite en faveur de son adoption comme plate-forme la plus adaptée pour un public novice et débutant en e-learning, typique de l'Afrique subsaharienne. Il s'est en outre révélé comme la plate-forme la plus ergonomique lors d'une étude comparative que nous avons menée pour l'Université de Lubumbashi en 2006 (Fyama, 2006).

### 5.2.1.2.2 Les fonctionnalités de Dokeos

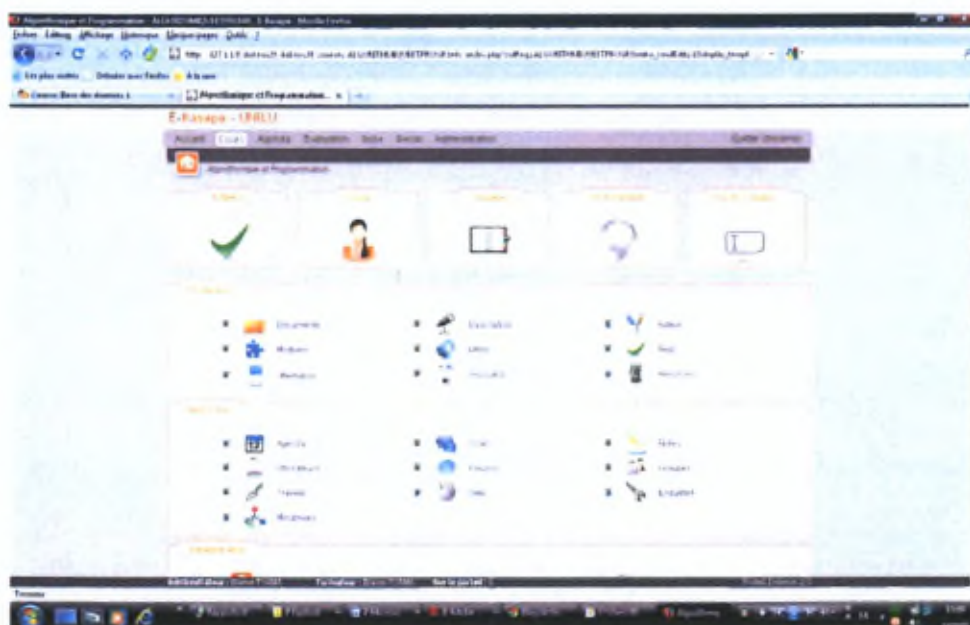


Figure 5.2 : Exemple de page d'accueil d'un cours Dokeos.

<sup>42</sup> General Public Licence, licence des logiciels dont le code source est à accès libre.

<sup>43</sup> Un fork, ou embranchement, est un nouveau logiciel créé à partir du code source d'un logiciel existant.

L'interface d'accueil d'un cours sur Dokeos (voir figure 5.2) offre les fonctionnalités suivantes :

*Gestion des cours et catégories.* Dokeos propose, en plus des catégories de cours communes à tous les créateurs de cours, la possibilité de créer des catégories personnelles connues et visibles du concepteur seul, à des fins de classement et d'organisation de son propre espace de cours. Il y décidera notamment de l'organisation des cours, des inscriptions, ainsi que de la visibilité de ses cours par les autres utilisateurs.

*Description du cours.* Cet outil invite le concepteur du cours à le décrire de manière synthétique et globale dans une logique de cahier des charges. Cette description pourra servir à donner aux utilisateurs un avant-goût de ce qui les attend.

*Documents.* Cet outil fonctionne comme le gestionnaire des fichiers d'un ordinateur. On peut y transférer des documents de tous types (html, Word, Powerpoint, Excel, Acrobat, Flash, Quicktime, etc.). On peut aussi les organiser dans des dossiers que l'on crée. La seule contrainte réside dans le fait que l'utilisateur doit posséder, sur son ordinateur, l'application ou la visionneuse permettant de relire les formats de fichiers mis à sa disposition.

*Liens.* Cet outil permet de constituer une bibliothèque de ressources pour les utilisateurs et en particulier de ressources externes. Dokeos offre la possibilité à l'enseignant d'organiser les liens en catégories afin de faciliter la recherche d'informations par les étudiants.

*Tests.* Le module de tests permet à l'enseignant de créer des tests d'auto-évaluation pouvant contenir un nombre quelconque de questions. Il existe différents types des tests disponibles :

- choix multiple à réponse unique ;
- choix multiple à réponses multiples ;
- remplissage de blancs ;



- appariement c'est-à-dire un exercice visant à établir une correspondance entre une série de propositions et une seconde série de réponses ;
- question ouverte ;
- zone sur image (*hotspot*, par exemple localisation d'une ville sur une carte géographique).

Un test rassemble un certain nombre de questions, pas nécessairement du même type, sous un thème commun.

*Agenda.* L'agenda est un outil qui peut être attaché à chaque cours ou sous forme d'un agenda global propre à chaque utilisateur. Un agenda est attaché à chaque cours pour programmer les différentes tâches. L'agenda global est accessible depuis l'onglet «Mes cours», et reprend les événements des cours dont l'enseignant est responsable. Il peut être enrichi d'événements privés à l'enseignant et qui n'apparaîtront pas dans les cours des apprenants.

*Annonces.* Cet outil permet d'envoyer un message par courriel aux utilisateurs afin de publier une information importante liée à un cours. L'envoi de courriels, constitue un important outil de remise à l'ordre et de relance lorsque l'enseignant voit que les étudiants prennent du retard et que son cours est de plus en plus abandonné. Il peut notamment signaler aux utilisateurs que le concepteur a déposé un nouveau document, que la date de remise des rapports approche ou qu'un des apprenants a réalisé un travail de qualité. En définitive, dans son fonctionnement, cet outil est assez similaire à l'agenda et semble de toute façon être en surplus.

*Forum.* C'est un outil de communication et de discussion asynchrones. La différence par rapport au courriel est que le forum situe la discussion dans un espace public ou semi-public, c'est-à-dire que plusieurs personnes identifiées peuvent y accéder à la fois. L'utilisation de cet outil ne requiert qu'un simple navigateur et il n'ya donc pas besoin d'un d'outil de courriel.

*Partage de fichiers.* Comme l'indique son nom, c'est un outil de partage qui offre la possibilité à l'enseignant d'adresser un fichier à un ou plusieurs apprenants. Il permet

aussi aux apprenants d'envoyer un fichier à l'enseignant ou à leurs collègues, cette dernière option est possible après autorisation expresse de l'administrateur de la plateforme. En outre, les fichiers envoyés peuvent être commentés par les personnes à qui ils sont destinés.

*Utilisateurs.* Cet outil fournit la liste des personnes inscrites au cours. Il sert à gérer les inscrits, à les partager en groupes, à leur donner des rôles ou des responsabilités et à les suivre pendant une activité quelconque.

*Groupes.* Cet outil a pour rôle principal de créer et d'administrer des groupes de travail. Les groupes se créent soit manuellement, soit automatiquement, et peuvent être dotés d'outils propres au groupe (documents, agenda, travaux, annonces, forum).

*Discuter.* L'outil « Discuter » est un outil de communication synchrone.

*Travaux.* Cet outil très simple permet aux apprenants d'envoyer des documents vers le cours. Il peut servir à réceptionner des rapports individuels ou collectifs, des réponses à des questions ouvertes ou toute autre forme de document.

*Outils de suivi.* Cet outil permet de faire le suivi des apprenants durant l'apprentissage de deux façons. La première façon est dite « globale », elle englobe tous les apprenants et tente de donner une vue d'ensemble sur les différentes statistiques concernant le cours en termes de nombre d'apprenants, de temps passé sur le cours, de score moyen des apprenants, de progression moyenne des apprenants, etc. La deuxième façon d'effectuer le suivi est dite « nominative » et concerne le suivi d'un seul apprenant en particulier.

*Maintenance du cours.* Cet outil permet d'assurer la maintenance du cours séparément de ses propriétés<sup>44</sup>, de supprimer ou de purger un cours, de copier tout ou partie d'un cours vers un autre. Ou encore de sauvegarder ou de réimporter un cours déjà sauvegardé. Toutes ces opérations s'effectuent très simplement et rapidement.

---

<sup>44</sup> Les propriétés d'un cours rassemblent les formats des fichiers, leur origine, leur localisation, leur format à l'écran d'affichage etc. que la maintenance ne peut manipuler.



*Enquêtes.* Dans le but d'améliorer le cours et de le perfectionner, cet outil permet d'organiser un sondage au sein des apprenants afin d'avoir leurs avis sur les différents aspects qui concernent le déroulement d'un cours.

*Vidéoconférences.* Une fois paramétré par l'administrateur, l'outil vidéoconférence permet d'établir deux modes d'échanges : les «classes virtuelles» où seul l'enseignant expose, et les «réunions virtuelles» où tous les participants peuvent intervenir.

### 5.2.1.2.3 Notre bilan de Dokeos

Notre utilisation ainsi que l'avis de plusieurs autres utilisateurs à travers le monde donne les avantages suivants à Dokeos :

1. Une mise en œuvre facile, simple et intuitive. Atout de poids en Afrique subsaharienne.
2. Une simplicité et une souplesse d'utilisation concernant l'usage des outils.
3. Elle dispose d'un véritable outil de gestion de plugins sous forme de modules : il suffit d'uploader un fichier zip pour que celui-ci soit installé. Il existe une offre grandissante de fonctionnalités à travers les plugins.
4. Une facilité d'ajouter des fonctionnalités grâce au développement et une remarquable réactivité de l'équipe des développeurs pour fixer les bugs et les failles de sécurité.
5. Une documentation assez riche et complète de la plate-forme disponible en ligne, en français et gratuitement.
6. La fonction «parcours pédagogique» permet de proposer un scénario pédagogique à l'apprenant. Il est présenté sous la forme d'un sommaire très facile à consulter.
7. le service est gratuit et disponible en plusieurs langues.
8. Notre expérience de l'utilisation de Dokeos montre une appropriation assez aisée de ses utilisateurs. Ces raisons font de Dokeos la plate-forme la mieux adaptée au contexte de l'Afrique subsaharienne en général, et celui du Congo en particulier.

Adaptable à différents contextes de formation, Dokeos est utilisé non seulement dans les écoles et les universités, mais également dans les centres de formation, les associations et les entreprises. C'est une plate-forme personnalisable qui offre un environnement de travail flexible et sur mesure. Extrêmement facile d'utilisation tant du côté étudiant que du côté enseignant, elle se caractérise par une prise en main rapide et très intuitive. Ceci en fait l'outil de choix pour l'Université de Lubumbashi. Elle offre une gestion sobre et intuitive des outils et des espaces d'administration. La gestion quotidienne de la plate-forme ne requiert aucune compétence technique particulière. La plate-forme s'installe aisément et l'usage d'un simple navigateur permet de gérer les différents espaces ainsi que les utilisateurs enregistrés. Sa documentation est disponible en ligne et traduite dans plusieurs langues. Elle est extrêmement détaillée et documentée, et des présentations multimédias interactives sont disponibles afin de mieux comprendre la philosophie et la pédagogie de la plate-forme. Enfin, la création d'un consortium de partenaires et d'une conférence d'utilisateurs annuelle montrent les ambitions qu'affichent clairement la communauté Dokeos. Dokeos structure ses développements en modules afin d'assurer une meilleure gestion des mises à jour et de gagner en flexibilité et en souplesse. Dokeos favorise par ailleurs l'apprentissage collaboratif tel que nous l'avons constaté au cours de nos expérimentations et constitue un excellent outil pour compléter l'enseignement en présentiel.

#### 5.2.1.2.4 Les points faibles

Nous énumérons les points faibles de Dokeos comme suit:

Les groupes. : un étudiant une fois inscrit dans un groupe ne peut pas s'en désinscrire. Les dossiers «documents» et «liens» d'un groupe ont une capacité réduite de stockage (7 Mo).

La Messagerie instantanée n'est pas très ergonomique. Ceci dit, on peut maintenant grâce aux modules complémentaires changer le design de l'outil pour en utiliser un meilleur.



Les espaces de travail restent cloisonnés : un gestionnaire d'espace ne peut pas copier les informations qu'il a créées dans un autre espace (planning, annonces, documents, etc.).

Il y a peu d'outils de communication synchrone. L'ajout d'une fonctionnalité comme un tableau blanc ou un outil de visioconférence constituerait une plus-value remarquable pour le travail au sein de Dokeos.

L'interface ergonomique des parcours pédagogiques reste à améliorer, car l'enseignant doit traverser un nombre important d'écrans pour atteindre le contenu. Si la ressource a beaucoup d'items, comme il n'y a pas de menu dynamique, cela fait beaucoup de « cliquer-glisser » et de clics pour suivre le parcours.

Le SCORM n'est pas complètement implémenté par la plate-forme. Certains packages produits par des éditeurs conformes et certifiés SCORM passent parfois difficilement ou en partie seulement sur Dokeos. Comme pour Moodle, une des faiblesses de Dokeos est son intégration très imparfaite des contenus SCORM.

## 5.3 Etude comparative et catégorielle des fonctionnalités Moodle vs Dokeos

Nous avons effectué une étude comparative des fonctionnalités des deux plates-formes selon leurs catégories afin d'avoir une vision de leurs profils et de cerner à quel niveau se trouvent leurs différences réelles par rapport au modèle de référence SCORM. Nous établirons donc, par catégorie, un tableau résumant le parallélisme établi entre les fonctionnalités homologues des deux plates-formes. Cette classification reprend l'étude d'usage dans le site du ministère français de l'éducation<sup>45</sup>. Dans la série de tableaux qui suit, nous illustrons des extraits de cette comparaison ; nous mettons oui ou non pour dire si l'outil existe dans la plate-forme ou pas et lorsqu'il a une caractéristique particulière nous mettons un commentaire dans la cellule. Les tableaux seront repris dans leurs détails en

<sup>45</sup> <http://www.educnet.education.fr>: site du ministère français de l'éducation (consulté en juillet 2011).

annexe. Le modèle des fonctionnalités (outils) émane de l'architecture de la plate-forme WebCT, que nous ne reprenons pas.

### 5.3.1 Outils de communication

Voici un inventaire de la présence ou pas d'outils de communication au tableau 5.1.

Tableau 5.1 : Extrait de la comparaison des outils de communication

| Plate-forme                                 | Moodle | Dokeos |
|---|--------|--------|
| Forum                                       | Oui    | Oui    |
| Messagerie                                  | Oui    | Oui    |
| Awareness (listes des personnes connectées) | Oui    | Oui    |
| Fiche d'identité                            | Oui    | Oui    |

### 5.3.2 Outils d'organisation

La gestion des étudiants, celle des cours et celle des enseignants se font de manière différente sur chacune des plates-formes.

Tableau 5.2 : Extrait de la comparaison des outils d'organisation

| OUTILS D'ORGANISATION                         |   |   |
|---|---|---|
| Plate-forme                                   | Moodle                                    | Dokeos  |
| Agenda commun                                 | Oui                                       | Oui   |
| Agenda personnel                              | Pas très clairement défini                | Oui   |
| Gestion des étudiants (planning, inscription) | Oui, suivi automatique des étudiants.     | Création par l'administrateur ou le formateur des comptes étudiants. Planning proposé par le formateur. |
| Gestion des enseignants                       | Création par l'administrateur des comptes | Création par l'administrateur des comptes   |
| Gestions des cours                            | Gérés par les formateurs                  | Gérés par les formateurs  |

### 5.3.3 Outils de partage.

Les outils des travaux des étudiants, par exemple, sont implémentés aussi de manière propre à chaque plate-forme. Les groupes sont partout créés par l'enseignant du cours, mais les interactions entre membres apprenants du groupe sont très limitées. Les membres d'un



groupe, selon la plate-forme, peuvent se voir entre eux ou non. Cette visibilité peut être encore restreinte à des sous-groupes, pour certains outils de communication qui accompagnent les travaux des groupes d'apprenants. Pour les autres outils de partage la situation est présentée dans le tableau 5.3.

Tableau 5.3 : Extrait de la comparaison d'outils de partage

| OUTILS DE PARTAGE  |   |        |
|--|---|--------|
| Plate-forme  | Moodle  | Dokeos |
| Dépôts de documents  | Oui   | Oui    |
| Mise en ligne de cours   | Oui, structuration du contenu par rubriques. Banque d'exercices, etc. | Oui    |
| Salle de cours (partage tous les documents avec tous les apprenants) | Oui   | Oui    |
| Historique   | Apprendre c'est aussi une activité d'observation.                     | Oui    |
| Cours (audio, visuel...)   | Oui   | Oui    |
| Tableau blanc  | Non   | Non    |
| Ouverture, alertes, flux RSS <sup>46</sup>                           | Possibilité d'afficher des flux RSS de l'extérieur.                   | Oui    |

### 5.3.4 Outils de contrôle

Ce que nous notons, à ce niveau c'est surtout la différence entre les outils de questionnements qui existent, certes, dans toutes les deux plates-formes mais chacune d'entre elles leur donne le format de son choix. Le suivi des étudiants se fait partout mais chacune des plates-formes le fait à sa façon. Elles présentent parfois des formats de questionnement et de suivi des apprenants, conformes à certaines normes à des degrés différents.

Tableau 5.4 : Extrait de la comparaison d'outils de contrôle.

| Plate-forme                     | Moodle  | Dokeos |
|---------------------------------|---|--------|
| QCM (Question à Choix Multiple) | QCM mais aussi des Questions à appariements, etc. | Oui    |
| Résultats et notes              | Oui   | Oui    |
| Statistiques de suivi des cours | Oui   | Oui    |
| Contrôle des connexions         | Oui, accompagné d'un journal détaillé.            | Oui    |

<sup>46</sup> RSS: Really Simple Syndication, Système de mise à jour automatique sur un site des informations venues d'un autre site.

### 5.3.5 Aspects pédagogiques

Nous relevons les fonctions principales de l'enseignant que l'on retrouve dans toutes les trois plates-formes. Il s'agit de : rédiger la description du cours, publier le cours, administrer des forums de discussion, gérer une liste de liens, créer des groupes de participants, composer des exercices, publier des annonces, consulter des statistiques de fréquentation et de réussite aux exercices. Pour ce qui concerne les profils et rôles des différents acteurs, ils sont globalement répartis de la même façon. Le formateur gère les outils pour les rendre visibles, exploitables par les apprenants. Les apprenants suivent les cours et l'administrateur modifie les droits et les profils des utilisateurs si besoin. Un extrait de comparaison est illustré dans le tableau 5.5.

Tableau 5.5 : Extrait de la comparaison des outils de gestion des aspects pédagogiques

| Plate-forme                                     | Moodle   | Dokeos   |
|---|--|--|
| Nombre d'utilisateurs maximum de la plate-forme | Illimité.  | Illimité.  |
| Profils et rôles                                | Les administrateurs peuvent tout faire sur le site, dans tous les cours. Les créateurs des cours peuvent créer de nouveaux cours et y enseigner. | Le Formateur gère les outils pour les rendre visibles et exploitables par les apprenants |
| Évaluation et contrôle                          | Oui  | Oui  |
| Fonctions de l'enseignant                       | Plate-forme mise en place en se fixant pour objectif le socioconstructivisme   | Rédiger la description du cours, publier le cours...                                     |

### 5.3.6 Aspects techniques

Notons que toutes ces plates-formes au niveau techniques sont comparables, elles fonctionnent notamment sur plusieurs systèmes d'exploitation. Elles sont multilingues et Dokeos va jusqu'à l'implémentation de 34 langues. La maintenance des cours est globalement assurée par l'enseignant, tandis que celle de la plate-forme et sa mise à jour se font par l'administrateur. Le tableau 5.6, reprend un extrait de cette comparaison.



Tableau 5.6 : Extrait de comparaison des aspects techniques.

| Plate-forme                        | Moodle  | Dokeos   |
|------------------------------------|---|--|
| Format propriétaire                | Oui   | Non  |
| Personnalisation de la plate-forme | Oui   | Oui  |
| Ergonomie de la plate-forme        | Menu à droite et cours au centre. Navigation facile car l'arborescence de tous les cours apparait sur la page principale. | L'administrateur peut manipuler l'ergonomie de toute la plate-forme, tandis que l'enseignant ne peut configurer que l'ergonomie de son propre cours. |
| Maintenance et mise à jour         | Oui, hebdomadaire.  | Oui  |
| Sécurité                           | Oui, identifiant + mot de passe.  | Oui, identifiant + mot de passe.   |

### 5.3.7 Aspects financiers

Les aspects financiers, par ailleurs importants, pour un projet en Afrique subsaharienne, sont ici encourageants pour Moodle et Dokeos. En effet les deux sont open source et gratuits.

Tableau 5.7 : Comparaison des aspects financiers.

| Plate-forme                       | Moodle   | Dokeos  |
|-----------------------------------|--|---|
| Coût de la plate-forme            | Gratuite, logiciel libre.  | Plusieurs logiciels, l'un gratuit, deux solutions payantes. La tendance est plutôt commerciale. |
| Coût par étudiant                 | Gratuit.   | Gratuit.  |
| Tarif d'inscription de l'étudiant | Gratuit ou suivant la politique locale de l'institution d'accueil. | Politique locale de l'institution de formation.   |

Nous avons cependant noté pour Dokeos qu'une certaine assistance technique serait payante et la vidéoconférence est proposée à part comme module payant. Un extrait de comparaison est donné dans le tableau 5.7.

### 5.3.8 Discussion

A travers tous ces tableaux comparatifs, il apparaît clairement que Moodle et Dokeos sont des outils similaires. Leur philosophie d'organisation reste la même et on retrouve dans 90% des catégories les mêmes fonctionnalités dans les deux plates-formes. La

différence majeure se trouve naturellement dans la taxinomie, dans la présentation ainsi que dans la hiérarchie des fonctionnalités. Ceci nous laisse appréhender deux écoles différentes tant dans la conception de l'outil informatique que dans la vision pédagogique des types d'enseignements à privilégier sur la plate-forme. Au niveau informatique, nous allons nous attarder sur les productions de contenus des deux plates-formes, voir en profondeur ce qui les distingue pour implémenter un outil assurant l'interopérabilité des packages ainsi qu'une compatibilité et un échange automatique des productions entre les outils de communication des deux plates-formes.

Au regard des outils proposés par les deux plates-formes, il apparaît qu'elles prennent en compte toutes les caractéristiques fonctionnelles nécessaires pour favoriser un apprentissage collaboratif avec notamment l'intégration des aspects de gestion des rôles, des droits, des ressources, des outils asynchrones et synchrones, des activités individuelles et collaboratives, de la production etc. En nous appuyant sur les travaux d'Ellis et Wainer, on peut étudier les plates-formes par des fonctionnalités offertes en termes d'outils mis à disposition des apprenants et enseignants. Pour cela, une classification fonctionnelle s'appuyant sur un modèle semblable à celui des trois « C » (Ellis, et al., 1994) ou sur un modèle dérivé (David, 2001; Tarpin-Bernard, 1997) en quatre dimensions, est possible pour nous permettre d'établir un ordre prioritaire à donner aux outils à implémenter dans le prototype informatique d'interopérabilité et prévoir de manière méthodique leur utilité sur terrain à l'UNILU.

Le modèle des trois « C » dérivé en quatre dimensions, repose sur quatre espaces d'interactions:

1. L'espace de production : il propose les fonctionnalités qui permettent une activité pendant laquelle les participants, individuellement ou en groupe, réalisent des travaux.
2. L'espace de coordination : il exprime les relations entre les utilisateurs et leurs activités. La coordination assure l'efficacité du groupe, dans la réalisation de la tâche, en ce sens qu'elle favorise une organisation de la tâche et des acteurs impliqués dans celle-ci. Elle permet la production collective d'un groupe en



assurant la synchronisation relative au travail commun en jugulant de manière remarquable les éventuels conflits possibles qui peuvent apparaître.

3. L'espace de communication : il permet aux participants d'émettre ou de recevoir des données persistantes. Elles sont obligatoirement liées à la tâche du groupe et sont émises à partir d'un espace privé du participant à un ou plusieurs espaces communs ou privés. L'activité de communication est considérée comme un outil au service de la production et de la coordination.
4. L'espace de conversation : il permet aux participants de dialoguer sans s'échanger des données persistantes et est le lieu privilégié des expressions de convivialité entre membres d'un même groupe ou entre utilisateurs conviés à une même tâche. Il s'agit d'une activité informelle qui peut être réalisée de différentes manières synchrone ou asynchrone, textuelle, orale ou visuelle.

D'autres auteurs ont également proposé d'ajouter des dimensions orthogonales aux trois espaces initiaux. Par exemple, Ferraris propose l'espace de régulation qui permet de décrire les règles de coopération entre les individus impliqués dans un apprentissage collectif (Ferraris, et al., 2000). La figure 5.3 illustre les quatre dimensions conçus par David (David, 2001) et décrit les tâches principales associées à chaque cercle et ses intersections. Nous avons numéroté ces différentes parties comme les différents usages possibles pour des apprenants comme des tuteurs, tel que le suggère Laforcade (2006).

L'étude des plates-formes Moodle et Dokeos nous a permis, comme on l'a montré dans les sections précédentes, d'extraire les différents outils pour ensuite les positionner par rapport aux usages. Nous en consignons une partie dans le tableau 5.8.

Au travers nos expérimentations des plates-formes à l'UNILU et tout au long de ce travail, nous avons réalisé combien il est relativement impérieux de catégoriser les outils étant donné les différents usages qui peuvent en être faits. De nombreux outils ne sont pas mentionnés dans le tableau car ils correspondent à des outils similaires à ceux déjà listés. Cependant nous faisons ici un rendu des usages constatés des outils pendant les expérimentations de terrain qui aboutissent à un classement d'outils selon leurs priorités pour une exécution optimale des enseignements en ligne dans le contexte de l'UNILU. Car,

rappelons-le, l'échange des contenus doit être suivi de l'utilisation de certains outils de la plate-forme pour vivifier l'apprentissage.

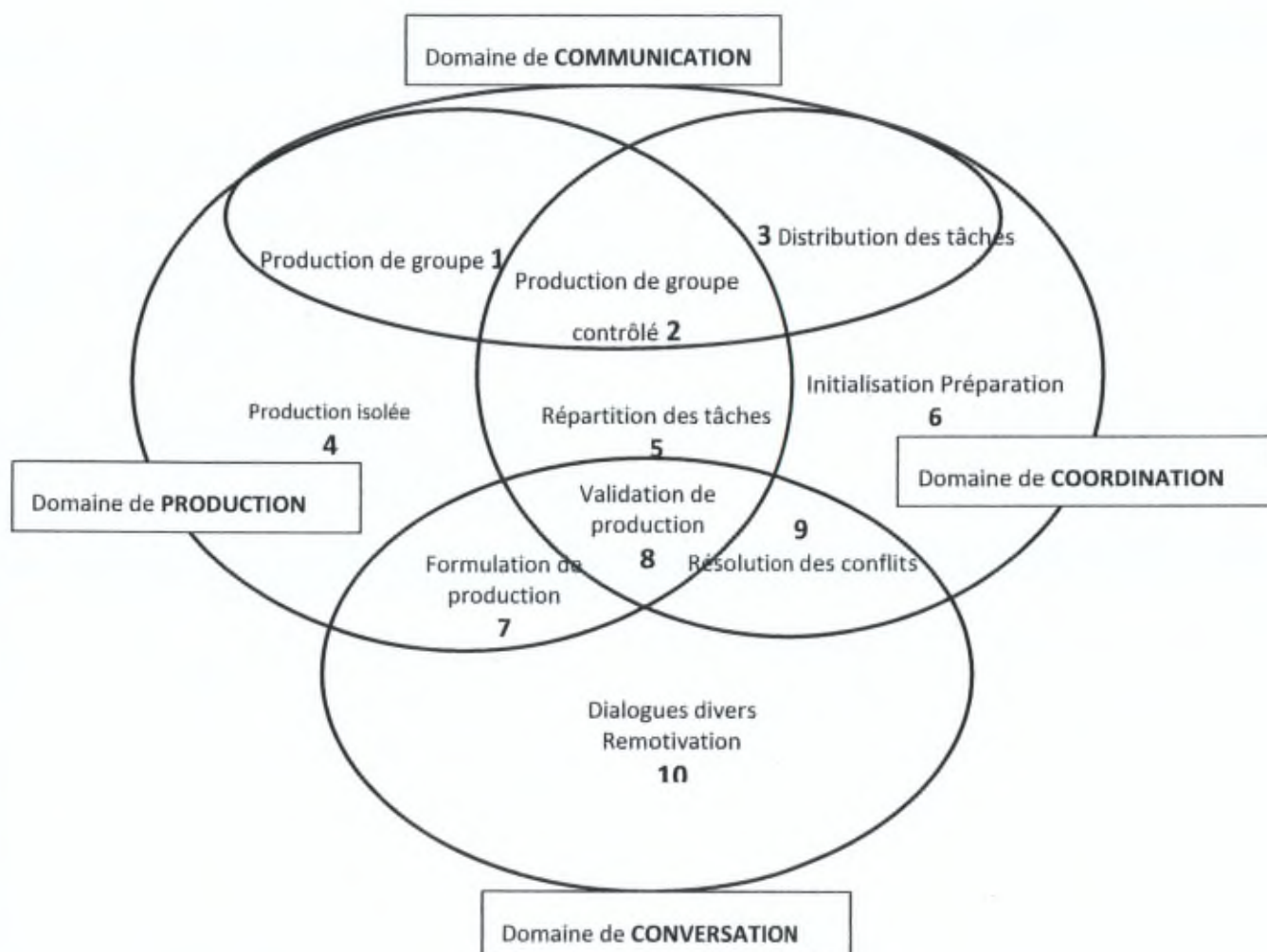


Figure 5.3 : Le modèle en quatre « C » tiré de (David, 2001)

Ces outils participent aussi à l'élaboration d'un scénario narratif qui constituera le modèle de base sur lequel se fonderont tous les projets des sessions d'apprentissage afin de les adapter au contexte de l'UNILU.

Les usages définis dans le modèle des trois « C » sont plus nombreux pour les outils qui ont été les plus utilisés dans nos expérimentations décrits au chapitre 3. On découvre notamment, au-delà de l'effet de mode, l'importance indéniable de la vidéoconférence et de la messagerie électronique qui pourrait servir pour neuf usages différents (tableau 5.8). Elles



sont suivies du forum et de la messagerie instantanée qui peuvent rassembler quatre usages chacun. La prise en compte de l'importance de ces outils devrait présider aux choix guidant la mise en œuvre du prototype d'interopérabilité.

Tableau 5.8 : Outils des plates-formes et usages vis-à-vis du modèle en quatre « C ».

|                           |                     |                          |                            |
|---------------------------|---------------------|--------------------------|----------------------------|
| Chat                      | 7,8,9,              | Partage d'application    | 1,2,3                      |
| Forum                     | 7,8,9,10            | Messagerie instantanée   | 7,8,9,10                   |
| Messagerie électronique   | 7,8,9,10, 1,2,3,5,6 | Vidéoconférence          | 7,8,9,10<br>1,2,3,5,6      |
| Espace de travail partagé | 1,2,3               | FAQ                      | 9                          |
| Quizz, QCM                | 4                   | Présentation de vidéos   | 4,3,6,5                    |
| Liste de diffusion        | 7,8,9,10,1,2,3,5,6  | Tableau blanc            | 1,2                        |
| Accès Web                 | 1                   | Questionnaires, sondages | 1                          |
| Agenda                    | 5,6                 | Tableau de bord          | 5                          |
| Outils d'historiques      | 4                   | Outils de suivi          | 4,1(seulement pour Dokeos) |

Après acquisition des ressources, l'usage des outils adéquats à la session de formation s'impose. Nous concluons en disant que les expérimentations (section 3.4.3.4) ont servi à identifier les outils devant entrer dans l'élaboration d'un cahier des charges conforme à la réalité du terrain. Le modèle des trois «C» vient nous fournir des éléments permettant un classement selon l'ordre des priorités de ces outils identifiés au préalable, par rapport à leur rôle dans la plate-forme. En définitive, les expérimentations du chapitre 3, la comparaison des outils dans les sections précédentes et le modèle des trois «C» nous ont permis respectivement d'identifier les outils adéquats à un apprentissage en ligne à l'UNILU,

de comparer les rôles des outils dans les deux plates-formes, et de vérifier leur existence et enfin d'établir l'ordre de priorité dans lequel on devrait les choisir par rapport aux usages du cahier des charges. Cet ordre de priorité est montré au tableau 5.8, le principe est que plus un outil a des usages, plus il est prioritaire.

## 5.4 Description d'un contenu Moodle

Les cours dans Dokeos et Moodle sont fondamentalement différents et il faudra donc que l'importation soit adaptée au scénario local d'apprentissage concocté à partir des expérimentations de terrain à l'Université de Lubumbashi. Ceci nous donne à la fois la priorité d'utilisation des outils d'accomplissement de l'enseignement ainsi que les tâches qu'ils permettent d'effectuer.

Nous allons analyser de près le contenu d'un package Moodle pour comprendre son mécanisme et préparer l'implémentation de l'outil d'interopérabilité. D'ores et déjà, le premier constat que nous faisons sur les contenus à l'export est qu'ils se présentent comme des fichiers zip avec un certain nombre de répertoires et un fichier jouant le rôle de manifeste. Nous allons mener notre étude suivant deux axes : le premier axe est la composition et l'agrégation du package, et le deuxième concerne le mécanisme d'échange et d'intégration entre la package et la plate-forme.

### 5.4.1 Agrégation d'un Package Moodle

Le package Moodle est constitué du manifeste moodle.xml situé à la racine du container et des fichiers physiques. En pratique, il s'agit en fait d'une sauvegarde faite à un moment donné d'un ou plusieurs cours, en utilisant le lien «*Sauvegarde*» dans le menu d'administration. Ceci crée un fichier compressé (zip) qui peut se télécharger sur le disque dur d'un ordinateur. Ce fichier compressé peut être réimporté ensuite dans n'importe quelle autre installation de Moodle, pour autant que les versions installées de Moodle ne diffèrent pas trop d'un serveur à l'autre, à l'aide du lien «*Restauration*» du menu d'administration. On peut choisir les types d'activités à exporter (forums, ressources, etc.) avec ou sans les données des utilisateurs, par exemple les résultats des sondages, les messages des forums, les notes, les devoirs remis, etc. On peut aussi choisir quels utilisateurs exporter : ceux du



cours, tous ceux du site ou aucun. Enfin, on peut exporter ou non les historiques du cours, les fichiers des utilisateurs, ou encore les fichiers du cours appartenant à l'enseignant.

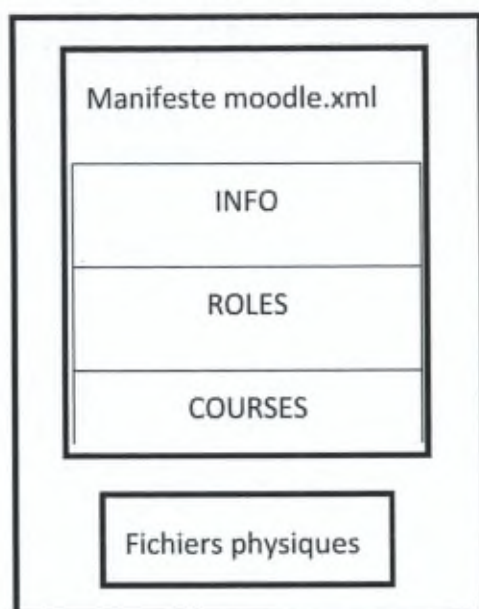


Figure 5.4 : Agrégation d'un package Moodle.

La documentation de Moodle étant très éparpillée et très peu canalisée sur le net, nous ferons ici une description basée essentiellement sur les expérimentations effectuées au cours de nos recherches ainsi que sur la rarissime documentation concernant le développement de Moodle.

#### 5.4.1.1 Le manifeste moodle.xml

Le manifeste est un fichier XML qui constitue un inventaire structuré du contenu du package. Le fichier manifeste représente donc les informations nécessaires à la description du contenu du package. Il comporte dans le cas de Moodle l'élément racine `<MOODLE_BACKUP>` dont les nœuds enfants sont les trois sections principales suivantes `<INFO>`, `<ROLES>`, `<COURSE>` comme illustré à la figure 5.5.

##### 5.4.1.1.1. La section « INFO »

Cette section est constituée d'informations générales qui décrivent le cours comme un tout. Elle comprend plusieurs éléments descendants qui fournissent notamment le nom du cours,

la version de Moodle utilisée, la version de la sauvegarde, la date de création, les informations contenues dans les fichiers, etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<MOODLE_BACKUP>
  <INFO>
    <NAME>backup-aw10-20080226-1115.zip</NAME>
    <MOODLE_VERSION>2006101007</MOODLE_VERSION>
    <MOODLE_RELEASE>1.9.7</MOODLE_RELEASE>
    <BACKUP_VERSION>2006082304</BACKUP_VERSION>
    <BACKUP_RELEASE>1.7</BACKUP_RELEASE>...
  </INFO>
  <ROLES>
  </ROLES>
  <COURSE>
    <HEADER>
      <ID>38</ID>
      <CATEGORY>
        <ID>6</ID>
        <NAME>English for Academic Purposes</NAME>
      </CATEGORY>
      <PASSWORD></PASSWORD>
      <FULLNAME>Academic Words 10</FULLNAME>
      <SHORTNAME>AW10</SHORTNAME>
      <IDNUMBER></IDNUMBER>
      <SUMMARY> course based on the tenth sublist of the
      AWL<br /></SUMMARY>
      <FORMAT>topics</FORMAT>...
    </COURSE>
  </MOODLE_BACKUP>
```

Figure 5.5 : Extrait d'un fichier moodle.xml.

- les descendants de <INFO> : l'élément <INFO> du manifeste possède beaucoup d'éléments descendants pouvant atteindre une profondeur de 5 et nous décrivons ici ses éléments principaux qui sont :

<NAME>. Le nom de la sauvegarde.

<MOODLE\_VERSION> et <MOODLE\_RELEASE>. La version de la plate-forme Moodle.

<BACKUP\_VERSION> et <BACKUP\_RELEASE>. La Version de la sauvegarde d'un contenu d'apprentissage.

<DATE>. La date de la création de la sauvegarde.

<ORIGINAL\_WWWROOT>. Le répertoire racine du serveur Web d'hébergement.



<ZIP\_METHOD>. La méthode de compression du fichier zip : elle est soit interne, soit externe au système.

<DETAILS>. Différents détails de description de la session d'apprentissage en tant qu'un tout. C'est l'élément le plus complet de la section <INFO> du manifeste, car il possède une longue descendance de près de 11 éléments. Nous nous contenterons de citer ici seulement, les détails concernant les utilisateurs, le cours lui-même, les pages Web du contenu, les blogs du contenu, les modules qui constituent le cours, etc. Il s'agit des éléments : MOD, METACOURSE, USERS, LOGS, USERFILES, COURSEFILES, SITEFILES, GRADEBOOKHISTORIES, MESSAGES, BLOGS, BLOCKFORMAT. Notons que nous retrouvons aussi différents modules du contenu.

#### 5.4.1.1.2 La section « ROLES »

Après avoir identifié les informations détaillées sur le contenu d'apprentissage ainsi que sur les utilisateurs sauvegardés, nous allons analyser comment sont sauvegardés les droits et les permissions de chacun des utilisateurs par rapport au contenu. Quelles sont les fonctionnalités de Moodle qui accompagnent l'utilisateur dans une activité quelconque ? Et dans quel contexte du contenu ? Pour arriver à cet objectif, la communauté Moodle définit certains concepts (Cojean, 2007):

- Un *rôle* : il identifie le statut d'un utilisateur dans un certain contexte (par exemple : enseignant, étudiant ou modérateur de forum sont des exemples de rôles) ;
- Une *capacité* : c'est la description d'une fonctionnalité particulière de Moodle. Elles peuvent être associées aux rôles (par exemple, *mod/forum:replypost* est une capacité) ;
- Une *permission* est une certaine valeur qui définit l'usage d'une certaine capacité par un rôle donné. (autoriser, empêcher, etc.) ;
- Un *contexte* est un «endroit» dans Moodle, tel qu'un cours, un module d'activité, un bloc, etc.

Moodle permet à des utilisateurs autorisés de créer des rôles en nombre illimité. Un rôle étant une liste de permissions attribuées à chacune des actions connues dans Moodle. Les rôles peuvent être attribués à des utilisateurs dans un certain contexte.

L'élément **ROLES** du manifeste moodle.xml possède un seul enfant **ROLE** qui lui donne à son tour plusieurs descendants notamment : **ID** pour le numéro d'identification du rôle, **NAME** pour le nom du rôle, **NAMEINCOURSE** le nom du rôle dans le cours, **CAPABILITIES** pour décrire les capacités. A travers l'élément enfant **CAPABILITY** en termes de permissions, des noms, des mises à jour etc. Et les éléments enfants du nœud sont **NAME**, **PERMISSION**, **TIMEMODIFIED**, **MODIFIERID**, **CAPABILITY**.

### 5.4.1.1.3 La section « COURSE »

C'est la plus longue section du fichier manifeste : elle peut posséder jusqu'à plus de 200 descendants servant à décrire un cours ou un module de cours de Moodle. Nous avons catégorisé ces éléments en deux groupes : le groupe des éléments traditionnels, et celui des éléments conditionnels.

#### 5.4.1.1.3.1 Les éléments traditionnels

Les éléments traditionnels sont les éléments suffisants pour décrire le contenu d'un cours, sans parler de son utilisation et des outils qui l'accompagnent.

**HEADER.** C'est le premier élément enfant de **COURSE**. Il constitue l'en-tête qui accompagne toujours la description d'un cours et fournit notamment le numéro **ID** pour identifier le cours, sa catégorie, le mot de passe du cours, le résumé du contenu de cours, l'enseignant, les étudiants, les groupes, l'inscription, les notifications, etc. Et pour cela il possède une grande quantité des descendants notamment **ID**, **PASSWORD**, **FULLNAME**, **CATEGORY**, **SHORTNAME**, **SUMMARY**, **GUEST**, **TEACHER**, **STUDENT**, **COST**, **ENROLLABLE**, **METACOURSE** ;

**METACOURSE.** Pour créer des relations parent-enfant entre cours. Avec des enfants comme **PARENTS**, **CHILD** etc. Un cours de Physique générale par



exemple peut être le parent (Méta cours) des cours suivants, Electricité, Optique, Mécanique, Physique quantique etc.

#### 5.4.1.1.3.2 Éléments conditionnels

Il y a plusieurs façons de sauvegarder un cours : on peut se contenter d'en mettre juste le contenu ainsi que les étapes à suivre pour arriver au bout, comme on peut aussi avoir le souci d'exporter les utilisateurs du cours, les messages associés au cours, ses annonces et toutes les capacités et activités qui l'accompagnent.

Plusieurs éléments spécifient ces options :

**MESSAGES.** Cet élément contient les messages, lus ou non, à sauvegarder avec leurs auteurs et leurs destinataires. Il possède plusieurs descendants.

**BLOGS.** Cet élément répertorie les blogs d'un cours à sauvegarder. Il précise, entre autres, comment les identifier, leur sujet, leur résumé, leur contenu, leurs auteurs, le groupe concerné, ainsi que le module du cours concerné par le blog.

**BLOCKS.** Certaines parties d'un cours peuvent être rassemblées en blocs et être décrites comme un tout, ce qui se précise par cet élément. Il décrit notamment le poids du bloc (nombre d'octets), sa visibilité (qui a le droit de le voir parmi les utilisateurs), sa position dans le cours, son nom, etc. Certains éléments de description sont : NAME, WEIGHT, VISIBLE, POSITION, PAGETYPE, etc. ;

**SECTIONS.** Le cours peut aussi être divisé en différentes sections et modules de parcours, dont cet élément par son enfant SECTION sert à décrire le résumé, la visibilité de la section ainsi que son numéro. Et ses quelques éléments enfants sont : SUMMARY, VISIBLE, ID, MODS, etc.

**FORMATDATA.** Cet élément fournit le format des données constituant le cours (présentation powerpoint, fichiers PDF, documents textes, etc.).

**USERS.** Cet élément traite des utilisateurs en les décrivant notamment à travers leurs coordonnées (téléphone, nom, adresse, e-mail), leurs rôles dans le

cours, leurs préférences, etc. Et USERS possèdent plusieurs descendants parmi lesquels on peut citer USER qui est le seul enfant. Les autres comme CONSTITUTION, MSN, CONFIRMED, AIM, PHONE, COUNTRY, FIRSTACCESS, etc. sont des descendants de USER.

LOGS. Cet élément reprend le suivi des apprenants à travers leurs connexions au cours, le temps passé sur un module, l'heure du début de la connexion, l'heure de la fin de la connexion, ou encore le module concerné par la session. Il a un élément enfant qui est LOG et qui à son tour a une suite des descendants qui décrivent la session : USERID, TIME, MODULE, IP, ACTION, INFO etc. ;

GRADEBOOK. Cet élément s'appuie sur une longue série des descendants pour classer les différents modules des cours par rapport à un sujet ou une thématique précise traitée par le cours. Il spécifie aussi le niveau d'études ou les compétences à atteindre à l'issue d'une formation.

Nous recensons aussi les éléments SCALES qui donnent les étapes à suivre, EVENTS qui décrivent les événements du cours ou du module, GROUPS qui s'occupent des groupes, GROUPINGS, GROUPINGSGROUPS, pour les groupements, MODULES pour traiter des modules.

#### 5.4.1.2 Bilan de moodle.xml

Nous venons de voir l'agrégation d'un contenu Moodle ainsi que l'examen de son manifeste. Le fonctionnement de ce dernier s'articule sur trois axes : la description du cours comme un tout (avec l'élément INFO), la répartition des rôles des utilisateurs et fonctionnalités de Moodle accompagnant le cours (avec l'élément ROLES) et enfin la description détaillée du cours lui-même en modules (catégories) ainsi que les adjuvants aux différentes activités qui accompagnent le cours (avec l'élément COURSE).

### 5.5 Description d'un contenu Dokeos

La présentation d'un contenu de Dokeos relève, d'après notre usage de l'outil et la documentation officielle de Dokeos (Millet, et al., 2009), deux aspects d'exportation et



d'importation. Le premier est que la sauvegarde en vue d'une exportation d'un simple cours débouche sur un fichier zip que l'on stocke sur le disque dur et qui constitue une archive des documents du cours. Certains outils ayant servi pendant le cours tels que «Agenda», «Wiki», «Partage de fichiers», «Travaux» sont exclus de la sauvegarde par défaut. Le second est que lors de l'exportation, si le cours concerné est accompagné d'un «parcours pédagogique», cette fois-là Dokeos propose de l'exporter sous format SCORM.

### 5.5.1 Description du fichier de sauvegarde d'un cours.

Certains outils de Dokeos sont aptes à exporter ou à importer des données (documents, utilisateurs, tests ou Quiz, travaux etc.). Quelques outils importent ou exportent les données présentées sous forme des fichiers CSV ou XML. D'autres importent et exportent des contenus sous forme des fichiers archives zip sans autres spécifications.

### 5.5.2 Description du fichier d'export au format SCORM de Dokeos

La philosophie de Dokeos ne prévoit pas, à la base, la conception d'un cours comme un tout agencé, manipulable et déplaçable à souhait. Le cours est présenté depuis sa page d'accueil, via notamment une description, des documents utiles à l'apprentissage, des travaux à effectuer, un forum accompagnateur, etc. Cette présentation poursuit le but d'agencer les différents éléments du cours pour une session de formation. Ces éléments sont en définitive matérialisés par des outils qu'un apprenant peut utiliser suivant un rythme convenu avec l'enseignant ou voulu par ce dernier. L'enseignant est donc libre d'indiquer les conditions d'évolution des enseignements en utilisant à sa guise les outils, d'imposer les contraintes d'accessibilité et de visibilité pour amener l'apprenant à cheminer dans le sens préconisé par lui pour un cours donné.

Cependant, tout devient différent à partir du moment où l'enseignant d'un cours décide d'un scénario d'apprentissage rythmé selon un outil nommé «parcours d'apprentissage». Un parcours d'apprentissage, ou parcours pédagogique au sens de Dokeos, est une séquence d'apprentissage découpée en chapitres eux-mêmes découpés en étapes. Il peut être organisé en fonction d'un contenu. Il constituera alors une sorte de table

des matières. Il peut aussi être organisé en fonction d'activités. Il s'apparentera alors à un agenda de « choses à faire » pour acquérir la maîtrise d'un savoir ou d'une compétence. Les étapes successives du parcours peuvent être nommées « semaines », « modules », « séquences » ou avec toute autre appellation répondant à la nature du scénario d'apprentissage envisagé par l'enseignant. En plus d'être structuré, un parcours peut être séquencé. Cela signifie que certaines étapes peuvent constituer des pré-requis pour d'autres : par exemple on n'atteint pas l'étape n° 2 sans être passé par l'étape n° 1 par exemple (Pecquet, 2007).

Une séquence peut être suggestive en montrant les étapes l'une après l'autre ou contraignante en obligeant l'étudiant à suivre les étapes dans un ordre imposé. Il est important de comprendre qu'un parcours est plus que le découpage d'une matière : il est un itinéraire à travers le savoir qui inclut potentiellement des épreuves, des temps de discussion, d'évaluation, d'expérimentation, de publication, de regards-croisés, etc. C'est pourquoi l'outil de parcours de Dokeos constitue une sorte de « méta-outil » qui utilise de nombreux autres outils pour les séquencer ou les ordonner afin d'atteindre les objectifs d'apprentissage. En définitive, le parcours d'apprentissage est une sorte d'outil intégrateur des autres outils comme expliqué plus haut qui permet de faire d'un cours un tout cohérent exprimant une entité de parcours conçue par l'enseignant pour permettre l'exécution d'un programme de cours selon sa vision. Dokeos prévoit qu'un cours ainsi constitué peut être archivé et exporté conformément à la norme SCORM.

Mais cette conformité à la norme SCORM pose des problèmes que nous allons identifier pour comprendre quelle est la constitution d'un package Dokeos à partir d'un parcours pédagogique.

### 5.5.3 Constitution d'un package Dokeos :

Le package Dokeos est une unité logique de formation héritée, comme nous l'avons souligné à la section précédente, du méta-cours « parcours pédagogique ». Il peut véhiculer un morceau de cours ou une formation complète contenant plusieurs cours. Il contient toutes les informations indispensables à l'utilisation des contenus pour une séquence



d'apprentissage donnée et se présente comme un contenu SCORM, c'est-à-dire un fichier zip avec un manifeste XML à sa racine. Le manifeste du package Dokeos s'appelle, comme dans la spécification IMS, *imsmanifest.xml*. Il est la plupart du temps accompagné d'autres fichiers faisant office de sous-manifeste (par exemple *ims\_xml.xsd*, *ims\_qtiasiv1p2.xsd*), et de fichiers physiques constituant la séquence d'apprentissage représentée. Quant au manifeste, conformément au modèle SCORM, il est un inventaire structuré du contenu du package et contient donc les informations nécessaires à sa description. Il comporte quatre grandes sections à l'exemple de ce que nous avons vu dans la section 4.5.1 du chapitre précédent dans le *Content Aggregation Model* du modèle SCORM :

<metadata>. Cet élément contient des métadonnées décrivant le manifeste. Il contient les informations appropriées qui décrivent le contenu d'un package comme un tout. Il est souvent considéré comme le nœud parent pour des métadonnées qui définissent un package comme un tout. Il fournit également la version SCORM à laquelle Dokeos est compatible.

<organizations>. Cet élément contient la structure du contenu ainsi que la provenance du contenu, en l'occurrence ici « *dokeos\_scorm\_export* ».

<Resources>. Cet élément définit les ressources d'apprentissage empaquetées en termes d'identification des SCO et actifs (voir sections 4.5.1.1 et 4.5.1.2) constituant le package.

Un cours conforme à la norme SCORM est capable d'établir un dialogue avec la plateforme d'accueil. Ce dialogue peut rester basique, c'est à dire le cours informe la plate-forme que telle étape est commencée ou terminée, ou être plus évolué, par exemple le cours récupère en plus le nom de l'étudiant qui étudie le cours, son groupe, etc. La pratique nous apprend pour l'instant que la conformité de Dokeos au modèle SCORM reste basique et que les cours exportés ne portent pas d'aussi grands renseignements que ça. Certaines données n'apparaissent pas ou ne sont tout simplement pas reconnues.

```

<?xml version="1.0" encoding="UTF-8" ?>
<manifest xmlns="http://www.imsproject.org/xsd/imscp_rootv1p1p2"
  identifier="SingleCourseManifest" version="1.1"
  xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_rootv1p2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsproject.org/xsd/imscp_rootv1p1p2
    imscp_rootv1p1p2.xsd http://www.imsglobal.org/xsd/imsmd_rootv1p2p1
    imsmd_rootv1p2p1.xsd http://www.adlnet.org/xsd/adlcp_rootv1p2
    adlcp_rootv1p2.xsd">
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>1.2</schemaversion>
  </metadata>
  <organizations default="dokeos_scorm_export">
    <organization identifier="dokeos_scorm_export">
      <title>Bases de données 1</title>
      <item identifier="ITEM_1" identifierref="RESOURCE_1"
        invisible="true">
        <title>Définir la modélisation</title>
        <adlcp:prerequisites type="aicc_script" />
        <adlcp:masteryscore />
        <item identifier="ITEM_11" identifierref="RESOURCE_11"
          invisible="true">
          <title>Langage SQL</title>
          <adlcp:prerequisites type="aicc_script" />
          <adlcp:masteryscore />
        </item>
      </item>
    </organization>
  </organizations>
  ...

```

Figure 5.6 : Extrait d'un fichier manifeste Dokeos.

Dokeos précise en outre que lors de l'exportation d'un cours tous les documents exportables au format SCORM se trouvent dans le fichier archive au format SCORM à l'exception des données de certains outils tels : «Travaux», «Groupes», «Partage de fichiers» et «Utilisateurs».

## 5.6 Caractérisation des conditions de compatibilité

Nous avons à faire clairement à deux types de contenus qui doivent être interchangeables d'une plate-forme à une autre. Cependant, l'échange des contenus ne suffit pas pour assurer une interopérabilité Moodle-Dokeos entre l'Université de Lubumbashi et ses partenaires belges. Il faudrait permettre à l'enseignant belge d'être capable de suivre ses étudiants de l'UNILU à partir de la plate-forme de son université d'origine, que nous supposons ici être Moodle ou Dokeos comme développé à la section 5.1. Faire le suivi implique bien entendu l'interaction des outils de communication, d'échange des documents et de suivi pédagogique, entre les deux plates-formes.



Afin de certifier un outil informatique comme artefact favorisant un réel apprentissage, et répondant aux exigences pédagogiques, notre prototype informatique devrait remplir un certain nombre de critères pédagogiques pour l'apprentissage en ligne. Ainsi Miao recense dans ses travaux diverses exigences d'implémentation pour un environnement informatique support à l'apprentissage (Miao, 2000) et, en son sens, cet environnement devrait être :

1. Un support de l'orientation sociale pour permettre à chaque utilisateur de connaître sa position dans le scénario, où il peut aller, comment rejoindre ses collègues, où acquérir de nouvelles informations, et quels outils utiliser.
2. Un support de la conscience de groupe (*awareness*) pour permettre à chaque utilisateur de se présenter aux autres et de connaître les autres utilisateurs et ce qu'ils font.
3. Un support de riches formes d'interactions sociales pour permettre aux utilisateurs d'interagir entre eux sous forme de coopération à des lieux ainsi que des moments différents ou identiques.
4. Un support de la personnalisation des environnements d'apprentissage pour permettre aux utilisateurs de personnaliser leur environnement d'apprentissage pour réaliser différents types de tâche.
5. Un support de la représentation de formes variées d'idées pour permettre aux utilisateurs de représenter des formes variées d'informations ainsi que leurs intentions.
6. Un support de la représentation des relations entre ces idées.
7. Un support de la mise à disposition d'informations de référence : permettre aux utilisateurs de lier une information à une autre information relative ou plus détaillée.
8. Un support de la négociation des connaissances partagées, en ce sens que lors d'un apprentissage collaboratif, pendant les interactions sociales entre co-apprenants, ils doivent disposer d'une représentation non seulement de leurs propres connaissances (métacognition), mais également des connaissances de leur partenaire.

9. Un support de la définition de rôles et, par extension, des responsabilités ou obligations attribuées.
10. Une mise à disposition d'aide ou directives selon les stratégies des enseignements : il s'agit de guider les apprenants et les tuteurs dans le suivi des activités d'apprentissage, en accord avec la stratégie pour laquelle des situations distinctes et leurs transitions sont spécifiées.
11. Un support de la synchronisation des activités collaboratives pour permettre aux utilisateurs de toujours réaliser leur tâche principale et d'éviter des comportements inattendus.
12. Un support du changement entre différentes stratégies d'apprentissage pour permettre aux utilisateurs de définir leurs propres stratégies d'apprentissage et de pouvoir en changer.
13. Un support de la définition de plan d'action pour permettre aux utilisateurs de définir des actions et leurs relations.
14. Un support de l'allocation de ressources.
15. Une décharge de la difficulté des apprenants à réaliser leurs plans d'actions (support à l'organisation de la tâche).
16. Un support pour l'exécution des plans d'actions définis par l'utilisateur.

Dans la pratique, l'auteur souligne évidemment que les environnements informatiques ne répondent pas de manière exhaustive à toutes ces exigences, mais s'efforcent de proposer un certain nombre de fonctionnalités correspondant à un sous-ensemble de ces critères. Pour nos travaux, nous nous focaliserons plus sur les fonctionnalités susceptibles de favoriser une ambiance à la fois collective et collaborative. Ainsi nos expérimentations ont débouché sur un scénario narratif qui montre quels outils sont indispensables pour un déroulement optimal des enseignements en ligne à l'UNILU conformément au contexte local. Dans la suite, après la comparaison des outils des plateformes, le modèle des trois « C » est venu corroborer nos résultats lors de nos différents usages des outils. Il nous a permis de mieux appréhender leur hiérarchie lors du déroulement d'une session d'apprentissage en ligne. Enfin, nous nous appuyons sur les



exigences de Miao pour que notre prototype informatique d'interopérabilité réponde au maximum aux exigences favorisant un apprentissage en ligne qui soit collaboratif et collectif, en permettant à l'enseignant d'avoir des moyens d'effectuer un suivi à distance tout en restant conforme aux standards pédagogiques garantissant un déroulement acceptable des enseignements du point de vue pédagogique.

Notre prototype d'interopérabilité aura pour objectif à long terme de faire interagir les fonctionnalités suivantes entre de Moodle et Dokeos:

- les outils de communication synchrone ;
- les outils de communication asynchrone ;
- les outils de communication synchrone et visuelle (Visioconférence) ;
- les outils d'*awareness* indiquant quel autre utilisateur est en ligne ;
- les outils d'organisation des tâches Agenda ;
- les outils d'échange de documents divers ;
- les outils de conservation des traces de l'évolution d'un étudiant ;
- les outils d'identification des acteurs concernés par la séquence d'apprentissage.

Signalons qu'à ce stade, dans le cadre d'une étude de faisabilité, nous mettons un accent particulier uniquement sur les outils de communication asynchrone.

## 5.7 Méthodologie pratique de mise en œuvre

Les deux plates-formes sont presque similaires dans leurs philosophies et la grosse différence se trouve dans la nomination des métadonnées ainsi que leur mise en œuvre au sein du manifeste XML, *moodle.xml* pour Moodle et *imsmanifest.xml* pour Dokeos (voir tableau 5.9). Notre future application aura donc comme principale la traduction de l'ensemble des métadonnées Moodle, nommées *Moodlecore* par certains développeurs de la communauté Moodle, en métadonnées du *Content Agregation Model* de SCORM héritée de l'*IMS Content Packaging* et du LOM. Les métadonnées ici ont des équivalences contextuelles qu'il va falloir exploiter dans l'implémentation de l'application d'interopérabilité.

Il ne suffit pas de transférer les données d'une plate-forme à une autre, encore faut-il un suivi des étudiants. Nous focaliserons dès lors, nos travaux sur les outils de communication. En effet le suivi des étudiants, le travail collaboratif et la communication sont souvent invoqués comme des procédés à valeur ajoutée favorisés par des outils de type plate-forme (Ecoutin, 2001). Ce constat nous l'avons d'ailleurs fait dans le scénario prévisionnel narratif que nous avons pu élaborer lors de nos expérimentations à l'UNILU (section 3.4.3.2).

De ce qui précède, l'interopérabilité Moodle-Dokeos au niveau technique, devrait atteindre les objectifs suivants :

1. Du côté Dokeos, contribuer à approfondir, un tant soit peu, la compatibilité de Dokeos au modèle SCORM en apportant un plus à l'outil d'exportation au format SCORM.
2. Du côté Moodle, trouver une équivalence entre la taxinomie des métadonnées de *moodle.xml* et celle du LOM utilisée dans le modèle CAM de SCORM et implémentée dans Dokeos.
3. Créer un outil qui possède des modules de traitement SCORM, d'envoi des packages, de réception des packages et de stockage des packages.
4. Créer tous les outils nécessaires permettant une transmission automatique des éléments de communication synchrones et asynchrones entre Moodle et Dokeos pour permettre le déroulement des enseignements et un suivi à distance des étudiants. Nous donnerons une priorité à la messagerie instantanée<sup>47</sup> et au forum qui sont des outils qui nous sont imposés par le contexte local des enseignements.

La compatibilité de Moodle et Dokeos au modèle SCORM s'arrête pour l'instant aux cours et aux exercices. Leur compatibilité à SCORM manque de maturité pour se révéler réellement fiable. Entre ces deux plates-formes, de manière générale, une interopérabilité efficace et complète est très limitée visiblement pour des raisons

---

<sup>47</sup> Le cas de la messagerie instantanée n'a pas été implémenté dans le présent travail.



stratégiques stimulées par la concurrence et un souci des parts du marché des plates-formes (CSC, 2003). Il faut reconnaître que la concurrence ne favorise pas l'interopérabilité.

Tableau 5.9 : Les sections principales des fichiers *imsmanifest.xml* et *moodle.xml*.

| Éléments d' <i>imsmanifest.xml</i>   | Éléments de <i>moodle.xml</i>  |
|--|--|
| <metadata> contient des métadonnées décrivant le manifeste. Il contient les informations appropriées qui décrivent le package content comme un tout.                     | <INFO> est constitué d'informations générales qui décrivent le cours comme un tout.  |
| <Organizations> contient la structure du contenu.  | <ROLES> s'occupe de savoir quels sont les droits et permissions de chacun des utilisateurs par rapport au contenu, et quelles sont les fonctionnalités de Moodle qui accompagnent l'utilisateur dans son cursus. |
| <Ressources> définit les ressources d'apprentissage empaquetées en termes d'identification des SCO et assets constituant le package ainsi que leurs liens de dépendance. | <COURSE> décrit en détail le cours, son échelonnement ainsi que ses utilisateurs.  |

Dès lors, l'interopérabilité pour notre cas d'étude se limite à une démarche en deux temps : transférer les contenus d'une plate-forme à l'autre, et assurer un suivi effectif des étudiants. De là notre implémentation se divise en deux volets. Le premier volet, *l'interopérabilité statique*, s'occupe du déplacement et de la compatibilité des contenus d'une plate-forme à une autre. Le second volet, *l'interopérabilité dynamique*, établira une synchronisation en temps réel des outils de communication des deux plates-formes à distance. Cela pour un suivi effectif, efficace et efficient des étudiants. Le chapitre suivant nous révèle une proposition de mise en œuvre de cette interopérabilité entre Moodle et Dokeos.

L'avant-projet de création de l'application d'interopérabilité s'appuiera principalement sur les langages XML et ses technologies, surtout pour le volet statique de l'interopérabilité, et sur UML (*Unified Modeling Language*) qui est un langage pour visualiser, spécifier, construire et documenter tous les aspects et artefacts d'un système logiciel (OMG, 2003).

## CHAPITRE 6 : IMPLEMENTATION DE L'INTEROPERABILITE STATIQUE

### 6.1. Introduction

Nous avons défini l'interopérabilité statique comme étant, au niveau import/export des deux plates-formes Moodle et Dokeos, la capacité des contenus de l'une et de l'autre à être reconnus et utilisés par les deux à la fois. Plus précisément, l'interopérabilité peut être définie fonctionnellement et techniquement comme « *la capacité d'échanger des données entre systèmes multiples disposant de différentes caractéristiques en terme de matériels, logiciels, structures de données et interfaces, et avec le minimum de perte d'information et de fonctionnalités* » (NISO, 2004). Nous travaillerons donc dans ce chapitre à rendre possible un des aspects de l'interopérabilité, à savoir la migration des contenus en les transformant en contenus SCORM, puis en améliorant la capacité de chacune des plates-formes à intégrer des contenus SCORM.

Dès lors, la question est comment y arriver techniquement ? Notons d'abord que les contenus se présentent tous sous un fichier de format zip à la racine duquel se trouve un manifeste XML qui décrit la manière d'utiliser et d'identifier le contenu. Le contenu est à son tour décrit à l'aide des métadonnées. Les métadonnées à leur tour sont récupérées à travers le manifeste XML, par le mécanisme de *XML Binding* qui permet notamment d'implémenter dans un fichier XML toute la structure et l'identification des données. Les données de base à échanger en e-learning sont des objets pédagogiques qui peuvent avoir plusieurs natures : il peut s'agir de programmes d'enseignement, de transparents, de notes de cours, de pages Web, de logiciels de simulation, d'objectifs pédagogiques qui peuvent être utilisés, réutilisés ou référencés durant toute l'activité liée à l'enseignement ou à l'apprentissage. Pour obtenir une description plus précise des objets pédagogiques en termes d'informations sur les documents, sur les auteurs, sur leurs champs d'intérêt, leurs idées, leurs objectifs pédagogiques, on peut inclure des métadonnées comme ressources elles-mêmes qui sont structurées suivant des catégories ou champs sémantiques à l'exemple des neuf catégories des métadonnées du LOM illustrées à la section 5.2.4 (Bourda, et al., 2003).



Le langage XML est conçu pour structurer logiquement le contenu des documents avec pour objectif de pouvoir séparer le fond et la forme (nous proposons en annexe quelques concepts du XML). Il apparaît comme le moyen le plus adéquat pour décrire à la fois les métadonnées et la structure des documents afin d'assurer la pérennité de la documentation électronique. Plusieurs chercheurs pensent d'ailleurs que le XML soutient la manière de produire, de gérer et d'utiliser les objets pédagogiques (Gorga, 2003; Bourda, et al., 2003). Le langage XML est le format d'implémentation privilégié de nombreux standards de métadonnées pour assurer la pérennité et l'interopérabilité des données à l'intérieur d'un système ou entre systèmes. L'utilisation de standards s'impose, sur les éléments de métadonnées utilisés, sur leurs valeurs et sur leurs formats d'implémentation (Morel-Pair, 2007). Le consortium IMS pour sa part reconnaît que, par rapport à d'autres formats d'implémentation, le XML standard soutenu par le W3C, donne aux métadonnées leur puissance opérationnelle en matière de recherche, de gestion et d'interopérabilité (IMS, 2001).

Nous traiterons de la question des objets pédagogiques, puis nous considérerons quelques concepts liés au XML accompagnant notre travail, le XML lui-même étant expliqué en annexe. Ensuite nous étudierons les exigences de compatibilité SCORM et enfin nous procéderons à la transformation des contenus pour être à la fois compatibles SCORM et utilisables par les plates-formes.

## 6.2 Les Métadonnées des objets pédagogiques

### 6.2.1 Définitions et caractéristiques des métadonnées des objets pédagogiques

Rappelons qu'une métadonnée est une donnée sur une donnée. Pour l'apprentissage en ligne, elles permettent de décrire les ressources, ainsi que les relations entre ces ressources. Simard pour sa part stipule que les métadonnées *«sont des informations ajoutées à un texte ou un logiciel afin de donner des informations sur son contenu»*. Dans le cas d'un fichier informatique par exemple, le nom, la taille et la date de création représentent des métadonnées de ce fichier (Simard, 2002).

Aussi, les métadonnées sont dotées d'une nature ambivalente et polymorphe dans le sens où elles peuvent être utilisées pour décrire n'importe quelle donnée, objet, événement, ou personne dans le monde (Hodgins, 2002). Cependant, en matière de métadonnées, certains auteurs distinguent deux catégories principales. D'une part, les métadonnées objectives qui ont vocation de décrire des propriétés physiques telles que la date, l'auteur, les identifiants, etc. Elles peuvent la plupart du temps être générées automatiquement (Cardinaels, et al., 2005). D'autre part, les métadonnées subjectives qui relèvent des attributs plus variables et importants, elles sont librement créées et déterminées par une personne ou un groupe de personnes. Dans un contexte d'apprentissage en ligne, les métadonnées doivent aider la recherche d'objets pédagogiques à partir de critères didactiques et faciliter la mise en œuvre de l'apprentissage (Greenberg, 2002).

Pour Duval, afin de mieux remplir leur mission, les métadonnées doivent présenter les quatre caractéristiques suivantes (Duval, et al., 2002) :

1. La modularité permet de combiner des fragments provenant de différents schémas de métadonnées établis ou de différents vocabulaires en évitant toute ambiguïté ou incompatibilité. Les espaces de nommage servent à déterminer l'origine de l'ensemble des éléments de métadonnées qui sont alors liés par des règles et des conventions déterminées par un organisme de maintenance spécifique (`xmlns:adlcp = http://www.adlnet.org/xsd/adlcp_v1p3`, est l'espace de nommage des métadonnées de SCORM maintenue par le consortium ADL).
2. L'extensibilité donne la possibilité de créer des ajouts structurels aux schémas de métadonnées de base pour des applications spécifiques ou des besoins spécifiques d'une communauté d'utilisateurs.
3. Le raffinement donne la possibilité de créer des extensions sémantiques. Dans le cadre d'une application spécifique. Par exemple, les termes illustrateur, auteur, sculpteur, etc. pourront être utilisés à la place du terme plus général « créateur ». Le raffinement permet aussi de préciser le format utilisé pour exprimer une date ou



autoriser l'emploi d'un vocabulaire contrôlé, comme la classification décimale de Dewey (Smith-Yoshimura, et al., 2007).

4. Le multilinguisme rend possible l'expression des métadonnées dans différentes circonstances linguistiques ou culturelles. Il faut ici faire la distinction entre ce qui doit être lu par un utilisateur et ce qui doit être traité informatiquement.

Pour nos travaux, les métadonnées du LOM repris dans le modèle SCORM pour l'apprentissage en ligne remplissent déjà les critères ci-dessus car elles sont réputées modulaires, extensibles, raffinées et multilingues. Ceci constitue une motivation supplémentaire dans le cadre de l'UNILU à concevoir des moyens authentiques pour la mise en œuvre d'une interopérabilité des contenus que nous appelons ici interopérabilité statique.

## 6.2.2 A quoi servent les métadonnées des objets pédagogiques

De l'avis de la *National Information Standards Organisation*, les métadonnées assurent plusieurs fonctions (NISO, 2004) :

- l'identification de manière unique d'une ressource ;
- la découverte de ressources. Ces dernières peuvent être recherchées à partir de critères, elles peuvent être identifiées, localisées, rassemblées autour de critères communs ;
- l'organisation des ressources par sujet ou par audience ;
- l'interopérabilité grâce à des descriptions pouvant être comprises aussi bien par un humain que par une machine ;
- la garantie de l'archivage et de la préservation de la ressource en s'assurant que cette dernière ne tombera pas dans l'oubli et qu'elle continuera à être accessible dans le futur.

Plus particulièrement en e-learning, elles permettent aux systèmes, aux plateformes, aux applications et aux utilisateurs de gérer et d'atteindre un contenu d'apprentissage constitué d'objets pédagogiques sans avoir à interagir avec lui (Nilsson, et

al.). Pour l'interopérabilité statique, l'interchangeabilité concerne ici les contenus des plates-formes Moodle et Dokeos. En définitive, au niveau statique, la manipulation se fait sur du contenu d'apprentissage constitué d'objets pédagogiques décrits par les métadonnées.

## 6.3. L'objet pédagogique

Nous passons par la notion de l'objet pédagogique pour comprendre comment exporter les contenus de Moodle et de Dokeos, car toutes les productions des deux plates-formes ont pour vocation l'apprentissage et sont donc à ce titre des objets pédagogiques et possèdent les caractéristiques que nous allons développer ici.

### 6.3.1 Définition de l'objet pédagogique

La notion d'objet pédagogique ne fait pas l'unanimité quant à sa définition, chacun y allant de son point de vue, de son centre d'intérêt ainsi que de ses compétences. Le terme « objet pédagogique », de l'anglais *learning object*, est apparu de manière formelle pour la première fois à partir de 1994 lorsque Wayne Hodgins a baptisé le groupe de travail de l'association CEdMA<sup>48</sup> « *Learning Architectures, APIs and Learning Objects* » (Hodgins, 2002). Nous avons choisi quelques définitions pour nous permettre de mieux appréhender la question. Selon le groupe LTSC de l'IEEE, un objet pédagogique est défini comme « *toute entité, numérique ou non, qui peut être utilisée pour l'enseignement ou l'apprentissage* » (IEEE, 2002). Plus tard, il enchérira en arguant : « *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, systèmes d'enseignement à distance, environnements d'apprentissage collaboratifs)* » (IEEE, 2002). Pour Strijker, les objets pédagogiques se définissent comme des « *entités numériques, utilisables et réutilisables dans différentes situations pédagogiques* » (Strijker, 2004). L'Université du Wisconsin quant à elle définit les objets pédagogiques comme de « *petites unités d'apprentissage autonomes en ligne. Ils sont suffisamment petits pour être intégrés à une activité pédagogique, une leçon, un module ou un cours* » (Wisc-Online, 2007). Cet échantillon de définitions pris sur

<sup>48</sup> The Computer Education Management Association



tant d'autres laissent transparaître clairement la difficulté dans la communauté e-learning de se mettre d'accord sur une approche commune de définition. Lorsqu'on considère la définition du IEEE-LTSC, elle ne distingue pas le numérique du non numérique et embrasse tout ce qui peut *être utilisé* pour l'apprentissage, tel un auditorium de cours par exemple. Strijkers, pour le même concept, exclu complètement tout ce qui n'est pas numérique. Quant à l'Université du Wisconsin, elle se montre très restrictive en réduisant d'avantage la granularité des unités d'apprentissage en ne considérant que ce qui est « petit ». En fait comme le dit David Wiley, en définitive, il ya autant de définitions du terme « objet pédagogique » que des gens qui l'utilisent (Wiley, 2000).

D'autres chercheurs comme Pernin établissent que ce grand nombre de définitions s'explique par les tensions présentes autour du concept d'objet pédagogique (Pernin, 2003). Ces tensions sont illustrées par la figure 6.1. Au niveau économique, il s'agit de diminuer les coûts de production sans pour autant diminuer la qualité. Cela implique d'avoir des composants réutilisables et partageables. Au niveau pédagogique, le développement conséquent de la formation tout au long de la vie et la mise en place de parcours individualisés, à la carte, nécessitent d'avoir des composants réutilisables et surtout adaptables. Enfin au niveau technique, l'intérêt de l'approche par objet, largement mise en avant dans le développement informatique, n'est plus à démontrer : elle rend possible la réutilisation de composants dans de multiples contextes.

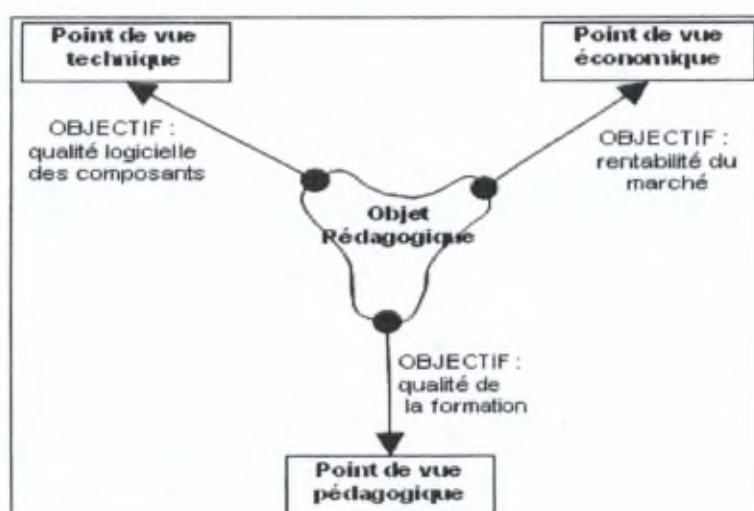


Figure 6.1 : L'objet pédagogique, un concept au centre de fortes tensions (Pernin, 2003).

Selon Pernin, il existe trois principales classes d'objets pédagogiques :

- les unités d'apprentissage qui permettent de structurer la formation et de l'organiser dans l'espace et dans le temps ;
- les activités pédagogiques qui définissent les modalités précises d'acquisition, de validation ou de communication d'une ou plusieurs connaissances ;
- les ressources pédagogiques, physiques ou numériques, nécessaires à la réalisation des activités.

## 6.3.2. Les caractéristiques de l'objet pédagogique

### 6.3.2.1 L'intention pédagogique

Les auteurs orientés pédagogie, tels que Polsani, pensent que c'est l'intention pédagogique qui détermine le statut d'objet pédagogique (Polsani, 2003). Un même objet mis dans deux contextes différents d'apprentissage ne produit pas les mêmes effets attendus. En prenant, par exemple, une image d'un orphelin de guerre amaigri dans une conférence internationale des organisations caritatives, elle ne produira pas la même émotion que dans un cours de santé publique sur la malnutrition.

### 6.3.2.2 La granularité

La granularité d'un objet pédagogique détermine la grandeur spatiale de l'objet et constitue un élément majeur de son intégration dans un cursus. Il peut partir d'une entité aussi petite qu'une simple phrase, un paragraphe explicatif, une image, un graphique, une animation, etc. En fait le flou persiste quant à savoir quelle est la taille maximale d'un objet pédagogique. On peut pressentir à ce niveau que l'objet pédagogique peut aller d'une simple entité, jusqu'au cursus complet. Hodgins définit cinq niveaux de granularité différents (Hodgins, 2002) :



- l'élément brut de données ( niveau de granularité le plus bas) correspond à des éléments de contenu situés purement au niveau des données, autrement dit l'élément physique de base constitue l'entité la plus petite ;
- l'objet d'information se focalise sur une information simple qui peut servir à expliquer un concept, illustrer un principe ou décrire une procédure ;
- l'objet d'application est un ensemble d'objets d'information qui se focalisent sur un objectif pédagogique unique (par exemple un chapitre de cours) ;
- l'assemblage s'étend à des objectifs pédagogiques plus larges, il correspond aux leçons ou aux chapitres ;
- la collection (niveau de granularité le plus élevé) correspond à des cours ou même à des cursus.

### 6.3.2.3 La réutilisabilité

Dès 1965, l'idée de la réutilisabilité apparaît avec Nelson qui établit un cadre permettant de construire du contenu à partir d'objets réutilisables issus de librairies électroniques en interconnexion préalable (Nelson, 1965). L'idée de composants réutilisables peut être appliquée au monde éducatif car une fois créé, un objet pédagogique doit servir dans différents contextes pédagogiques. Ces composants doivent être autonomes, peuvent être produits séparément, mais doivent pouvoir être modifiés pour correspondre aux besoins des utilisateurs. Certains auteurs comme Baumgartner reviennent sur le fait que la granularité d'un objet pédagogique est inversement proportionnelle à sa grandeur : les objets pédagogiques de petite granularité sont dénués de contexte et sont facilement réutilisables, alors que les objets pédagogiques de forte granularité sont très contextualisés pour des raisons pédagogiques et deviennent de fait très difficiles à réutiliser (Baumgartner, 2004).

### 6.3.2.4 L'agrégation

Un objet pédagogique peut certes être réutilisé tel quel, mais il peut aussi être créé par agrégation d'autres objets pédagogiques de granularité plus fine. Il répond ainsi au besoin d'appropriation de la part de l'enseignant, et de mise en contexte pour répondre aux

besoins spécifiques du public cible d'apprenants. Comme le montre la figure 6.2, le mécanisme d'agrégation en général ne peut se limiter à un assemblage d'objets pédagogiques existants. Elle peut aussi intégrer un objet B modifié de son état de départ et un objet D créé et ajouté pour l'occasion afin d'obtenir un nouveau format d'agrégation.

L'agrégation attire l'attention pour nos travaux d'autant plus qu'au niveau statique de l'interopérabilité, le déplacement des contenus (agrégats d'objets pédagogiques) demandera de les extraire de leur plate-forme d'origine pour les agréger ensuite de manière compatible avec la plate-forme d'accueil. Le modèle d'agrégation auquel nous recourrons pour nos travaux est le CAM (*Content Agregation Model*) du modèle SCORM (voir la section 4.5.1). Les objets pédagogiques y sont décrits avec des métadonnées issues du LOM de l'IEEE. Le dispositif informatique pour assurer l'interopérabilité devra aussi implémenter ce mécanisme global d'agrégation.

Steinmetz et Rensing vont même beaucoup plus loin dans l'utilisation de l'agrégation en appliquant le principe de réadaptation (Steinmetz, et al., 2007). Sur la figure 6.3, l'objet E est réadapté en un objet F. Il est tout d'abord décomposé (ou segmenté) en sous-objets A, B, C et D. Les objets A et C sont ensuite modifiés, l'objet D est écarté, la recomposition finale de l'objet F est accompagnée d'une modification de sa navigation. Ceci est pertinent pour nos travaux car le principe de réadaptation permettra à certains contenus Moodle ainsi qu'à leurs métadonnées de description d'être réadaptés afin de les rendre compatibles au modèle SCORM à un degré acceptable. Le déplacement de contenu, malgré ses apparences, n'est pas un processus linéaire. Il peut s'avérer complexe puisqu'on va d'un paradigme à un autre, d'un contexte à un autre et d'un système à un autre. Dans notre cas, l'interopérabilité Moodle Dokeos demandera une rémodularisation des contenus, une adaptation et une permutation des éléments de données dans le but d'obtenir une agrégation des contenus à la fois compatibles aux deux plates-formes.

### 6.3.2.5 L'accessibilité

Il est indispensable de pouvoir retrouver facilement un objet pédagogique. Il doit être étiqueté avec des métadonnées pour être stocké et référencé dans une base de données. Ce



processus est appelé «indexation». L'accès à un objet pédagogique est efficace lorsque le coût engendré par sa recherche en vue de sa réutilisation est inférieur au coût nécessaire pour créer un objet équivalent.

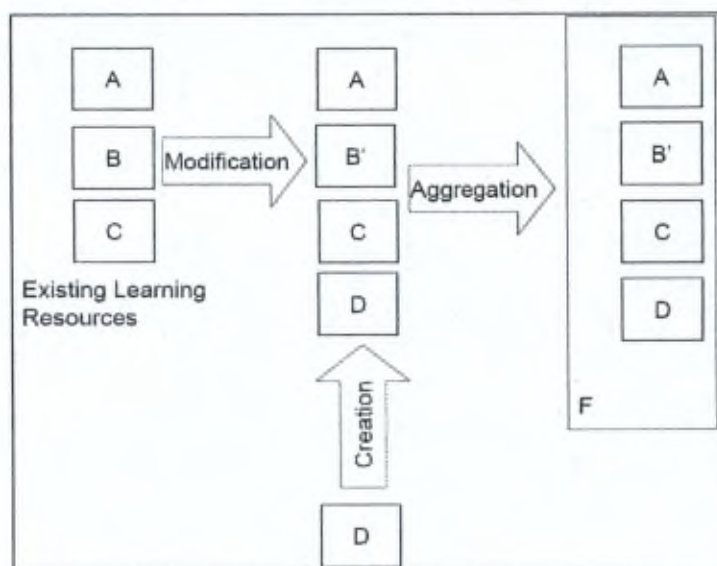


Figure 6.2 : Le principe d'agrégation (Steinmetz, et al., 2007)

L'objet pédagogique doit être diffusé le plus largement possible, ce qui rend nécessaire la possibilité d'échanger et de communiquer entre les systèmes de stockage. La qualité et la quantité des métadonnées jouent ici un rôle important.

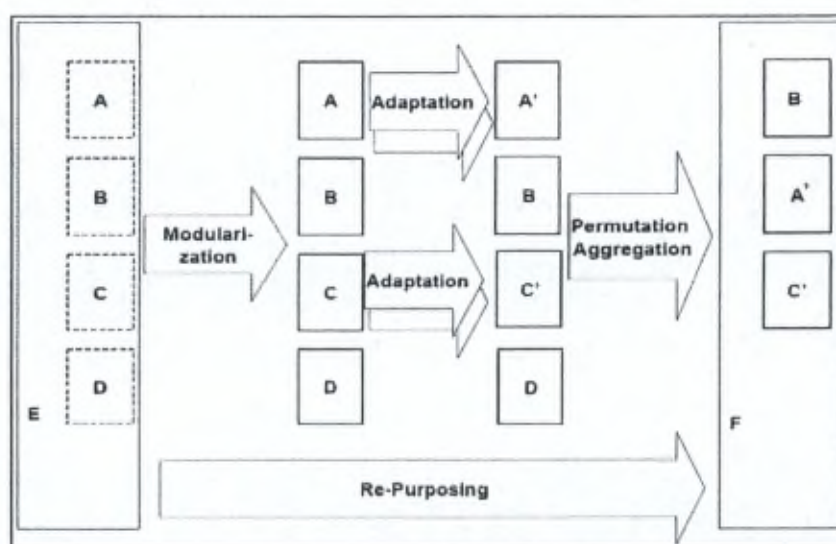


Figure 6.3 : Le principe de la réadaptation (Steinmetz, et al., 2007)

### 6.3.2.6 L'interopérabilité

Comme nous l'avons précédemment martelé dans ce travail, un objet pédagogique doit être autonome, c'est-à-dire indépendant du support de diffusion et de la plate-forme d'apprentissage. Cela soulève la notion d'interopérabilité qui passe nécessairement par la mise en place des standards et des normes.

Nous avons parcouru au troisième chapitre, et plus encore au quatrième, les standards les plus reconnus en matière des métadonnées et plus spécifiquement, pour l'apprentissage en ligne. Le LOM est le standard le plus utilisé, promu notamment dans le modèle SCORM avec la composante CAM (*Content Aggregation Model*). Il définit une structure garantissant l'interopérabilité de la description d'un objet pédagogique, structure sur laquelle nous travaillons pour l'implémentation de l'interopérabilité statique, c'est-à-dire l'échange des objets pédagogiques entre Moodle et Dokeos.

### 6.3.3 SCORM et LOM

Pour chaque niveau de composant, c'est-à-dire une ressource numérique élémentaire, un objet de contenu partageable et un agrégat de contenu, ou plus techniquement un actif (asset) ou un SCO, SCORM propose de définir un sous-ensemble de métadonnées issues du LOM, comme suit (Pernin, 2003):

Pour les ressources numériques, il s'agit de fournir des informations indépendantes du contexte d'utilisation pédagogique (par exemple, le format, la taille, la localisation). Ces métadonnées doivent faciliter la réutilisation et la recherche de telles ressources essentiellement durant la phase de création d'objets pédagogiques partageables, par exemple grâce à un catalogue spécialisé.

Pour les objets pédagogiques partageables, il s'agit de fournir des informations d'ordre pédagogique indépendamment d'un contexte particulier d'agrégation (par exemple, durée d'apprentissage, public cible, âge de l'apprenant). Ces métadonnées doivent faciliter la réutilisation et la recherche de tels objets,



essentiellement durant la phase de création d'objets pédagogiques partageables, également grâce à un catalogue spécialisé.

Pour les agrégats de contenu, il s'agit de fournir des informations relatives à l'agrégation des contenus (par exemple, structure, niveau d'agrégation). Ces métadonnées doivent faciliter la réutilisation et la recherche de telles unités grâce, là encore, à un catalogue spécialisé.

De façon pratique, SCORM établit une table de correspondance pour chacun des niveaux, en indiquant la nature (obligatoire, optionnelle ou «en attente») de chacun des 45 éléments (et leurs sous-éléments) définis du LOM (voir figure 6.4). Nous utiliserons ce principe lors de la transformation des métadonnées Moodle.

Plus concrètement, pour nos travaux, les métadonnées Moodle devront trouver leurs équivalents dans le standard LOM, mis en place, comme on le sait déjà, par l'IEEE en 2002 (IEEE, 2002; Grandbastien, 2002). Ce choix s'explique parce qu'il est le plus utilisé, et qu'il s'appuie en grande partie sur les travaux initiés par la fondation ARIADNE depuis 1997 (Duval, et al., 2002). Il permet de décrire un objet pédagogique à l'aide de métadonnées et favorise le partage et la réutilisation des objets pédagogiques. Son schéma de métadonnées comprend neuf catégories : général, cycle de vie, méta-métadonnées, technique, pédagogique, droits, relation, commentaire, classification. La figure 6.4 offre une représentation de l'ensemble des descripteurs de ce standard. Pour répondre aux besoins particuliers des utilisateurs, le standard LOM peut être spécialisé à travers des profils d'application.

## 6.4 XML éléments théoriques

### 6.4.1 Le langage XML

L'eXtensible *Markup Language* (XML) est un langage dérivé d'un langage développé dans les années 80, le SGML (*Standard Generalized Markup Language*). SGML est un langage généralement complexe à apprendre et à utiliser au quotidien. Il a été jugé trop complexe pour le Web, le HTML a donc été développé. Mais ce dernier, malgré de nombreuses

adaptations, est trop rigide et ne peut répondre aux besoins de plus en plus nombreux et complexes des développeurs Web. Il lui était essentiellement reproché de ne pas séparer la présentation des données de leur contenu. Le XML fut créé pour séparer le contenu des données de leur présentation. XML est un langage de description de documents structurés.

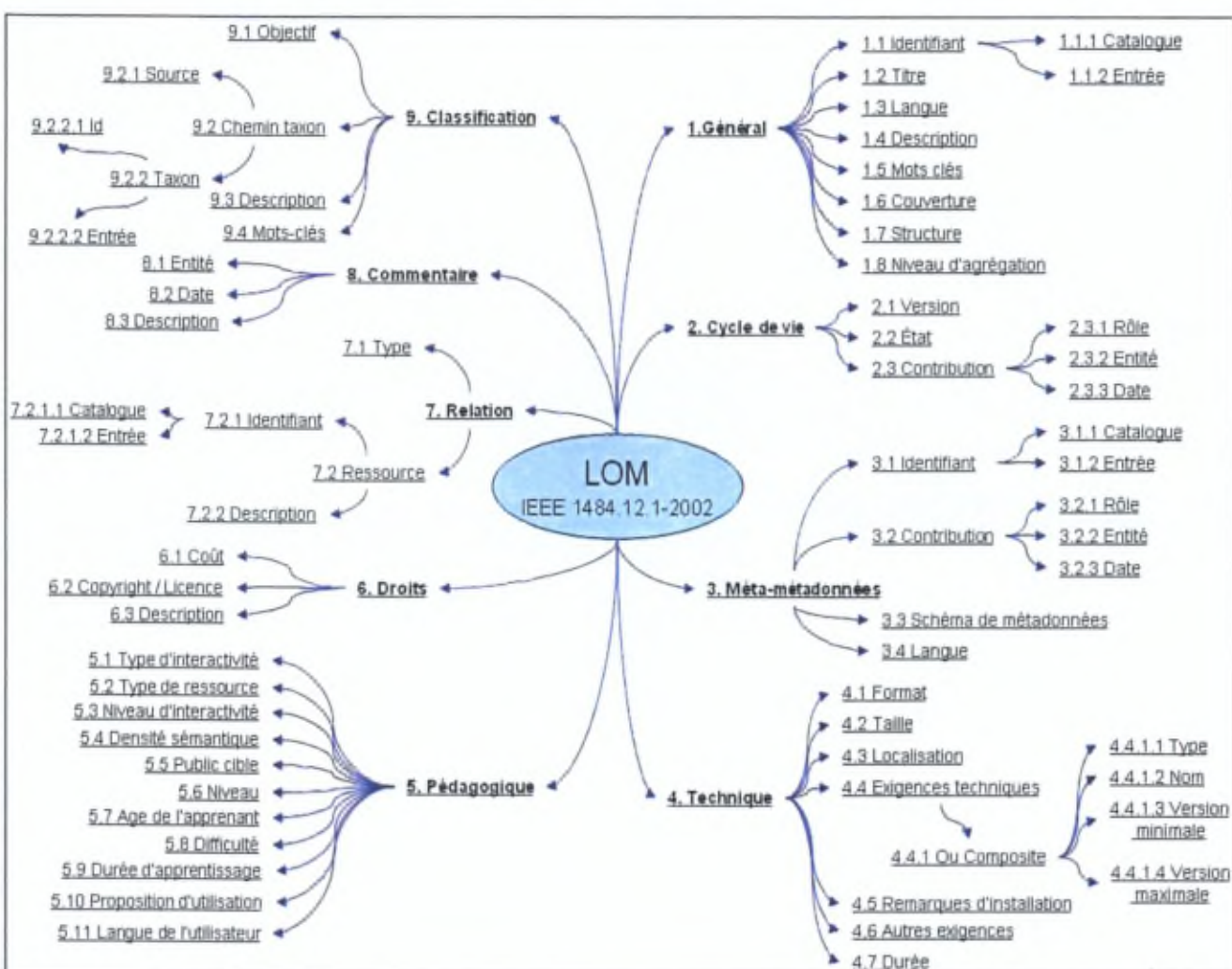


Figure 6.4 : Le Standard LOM v1.0 (IEEE, 2002).

Il permet de décrire la structure logique des ressources. En s'appuyant sur un système à balisage libre, XML permet de marquer les éléments qui composent la structure et les relations entre ces éléments (Bernd, 2001). XML apparaît actuellement comme le moyen le plus adéquat pour décrire, dans la perspective d'une diffusion sur le Web, à la fois les métadonnées et la structure des documents. XML structure les documents de manière



logique et arborescente. Le document XML est un arbre de nœuds correspondant à des éléments ayant une signification sur le plan structurel et sémantique.

C'est uniquement une structuration du contenu, sans propriétés de mise en page. Diverses présentations d'un même document peuvent être générées ensuite (HTML, PDF, etc.). Le rôle de chaque élément, le type de la valeur, les liens entre éléments sont précisés par des attributs. XML est un standard ouvert qui s'impose progressivement pour l'échange des documents. Intégralement basé texte, il est indépendant des formats des fichiers binaires ou des systèmes d'exploitation. Il s'associe à n'importe quel jeu de caractères, notamment Unicode. Il est donc pérenne pour le stockage de documents à long terme et interopérable. XML possède les avantages suivants :

1. En séparant le contenu de la présentation, XML a la possibilité d'afficher des données de différentes façons.
2. La modularité et la réutilisation des structures types : XML permet de définir librement la structure type d'un document.
3. La recherche des documents devient plus structurée et plus puissante.
4. Interopérabilité : les documents provenant de plusieurs sources peuvent être intégrés et manipulés par différentes applications.

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest>
  <organizations>
    <imsld:learning-design
319E87676D680FDEF45AFDF3B418A74A" level="A" uri="">          identifier="LD-
      <imsld:components>
        <imsld:roles
C920AA1D0C1C001E90AB9DAAF15A632B">                          identifier="LD-
          <imsld:learner
BECA5066F13942B1E0ADC96D9BB04148">                          identifier="LD-
            <imsld:title>Learner</imsld:title>
          </imsld:learner>
        </imsld:roles>
      </imsld:components>
    </imsld:learning-design>
  </organizations>
  <resources />
</manifest>
```

Figure 6.5 : Exemple d'un fichier XML.

## 6.4.2 XML et usages du document

### 6.4.2.1 Usages du document et XML

Le Consortium W3C pour le traitement et l'utilisation intégrés du document XML a créé les standards XPath (*XML Path*), XSL (*XML StyleSheet*), DOM (*Document Object Model*), XQuery (*XML Query*) qui sont des briques de fonctions XML. Ils sont normalisés ou en cours de normalisation et parfois doublés par des mécanismes créés par d'autres acteurs contribuant aussi à la promotion de ces standards. Nous ne parlerons pour ce travail que du DOM, que nous avons largement utilisé.

### 6.4.2.2 Traitement DOM

DOM (*Document Object Model*) correspond à une conception orientée objet de l'arbre-document et de ses nœuds. Le parseur DOM, composé de bibliothèques de fonctions, permet de créer et transformer les documents XML directement au travers d'un langage de programmation. Les fonctions DOM et XSLT sont proches, et le choix de l'un ou l'autre dépend surtout de l'environnement informatique et de la formation de celui qui implémente. Pour l'interopérabilité statique dans nos travaux nous avons choisi de combiner les deux. Ainsi le parseur DOM à travers ces bibliothèques de fonctions permet entre autres de :

1. Extraire la valeur d'un élément. Dans DOM, toute chose est un nœud. Les éléments nœuds n'ont pas de valeur textuelle. Le texte d'un élément nœud est stocké comme enfant de ce nœud, et le texte à son tour constitue un nœud texte. Pour avoir donc le texte d'un élément il faut obtenir la valeur du nœud enfant (nœud texte). La propriété `nodeValue` est utilisée pour avoir la valeur textuelle d'un nœud. La fonction `getAttribute()` retourne la valeur d'un attribut.
2. Changer la valeur d'un nœud. La propriété `nodeValue` est utilisée pour changer la valeur d'un nœud. La fonction `setAttribute()` sert à changer la valeur d'un attribut.



3. Déplacer un nœud. Quand un nœud est déplacé, tous ses enfants sont déplacés avec lui. La fonction `removeChild()` déplace un nœud donné et la méthode `removeAttribute()` déplace un attribut donné.
4. Remplacer un nœud. La fonction `replaceChild()` remplace un nœud spécifique et la propriété `nodeValue` remplace le texte dans un nœud texte.
5. Créer un nœud. La fonction `createElement()` crée un nouvel élément nœud, et pour créer un nœud texte on utilise la fonction `createTextNode()`. La création d'un attribut est assurée par la fonction `createAttribute()`.
6. Ajouter un nœud. Pour ajouter un nœud dans un nœud existant, on utilise la fonction `appendChild()`. Le nœud ajouté est placé juste après un nœud enfant existant déjà au sein de l'élément. Au cas où la position du nœud s'avère importante, on peut la fixer à l'aide de la fonction `insertBefore()`.
7. Copier un nœud. La fonction `cloneNode()` crée une copie d'un nœud donné. Elle possède un paramètre booléen (`true` ou `false`) qui indique si le nœud copié pourrait inclure les attributs ainsi que les nœuds enfants du nœud original.
8. L'objet `XMLHttpRequest` est utilisé pour échanger des données avec un serveur. Cet objet aide notamment les développeurs pour :

- mettre à jour une page Web sans avoir besoin de recharger la page ;
- faire des requêtes à partir d'un serveur après avoir chargé la page ;
- recevoir des données à partir d'un serveur après avoir chargé la page ;
- envoyer des données à un serveur à l'arrière-plan.

## 6.5 Transformation contenu Moodle en contenu SCORM

Dans cette section, nous examinons le *SCORM XML binding* qui décrit les ressources et les lie aux métadonnées dans un fichier XML conforme au modèle SCORM. Ensuite nous examinerons les exigences de SCORM pour qu'un contenu lui soit conforme. Enfin, nous montrerons la mise en œuvre de notre interopérabilité statique.

## 6.5.1 SCORM XML Binding

Scrutons à présent les métadonnées de SCORM, comprenons les lois auxquelles elles obéissent, leurs significations et la manière de les poser dans un fichier XML. Ceci dans le but de trouver pour chaque métadonnée de Moodle sa correspondante dans SCORM.

### 6.5.1.1 Présentation de SCORM XML Binding et équivalences

#### Moodle/SCORM

Le consortium ADL décrit la manière d'attacher un fichier manifeste XML à un package de contenu, nommée *XML binding*. Il décrit autant le contenu que le cursus de cheminement pour l'acquisition du savoir promu par l'unité d'apprentissage concernée. Il y a donc des règles spécifiques créées pour implémenter le *XML binding* :

- Le *XML binding* devra adhérer à la spécification XML 1.0 du consortium W3C ;
- Le *XML binding* doit maintenir la structure d'*IMS Content Packaging Information Model*.

Le fichier XML ainsi créé devrait obéir au schéma XSD, tel que spécifié sur le site Web officiel du consortium ADL, que nous illustrons dans l'exemple de la figure 6.6.

```
<manifest identifier="SAMPLE1" version="1.3" xml:base="mescours"
  xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
  xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1
imscp_v1p1.xsd
http://www.adlnet.org/xsd/adlcp_v1p3 adlcp_v1p3.xsd">
  <!-- imsmanifest contents -->
</manifest>
```

Figure 6.6 : Extrait d'un fichier manifeste.

L'implémentation se traduit donc dans un fichier manifeste nommé, comme nous l'avons vu au troisième chapitre, *imsmanifest.xml* dont la construction obéit à quelques règles et conventions parmi lesquelles :



1. Le *Manifest Multiplicity Requirements* qui regroupe toutes les exigences d'occurrence ou de multiplicité des éléments XML du fichier manifeste *imsmanifest.xml* tel qu'illustré dans le tableau 6.2.
2. Les types de données acceptées dans le *XML binding* sont détaillés dans le tableau 6.3.
3. Le *SCORM Content Packaging Application Profile* qui stipule aussi les exigences du profil d'application (voir tableau 6.4). Le profil «*resource*» correspond à un agrégat des fichiers physiques simples constituant un cours, tandis que le profil «*Content Agregation*» reprend non seulement les ressources du cours mais aussi des informations concernant les utilisateurs et le cursus prédéfini par le concepteur du cours.

Après les règles générales à respecter pour la constitution du fichier *imsmanifest.xml*, nous allons examiner, à titre d'illustration, les cas des éléments de la première couche du fichier. Pour les niveaux des descendants de l'élément racine, la méthodologie de transformation reste la même. Nous appliquerons aussi les règles d'agrégation de la méthodologie de Steinmetz développée à la section 6.3.2.4 pour transformer les métadonnées de Moodle vers SCORM. Enfin, nous donnerons pour chacun de ces éléments son équivalent Moodle.

Tableau 6.2 : Exigences de multiplicité des éléments du SCORM *XML binding*.

| Recommandation de Multiplicité<br>(Multiplicity Requirement) | Explication   |
|--|---|
| 1 and only 1   | L'élément doit exister une et une seule fois dans un élément parent.            |
| 0 or More  | L'élément peut ne pas exister ou exister plusieurs fois dans un élément parent. |
| 1 or More  | L'élément doit exister au moins une fois dans l'élément parent.                 |
| 0  | L'élément n'est pas autorisé.   |
| 0 or 1   | L'élément doit exister au plus une fois dans l'élément parent.                  |

Tableau 6.3 : Extrait des types de données du XML binding.

| Types de données | Description   |
|------------------|---|
| Container        | Ce type de données sert à indiquer que l'élément contient des sous-éléments et ne contient aucune valeur de données.              |
| String           | Il s'agit d'un ensemble de caractères. Une limite du nombre de caractères peut être indiquée.                                     |
| Timespan         | Il s'agit d'une longueur du format de temps en heures, minutes, secondes HHHH : MM : SS.SS.                                       |
| ID               | Objet qui sert à identifier de manière unique un élément au sein d'une instance d'un document XML.                                |
| IDRef            | Une référence à un ID.  |
| Boolean          | Une valeur booléenne, 'true' ou 'false'.  |
| Vocabulary       | Une liste restreinte des vocabulaires des éléments. La valeur d'un élément devrait être conforme à celle consignée dans la liste. |
| AnyURI           | Il désigne n'importe quel URI   |
| decimal          | Un décimal  |

Tableau 6.4 : Profil d'application.

| SCORM Content Packaging Application Profile | Exigences de Multiplicité des éléments au sein du manifeste |
|---|---|
| Content Agregation                          | <Requirement>   |
| Resource                                    | <Requirement>   |

#### 6.5.1.1.1 L'élément <manifest>

L'élément <manifest> représente une unité d'apprentissage qui englobe des références aux métadonnées, aux éléments d'organisation et aux ressources elles-mêmes. Cet élément est le nœud racine du fichier *imsmanifest.xml*. Tous les autres éléments portant le nom <manifest> compris au sein de l'élément racine du fichier *imsmanifest.xml* ont pour rôle de compartimenter les fichiers, leurs métadonnées ainsi que leurs structures d'organisation pour leur agrégation et réutilisation. Tous les espaces de nom devraient être également déclarés au sein du fichier manifeste lui même. On définit donc les règles suivantes pour <manifest>.

1. Espace de nom XML : [http://www.imsglobal.org/xsd/ims\\_cp\\_v1p1](http://www.imsglobal.org/xsd/ims_cp_v1p1)
2. Préfixe de l'espace de nom XML: imscp



3. Représentation de l'élément dans le XML binding : <manifest>
4. Exigences SCORM (voir tableau 6.5)

Tableau 6.5 : Exigences de l'élément <manifest> dans les modes d'agrégation CAM.

| SCORM Content Packaging Application Profile | Exigences de Multiplicité des éléments au sein du manifeste |
|---|---|
| Content Aggregation                         | 1 et 1 seule fois.  |
| Resource                                    | 1 et 1 seule fois.  |

5. Les types de données : <manifest> est un élément container, il n'a donc pas de valeur associée et peut posséder les attributs suivants :
  - Identifier (obligatoire) : C'est un attribut obligatoire qui identifie le manifeste. Il doit donc être unique. Il est donné par l'auteur durant le développement du manifeste. Et son type de données est xs: ID.
  - version (optionnel) : c'est un attribut optionnel qui identifie la version du manifeste, son type de données est xs: string. c'est donc une chaîne de caractère qui doit avoir maximum 20 caractères.
  - xml: base (optionnel) : C'est un attribut qui donne un chemin d'accès compensatoire pour les fichiers contenus dans le manifeste. Il est du type xs: anyURI et peut avoir maximum 2000 caractères.
6. Les éléments du nœud racine du manifeste sont : <metadata>, <organizations>, <resources>, <manifest>, <imsss:sequencingCollection>.
7. L'équivalent de l'élément <manifest> dans Moodle est <MOODLE\_BACKUP>.

#### 6.5.1.1.2 L'élément <metadata>

L'élément <metadata> contient les métadonnées qui décrivent le manifeste. Il contient toutes les informations appropriées qui décrivent le contenu comme un tout. Il est

le nœud racine de toutes les métadonnées du contenu ce qui sous entend que toutes les métadonnées de description du contenu devrait être ses enfants. On définit donc les règles suivantes pour <metadata>.

- 1. Espace de nom XML: [http://www.imsglobal.org/xsd/imscp\\_v1p1](http://www.imsglobal.org/xsd/imscp_v1p1).
- 2. Préfixe de l'espace de nom XML: imscp.
- 3. Représentation de l'élément dans le XML binding : <metadata>.
- 4. Exigences SCORM : les exigences de SCORM sur l'intégration de l'élément <metadata> dans l'élément <manifest> sont décrites dans le tableau 6.6.

Tableau 6.6 : Exigences de l'élément <metadata> dans les modes d'agrégation CAM.

| SCORM Content Packaging Application Profile | Exigences de Multiplicité des éléments au sein du manifeste |
|---|---|
| Content Aggregation                         | 1 et 1 seule fois.  |
| Resource                                    | 1 et 1 seule fois.  |

- 5. Les types de données : <metadata> est un élément container, il n'a donc pas de valeur associée et peut posséder les attributs suivants :
  - Attributs : cet élément ne se dote pas d'attributs ;
  - Éléments : - <schéma>, <schemaversion>, {metadata}.
- 6. L'équivalent de l'élément <metadata> dans Moodle est <INFO>.

6.5.1.1.3 L'élément <organizations>

L'élément <organizations> décrit une ou plusieurs structures ou organisations du contenu.

- 1. Espace de nom XML: [http://www.imsglobal.org/xsd/imscp\\_v1p1](http://www.imsglobal.org/xsd/imscp_v1p1).
- 2. Préfixe de l'espace de nom XML: imscp.
- 3. Représentation de l'élément dans le XML binding : <organizations>.
- 4. Exigences SCORM : les exigences de SCORM sur l'intégration de l'élément <organizations> dans l'élément <manifest> sont décrites dans le tableau 6.7.



Tableau 6.7 : Exigences de l'élément &lt;organizations&gt; dans les modes d'agrégation CAM.

| SCORM Content Packaging Application Profile | Exigences de Multiplicité des éléments au sein du manifeste |
|---|---|
| Content Agregation                          | 1 et 1 seule fois.  |
| Resource                                    | 1 et 1 seule fois.  |

Il doit contenir au moins une fois l'élément <organization>.

5. Les types de données : <organizations> est un élément container. Il n'a donc pas de valeur associée et peut avoir l'attribut default (obligatoire pour une agrégation du contenu). Cet attribut définit la structure par défaut du contenu à utiliser. Sa valeur doit référencer un attribut identifier d'un élément <organization> qui est un descendant direct de l'élément <organizations>. Il est du type xs: IDREF.
6. Les éléments : l'élément <organizations> possède un seul élément qui est <organization>. Ce dernier décrit une organisation hiérarchique particulière. Quand on parle d'organisation, il s'agit en fait d'un concept que SCORM nous laisse définir librement. Il peut s'agir d'une leçon, d'un module, d'un chapitre, d'un cours etc. Il représente en réalité une activité. Il doit apparaître au minimum une fois au sein de l'élément <organizations>.
7. L'équivalent de l'élément <organizations> dans Moodle est <ROLES>.

#### 6.5.1.1.4 L'élément <resources>

L'élément <resources> regroupe toute une collection des références à des ressources. Il n'y a pas de supposition d'ordre hiérarchique entre les éléments <resource> pris individuellement au sein de l'élément <resources>.

1. Espace de nom XML: [http://www.imsglobal.org/xsd/imscp\\_v1p1](http://www.imsglobal.org/xsd/imscp_v1p1).
2. Préfixe de l'espace de nom XML: imscp.
3. Représentation de l'élément dans le XML binding : <resources>.
4. Exigences SCORM : les exigences de SCORM sur l'intégration de l'élément <resources> dans l'élément <manifest> sont décrites dans le tableau 6.8.

Tableau 6.8 : Exigences de l'élément &lt;resources&gt; dans les modes d'agrégation CAM.

| SCORM Content Packaging Application Profile | Exigences de Multiplicité des éléments au sein du manifeste |
|---|---|
| Content Agregation                          | 1 et 1 seule fois.  |
| Resource                                    | 1 et 1 seule fois.  |

5. Les types de données : <resources> est un élément container. Il n'a donc pas de valeur associée et peut avoir l'attribut xml: base (optionnel) qui fournit un chemin d'accès à des fichiers qui sont dans le contenu. C'est une donnée du type xs: anyURI avec une longueur maximum de 2000 caractères.
6. Les éléments : l'élément <resource> est l'unique élément de <resources>, il constitue une référence à toutes les ressources qui constituent le package. Au sens de SCORM, nous avons deux types de ressources les SCO et les Assets.
7. Exigences SCORM pour <resource> : les exigences de SCORM sur l'intégration de l'élément <resource> dans un manifeste sont les suivantes ( tableau 6.9).

Tableau 6.9 : Exigences de l'élément &lt;resource&gt; dans les modes d'agrégation du CAM

| SCORM Content Packaging Application Profile | Exigences de Multiplicité des éléments au sein du manifeste |
|---|---|
| Content Agregation                          | 0 ou plus   |
| Resource                                    | 0 ou plus   |

Un élément feuille <item> est requis pour référencer une ressource (SCO ou actif). Si un <item> fait référence à une ressource, cette ressource devra être identifiée afin d'être fournie et lancée pour un apprenant. Si un <item> référence un <resource>, alors l'élément <resource> doit obéir aux règles suivantes :

- L'attribut type devrait avoir pour valeur webcontent ;
- Le adlcp : scormType devrait avoir comme valeur sco ou asset ;
- L'attribut href devrait être requis ;



8. Les types de données : <resource> est un élément un container. Il n'a donc pas de valeur associée et peut avoir les attributs suivants : identifier (obligatoire), type (obligatoire), href (optionnel), xml: base (optionnel), adlcp: scormType (obligatoire).
9. Les éléments de <resource> sont : <metadata>, <file> et <dependency>.
10. L'équivalent de l'élément <resources> dans Moodle est <COURSE>.

Nous venons de faire un parcours de la mise en œuvre du SCORM XML binding des descendants du premier degré de l'élément <manifest> du fichier manifeste. Nous allons, dans les deux sections suivantes, traiter le contenu Moodle en trouvant pour chaque métadonnée de *moodle.xml* un correspondant dans *imsmanifest.xml*. Une fois toutes les correspondances établies nous passerons à la modélisation de la conception de l'outil d'interopérabilité statique.

## 6.5.2 Concepts et exigences de conformité SCORM

Le consortium ADL définit un certain nombre de règles à respecter pour qu'un contenu soit conforme au modèle SCORM. Pour l'ensemble du package d'un contenu (*Content Packaging*), il définit d'abord quatre niveaux de conformité possible des contenus :

Conformité minimum (*SCORM Version 1.2 Content Packaging XML Conformant – Minimum*). Il s'agit ici de ne respecter que le strict minimum pour qu'un contenu soit reconnu conforme en mettant un accent sur tous les éléments et les attributs « obligatoires » du *SCORM XML binding*.

Conformité minimum avec éléments optionnels (*SCORM Version 1.2 Content Packaging Conformant – Minimum with Optional Elements*). Ici, en plus des éléments obligatoires on ajoute aussi ceux déclarés optionnels dans le *SCORM XML binding*.

Conformité minimum avec extensions (*SCORM Version 1.2 Content Packaging Conformant – Minimum with Extensions*). Ici, la conformité est minimum avec les éléments obligatoires mais admet en même temps des extensions conformes et valides par rapport à leur schéma XML.

Conformité minimum avec éléments optionnels et extensions (*SCORM Version 1.2 Content Packaging XML Conforming – Minimum with Optional Elements and Extensions*). Cette conformité regroupe à la fois la conformité minimum avec éléments optionnels et la conformité minimum avec extensions.

Dans nos travaux, nous avons essayé d'aller le plus loin possible avec la conformité dans l'implémentation. Mais, le niveau de conformité est plus guidé en amont par les objectifs du concepteur du cours au sein de Moodle. Selon ses objectifs un cours peut devenir apte à être transformé du niveau le plus bas de conformité ou au niveau le plus élevé. Le substrat de base du cours est déterminé sous Moodle, lorsqu'un cours est grossier à partir de Moodle, il n'atteindra pas un niveau élevé de conformité à SCORM.

### 6.5.2.1 Profils d'application des contenus

Le consortium définit plusieurs niveaux de conformité d'un contenu au modèle SCORM que l'on nomme profils d'application de la conformité ou *Content Application Profiles* (ADL, 2001). A ce jour, on en dénombre principalement deux : les packages des ressources (*Resources Packages*) et les packages d'agrégation des contenus (*Content Aggregation Packages*).

#### 1° Les packages des ressources.

Ici, le contenu d'apprentissage est assez basique et pas sophistiqué. Il est constitué principalement des fichiers multimédia, sons, images, vidéos, pages Web, des questionnaires ainsi que des données susceptibles d'être délivrées à un client Web. Au sens de SCORM il s'agit d'un package uniquement constitué des assets.

Ce profil définit des mécanismes d'emballage de ressources afin de les rendre interchangeables entre plates-formes sans se soucier de leurs organisations, du contexte d'apprentissage ainsi que du parcours pédagogique. Tout se joue ici au niveau de l'élément <resource> contenu dans l'élément <resources> du manifeste.

#### 2° Les packages d'agrégation des contenus.



Avec ce profil, un package SCORM est capable de mettre à disposition la structure des ressources et les cursus d'apprentissage. Ceci demande en amont d'avoir constitué toutes les ressources sous forme de SCO librement par le développeur ou concepteur. Ce profil donne les moyens d'inventorier et d'empaqueter tous les fichiers physiques nécessaires pour une unité d'apprentissage, et permet aussi d'établir les relations entre les fichiers eux-mêmes et avec les ressources, tout en incluant aussi le référencement à des fichiers externes au package. Ce profil permet en plus de séparer les ressources de contenus de la manière dont elles peuvent être organisées, en les rendant ainsi indépendantes du contexte d'utilisation. L'implémentation se fait au niveau de l'élément <organizations> du fichier manifeste.

#### 6.5.2.2 Règles pour un package

Le consortium ADL définit un certain nombre de règles à respecter pour qu'un contenu soit conforme au modèle SCORM. Pour que l'ensemble du package d'un contenu (*Content Packaging*) soit reconnu conforme, il devra se soumettre aux conditions suivantes :

1. Le manifeste doit être nommé *imsmanifest.xml*. Ce qui induit que la première fonction de l'outil devra changer le nom du fichier *moodle.xml* en *imsmanifest.xml* (par exemple `renameManifest()`).
2. Le fichier *imsmanifest.xml* doit être placé à la racine du package.
3. Tous les schémas auxquels les ressources sont référencées doivent être également à la racine du package.
4. Si le package est placé dans un PIF (*Package Interchange File*), alors le PIF doit être au format zip. Il faudra dézipper le contenu Moodle, puis zipper le contenu après transformation.
5. L'instance *imsmanifest.xml* doit être reconnue bien formée (*well formed*).
6. L'instance *imsmanifest.xml* doit être valide par rapport au schéma XSD définit par l'*IMS Content Packaging XML Schema Definition* (XSD).
7. L'instance *imsmanifest.xml* doit être valide par rapport au schéma des extensions de l'*ADL Content Packaging XML Schema Definition* (XSD).

8. Le contenu devra respecter à la fois les deux profils d'application des packages des ressources et des packages d'agrégation.
9. Le package devrait contenir au moins un SCO ou un asset comme ressource.
10. Toutes les ressources identifiées comme SCO doivent être au moins conformes au *SCORM Run-Time Environment* (RTE).
11. Toutes les métadonnées contenues dans le manifeste devraient être conformes au *SCORM Meta-data Application Profiles* (voir section précédente 6.5.2.1).

### 6.5.2.3 Règles pour un SCO

Comme nous l'avons souligné dans les chapitres précédents, un SCO est un grain d'une unité d'apprentissage. Il fait partie du modèle SCORM, en tant que tel, il est donc décrit avec les métadonnées issues du LOM qui se chargent de décrire les objets pédagogiques. Nous proposons donc un aperçu des règles que la mise en œuvre des métadonnées d'un SCO doit respecter pour que le SCO soit déclaré conforme au modèle SCORM.

1. Une instance des métadonnées XML d'un SCO doit contenir l'élément `<general>`, qui doit apparaître une et une seule fois et contenir les éléments suivants : `<identifier>` (réservé), `<title>` (obligatoire), `<catalogentry>` (obligatoire), `<language>` (optionnel), `<description>` (obligatoire), `<keyword>` (obligatoire), `<coverage>` (optionnel), `<structure>` (optionnel), et `<aggregationlevel>` (optionnel). Tous ces éléments possèdent à leur tour des descendants dont SCORM détermine la priorité ainsi que les occurrences.
2. L'instance doit contenir aussi l'élément `<lifecycle>` qui doit apparaître une et une seule fois et contenir les éléments optionnels suivants : `<version>`, `<status>`, et `<contribute>`. Ceux-ci possèdent à leur tour des descendants dont les règles d'implémentation sont données par SCORM.
3. L'instance doit contenir l'élément `<metametadata>` qui doit apparaître une et une seule fois et contenir les éléments suivants : `<identifier>` (réservé), `<catalogentry>` (optionnel), `<contribute>` (optionnel), `<metadatascheme>`



(obligatoire), et <language> (optionnel). Ces éléments à leur tour sont soumis à des règles définies par SCORM.

4. L'instance doit contenir l'élément <technical> qui doit apparaître une et une seule fois, et dont les éléments sont <format> (obligatoire), <size> (optionnel), <location> (obligatoire), <requirement> (optionnel), <installationremarks> (optionnel), <otherplatformrequirements> (optionnel), et <duration> (optionnel). Ces éléments à leur tour sont soumis à des règles édictées par SCORM.
5. L'instance doit contenir l'élément <educational> qui apparaît 0 ou une fois et qui doit contenir les éléments optionnels suivants: <interactivitytype>, <learningresourcetype>, <interactivitylevel>, <semanticdensity>, <intendedenduserrole>, <context>, <typicalagerange>, <difficulty>, <typicallearningtime>, <description>, et <language>. Des règles sont aussi définies par SCORM pour ceux-ci ainsi que leurs descendants.
6. L'instance doit contenir l'élément <rights> qui doit apparaître une et une seule fois, et dont les éléments sont <cost> (obligatoire), <copyrightandotherrestrictions> (obligatoire), <description> (optionnel). Les règles de chaque élément sont définies par SCORM.
7. L'instance doit contenir l'élément <relation> qui peut apparaître de 0 à plusieurs fois, et dont les éléments optionnels sont <kind> et <resource>. Ces éléments sont à leurs tours soumis à des règles de SCORM.
8. L'instance doit contenir l'élément <annotation> qui peut apparaître 0 ou plusieurs fois et dont les éléments optionnels sont <person>, <date>, et <description>.
9. L'instance XML des métadonnées du SCO doit contenir l'élément <classification>. Il doit être répété au maximum 40 fois selon la recommandation de SCORM. Ses éléments sont <purpose> (obligatoire), <taxonpath> (optionnel), <description> (obligatoire), et <keyword> (obligatoire). Comme pour tous les autres éléments, SCORM définit aussi les règles auxquelles les éléments de <annotation> sont soumis.

#### 6.5.2.4 Règles pour actifs (Asset)

L'implémentation d'un actif lors de l'agrégation d'un contenu SCORM se fait de la même façon que pour un SCO, à la différence que seul le SCO est capable d'échanger avec une plate-forme tandis qu'un actif est considéré juste comme une ressource utile à l'unité d'apprentissage. Le SCO est capable d'envoyer des informations, par exemple sur les tests et leurs pondérations, à une plate-forme et possède une sorte de « *chek box* » qui permet à une plate-forme de détecter des éléments d'un parcours pédagogique à suivre pour l'unité d'apprentissage.

### 6.5.3 Transformation technique

La transformation d'un contenu Moodle en contenu Dokeos ne peut se faire qu'en établissant une liste d'équivalences des métadonnées issues des ces deux mondes. Ce travail n'a pu se faire de manière linéaire, car nous avons constaté des métadonnées sans équivalence de part et d'autre. Il existe une différence de leurs emplacements dans les instances XML (par exemple un élément chez l'un peut plutôt être un attribut chez l'autre) et une différence dans la hiérarchie des mêmes concepts exprimés de part et d'autre. Notre travail a donc cherché à dégager « un consensus conceptuel » entre les métadonnées de Moodle et les métadonnées SCORM soumises par ailleurs à plusieurs règles de conformité comme nous l'avons vu dans les sections précédentes. Une autre difficulté rencontrée est que la taxinomie des métadonnées de Moodle n'est officiellement expliquée nulle part. Nous les avons donc interprétées à partir de leur signification courante dans le monde de l'enseignement en essayant de trouver leur équivalent LOM utilisé dans le *Content Agrégation Model* de SCORM. Au sens des théories d'agrégation de Steinmetz vues à la section 6.3.2.4, il nous a fallu passer par des transformations (par exemple des éléments qui deviennent attributs) et par des abandons pour des concepts de Moodle qui n'ont pas d'équivalents dans SCORM. Au bout de ce travail, nous avons pu dégager les équivalences dont un extrait est consigné dans le tableau 6.10.

Lorsqu'on observe la transformation d'un contenu Moodle en contenu SCORM au niveau de l'instance XML des manifestes, la déclaration de la version de sauvegarde chez



Moodle <BACKUP\_VERSION> correspond à l'attribut version de l'élément racine <manifest> de SCORM. Pour constituer un package des ressources SCORM, l'élément <organizations> devrait apparaître, comme la règle l'exige, mais doit par contre rester vide, vu que le cursus n'est pas intégré dans le package.

Tableau 6.10 : extrait d'une équivalence entre métadonnées SCORM et Moodle.

| Niveau de l'élément | Nom SCORM              | Equivalent Moodle                     | Priorité     | Occurrences |
|---------------------|------------------------|---------------------------------------|--------------|-------------|
| 1.                  | <manifest>             | <MOODLE_BACKUP>                       | Obligatoire  | 1 et 1      |
| 1.1                 | <metadata>             | <INFO>                                | Optionnel    | 1 et 1      |
| 1.1.1               | <title>                | <NAME>                                | Obligatoire  | 1 et 1      |
| 1.1.2               | <catalogentry>         | < ORIGINAL_WWWROOT>                   | Obligatoire  | Au moins 1  |
| 1.1.3               | <description>          | <DETAILS>                             | Obligatoire  | Au moins 1  |
| 1.1.4               | <agregationlevel>      | <ZIP_METHOD>                          | Optionnel    | 0 ou 1      |
| 1.1.5               | <version>              | <BACKUP_VERSION>                      | Obligatoire  |             |
| 1.2                 | <organizations/>       | <ROLES>                               | Obligatoire  | 1 et 1      |
| 1.3                 | <resources>            | <COURSES>                             |              |             |
| 1.3.1               | <resource>             | <SECTIONS>, <MODULES>, <BLOCKS>, etc. | Obligatoires | 1 et plus   |
| 1.3.1...            | <identifier>           | <ID>                                  | reservé      |             |
| 1.3.1...            | <learningresourcetype> | <MODTYPE>                             | optionnel    |             |
| 1.3.1...            | <aggregationlevel>     | <AGGREGATIONCOEF>                     | optionnel    |             |
| 1.3.1...            | <duration>             | <TIMEMODIFIED>                        | optionnel    |             |

Car en amont Moodle ne fournit pas clairement à l'exportation le cursus, mais rend disponibles les utilisateurs et leurs rôles dans le cours. En définitive tout cela dépend du concepteur du cours sous Moodle.

Pour les éléments <SCALES>, <QUESTION\_CATEGORIES>, <MODULES>, nous les considérons comme des SCO puisqu'ils expriment un cursus d'apprentissage en termes d'étapes successives menant à l'achèvement d'un processus d'apprentissage. Les autres éléments, qui ne relèvent que de la structuration des ressources, seront considérés comme des assets (c'est le cas des éléments comme <BLOCKS>, <SECTIONS> etc.). On distingue les SCO et les assets par l'attribut `adlcp:scormtype` de l'élément <resource>, mais les métadonnées pour les décrire restent identiques (voir section 6.5.2.4).

Tableau 6.11 : Illustration d'un exemple de transformation moodle.xml en imsmanifest.xml pour un package des ressources

| Format Moodle.   | Format SCORM correspondant.  |
|--|--|
| <pre> &lt;MOODLE_BACKUP&gt;   &lt;INFO&gt;     &lt;NAME&gt;backup-BDD-25.zip&lt;/NAME&gt;     &lt;BACKUP_VERSION&gt;2003112200   &lt;/BACKUP_VERSION&gt;     &lt;BACKUP_RELEASE&gt;1.2     development&lt;/BACKUP_RELEASE&gt;     &lt;DATE&gt;1072999538&lt;/DATE&gt;   &lt;/INFO&gt;   &lt;COURSE&gt;   &lt;MODULES&gt;   &lt;MOD&gt;     &lt;MOD&gt;       &lt;ID&gt;99&lt;/ID&gt;       &lt;MODTYPE&gt;resource&lt;/MODTYPE&gt;       &lt;NAME&gt;Bases de données     1&lt;/NAME&gt;       &lt;REFERENCE&gt;&lt;/REFERENCE&gt;       &lt;SUMMARY&gt;2&lt;/SUMMARY&gt;       &lt;ALLTEXT&gt; Introduction au     langage SQL&lt;/ALLTEXT&gt;       &lt;TIMEMODIFIED&gt;1070141467&lt;/TIMEMODIFIED&gt;     &lt;/MOD&gt;   &lt;/MODULES&gt; &lt;/COURSE&gt; </pre> | <pre> &lt;manifest&gt;   &lt;metadata&gt;     &lt;schema&gt;MOODLE_BACKUP&lt;/schema&gt;     &lt;schemaversion&gt;2003112200&lt;/schemaversion&gt;   &lt;/metadata&gt;    &lt;organizations/&gt;   &lt;resources&gt;     &lt;resource identifier="99"       adlcp:scormType="sco" href=""&gt;       &lt;metadata&gt;         &lt;title&gt;Bases de         données&lt;/title&gt;         &lt;description&gt;Bases         de données 1&lt;/description&gt;         &lt;educational&gt;          &lt;learningresourcetype&gt;Narrative         text&lt;/learningresourcetype&gt;         &lt;/educational&gt;       &lt;/metadata&gt;       &lt;file         href="BDD_1.htm"/&gt;       &lt;file         href="BDD_1.ppt"/&gt;       &lt;/resource&gt;     &lt;/resources&gt;   &lt;/manifest&gt; </pre> |

Dans le tableau 6.11 nous illustrons un extrait de changement opéré sur les métadonnées, pendant nos expérimentations (avec une flèche qui indique la création d'un



SCO lors de l'agrégation des contenus). On voit bien que l'élément <MOD> dont la valeur de l'élément <ID> est 99 est transformé en un SCO, que l'on détecte par les attributs adlcp:scormType dont la valeur est sco et identifier qui a pour valeur 99.

Le travail sémantique que demande la manipulation des métadonnées nous amène à les réorganiser suivant une autre hiérarchie du format de sortie qui correspond à une autre structure, tout en maintenant le plus possible l'équilibre sémantique pour garantir une même approche d'enseignement de part et d'autre et permettre ainsi d'atteindre les mêmes objectifs dans les deux contextes.

Tableau 6.12 : Illustration de la transformation des métadonnées Moodle pour un actif.

| Format Moodle   | Format SCORM correspondant  |
|---|---|
| <pre> &lt;MOODLE_BACKUP&gt; &lt;COURSE&gt; &lt;MODULES&gt; &lt;MOD&gt; &lt;SUBMISSIONS&gt;   &lt;SUBMISSION&gt;     &lt;ID&gt;129&lt;/ID&gt;     &lt;USERID&gt;0&lt;/USERID&gt;     &lt;TITLE&gt;word Exercice 10&lt;/TITLE&gt;   &lt;TIMECREATED&gt;1065706170&lt;/TIMECREATED&gt;   &lt;RESUBMIT&gt;0&lt;/RESUBMIT&gt;     &lt;MAILED&gt;0&lt;/MAILED&gt;   &lt;ISEXERCISE&gt;1&lt;/ISEXERCISE&gt;     &lt;/SUBMISSION&gt;   &lt;/SUBMISSIONS&gt; &lt;/MOD&gt; &lt;/MODULES&gt; &lt;/COURSE&gt; &lt;/MOODLE_BACKUP&gt; </pre> | <pre> &lt;manifest&gt; &lt;resources&gt;   &lt;resource identifier= "129"     adlcp:scormType="asset" href=" moddata/exercice/129"&gt;     &lt;metadata&gt;       &lt;title&gt;word Exercice 10&lt;/title&gt;       &lt;description&gt; word Exercice 10&lt;/description&gt;       &lt;educational&gt; &lt;learningresourcetype&gt;Exercice&lt;/learningresourcetype&gt;     &lt;/educational&gt;     &lt;/metadata&gt;     &lt;file href="word10.doc"&gt;       &lt;adlcp:location&gt; &lt;/adlcp:location&gt;     &lt;/file&gt;   &lt;/resource&gt; &lt;/resources&gt; &lt;/manifest&gt; </pre> |

On remarque au tableau 6.12 que l'élément <MOD> est bel et bien un asset : il est transformé en son équivalent SCORM qui est l'élément <ressource>, dont l'identifiant est l'élément <ID> côté Moodle, qui se transforme en un attribut identifier côté SCORM en gardant la même valeur. Son attribut adlcp:scormType est là pour certifier que l'élément Moodle <MOD> a été transformé en actif. La valeur booléenne de l'élément Moodle <ISEXERCISE>, qui confirme que la ressource utilisée dans le contenu contient un

exercice, est transformée en l'élément textuel `<learningresourcetype>` dont la valeur LOM est `Exercise`.

Tableau 6.13 : Illustration de la transformation des métadonnées Moodle pour un SCO.

| Format Moodle  | Format SCORM correspondant   |
|--|--|
| <pre> &lt;MOODLE_BACKUP&gt; &lt;COURSE&gt; &lt;MODULES&gt; &lt;MOD&gt; &lt;SUBMISSIONS&gt;   &lt;SUBMISSION&gt;     &lt;ID&gt;128&lt;/ID&gt;     &lt;USERID&gt;0&lt;/USERID&gt;     &lt;TITLE&gt;word Exercise 9&lt;/TITLE&gt; &lt;TIMECREATED&gt;1064314819&lt;/TIMECREATED&gt; &lt;RESUBMIT&gt;0&lt;/RESUBMIT&gt;   &lt;MAILED&gt;0&lt;/MAILED&gt; &lt;ISEXERCISE&gt;1&lt;/ISEXERCISE&gt;   &lt;/SUBMISSION&gt; &lt;/SUBMISSIONS&gt; &lt;/MOD&gt; &lt;MOD&gt; &lt;SUBMISSIONS&gt;   &lt;SUBMISSION&gt;     &lt;ID&gt;129&lt;/ID&gt;     &lt;USERID&gt;0&lt;/USERID&gt;     &lt;TITLE&gt;word Exercise 10&lt;/TITLE&gt; &lt;TIMECREATED&gt;1065706170&lt;/TIMECREATED&gt; &lt;RESUBMIT&gt;0&lt;/RESUBMIT&gt;   &lt;MAILED&gt;0&lt;/MAILED&gt; &lt;ISEXERCISE&gt;1&lt;/ISEXERCISE&gt;   &lt;/SUBMISSION&gt; &lt;/SUBMISSIONS&gt; &lt;/MOD&gt; &lt;/MODULES&gt; &lt;/COURSE&gt; &lt;/MOODLE_BACKUP&gt; </pre> | <pre> &lt;manifest&gt; &lt;resources&gt; &lt;resource identifier= ""   adlcp:scormType="SCO" href=" moddata/exercise/"&gt;   &lt;metadata&gt;     &lt;general&gt;       &lt;title&gt;Exercices word&lt;/title&gt;       &lt;description&gt; Ensemble d'exercices     &lt;/description&gt;     &lt;/general&gt;     &lt;educational&gt;       &lt;learningresourcetype&gt;Exercise&lt;/learningresourcetype&gt;     &lt;/educational&gt;     &lt;/metadata&gt;     &lt;file href="word9.doc"&gt;       &lt;adlcp:location&gt; moddata/exercise/128&lt;/adlcp:location&gt;     &lt;/file&gt;     &lt;file href="word10.doc"&gt;       &lt;adlcp:location&gt; moddata/exercise/129&lt;/adlcp:location&gt;     &lt;/file&gt;   &lt;/resource&gt; &lt;/resources&gt; &lt;/manifest&gt; </pre> |

Dans notre exemple au tableau 6.13 nous constatons que le SCO `Exercices word` est bâti sur deux actifs que sont les fichiers Word `word9.doc` et `word10.doc`. Ceux-ci font désormais partie d'un parcours pédagogique déterminé par le concepteur du cours. Les exemples que nous venons de donner corroborent à plusieurs égards la théorie d'agrégation de Steinmetz et Rensing (voir section 6.3.2.4) qui affirme que le passage des ressources d'un



système à un autre passe par une rémodularisation, une adaptation, une permutation de ces ressources et de leurs descripteurs pour arriver enfin à les réagréger autrement.

## 6.6 Modélisation de l'architecture de l'outil de transformation des données

Dans cette section, nous passons maintenant à la démarche du projet de conception de l'outil capable d'implémenter l'interopérabilité statique. Au niveau fonctionnel, notre outil devrait, pour atteindre les objectifs d'interopérabilité, accomplir les tâches suivantes :

1. Dézipper et extraire tous les éléments d'un contenu Moodle (Décompression et extraction).
2. Extraire tous les éléments nœuds de *moodle.xml* ainsi que leurs valeurs.
3. Traduire tous les nœuds en leur équivalent SCORM.
4. Réorganiser la hiérarchie, modulariser, adapter, permuter, agréger les éléments selon les exigences de SCORM. Rappelons que certains éléments seront transformés en attributs.
5. Zipper le contenu et l'injecter dans la base de données de Dokeos.

Nous utiliserons ici, le langage de modélisation unifié UML pour la modélisation. UML est destiné à la description des systèmes logiciels, nous utiliserons les diagrammes des classes qui décrivent naturellement la structure statique des objets dans un système de même que les relations qui les unissent. Au niveau de la transformation des contenus Moodle en contenus SCORM, le prototype d'interopérabilité sera dotée des composants qui fonctionneront selon les modèles présentés aux figures 6.7 et 6.8.

Sur ce modèle (figure 6.7), on voit bien que les éléments de <MOD>, <SUBMISSIONS>, <SUBMISSION> de Moodle participent à la génération de l'élément <resource> dont l'attribut `adlcp:scormtype` spécifie qu'il s'agit d'un asset, l'attribut `identifier` tire sa valeur de <ID>, tandis que `href` pointe sur <REFERENCE> pour retrouver l'URI de localisation du fichier physique concerné. Notons également que les

cardinalités des relations côté Moodle sont reprises de la documentation officielle de SCORM (ADL, 2002).

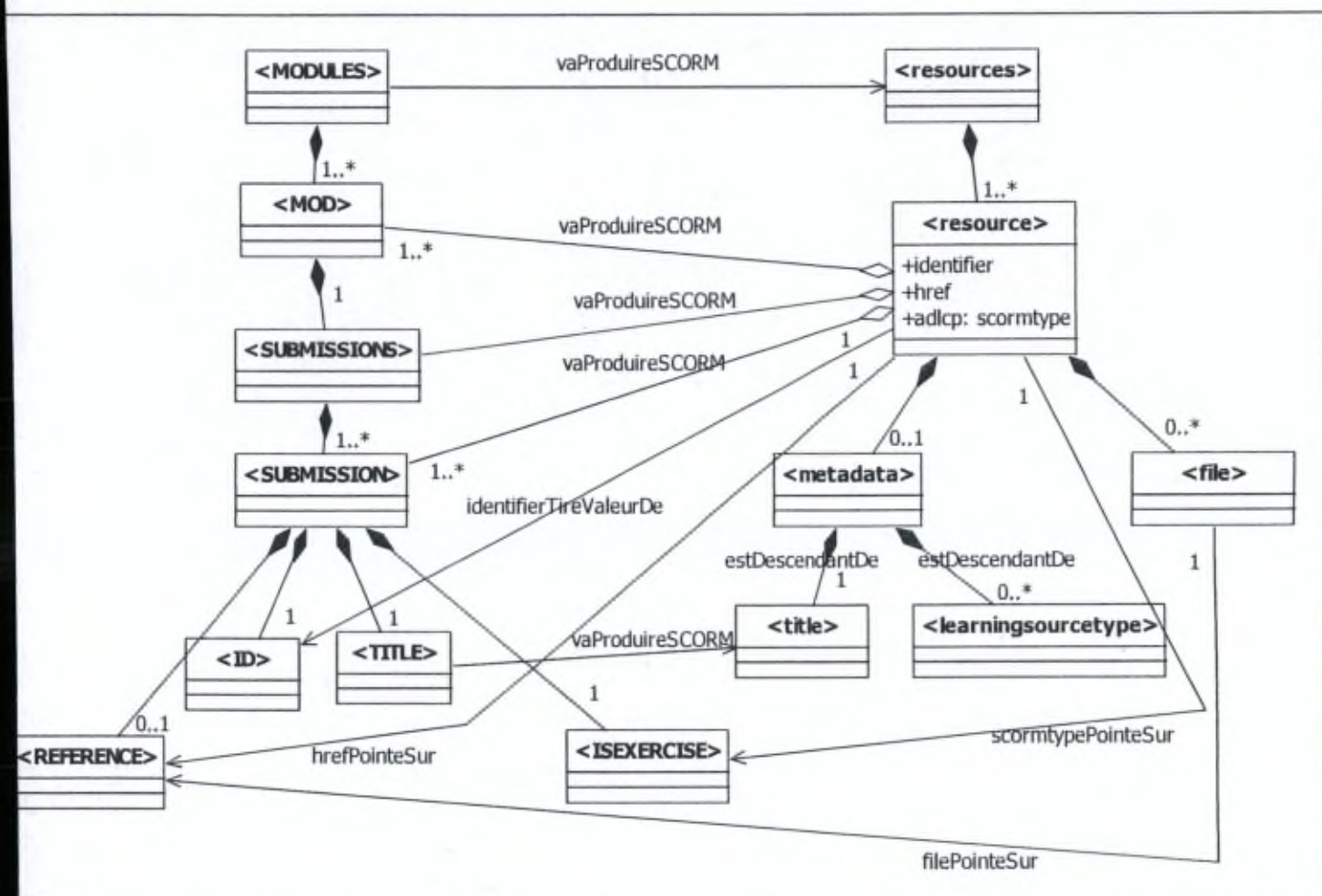


Figure 6.7 : Exemple de Diagramme des classes de la création d'un asset.

Examinons à présent le cas d'un SCO représenté à la figure 6.8. Ici <identifiant> ne tire plus sa valeur de <ID> mais, comme il est déclaré RESERVED dans les exigences de SCORM, sa valeur résultera des échanges entre le SCO et la plate-forme qui se réserve ainsi le droit de lui attribuer une valeur au sein de la plate-forme d'accueil. Ici un SCO est constitué de plusieurs fichiers constituant des ressources du type asset, dont la description est donnée par l'élément <metadata> de <file> avec des métadonnées conformes au profil d'application d'un asset. Ces données sont fournies au départ de Moodle par l'élément



<SUBMISSION>. Dans cet exemple, les éléments <adlcp:location>, <learningsourcetype>, <title> tirent leurs valeurs respectivement des éléments Moodle <REFERENCE>, <ISEXERCISE>, <TITLE>.

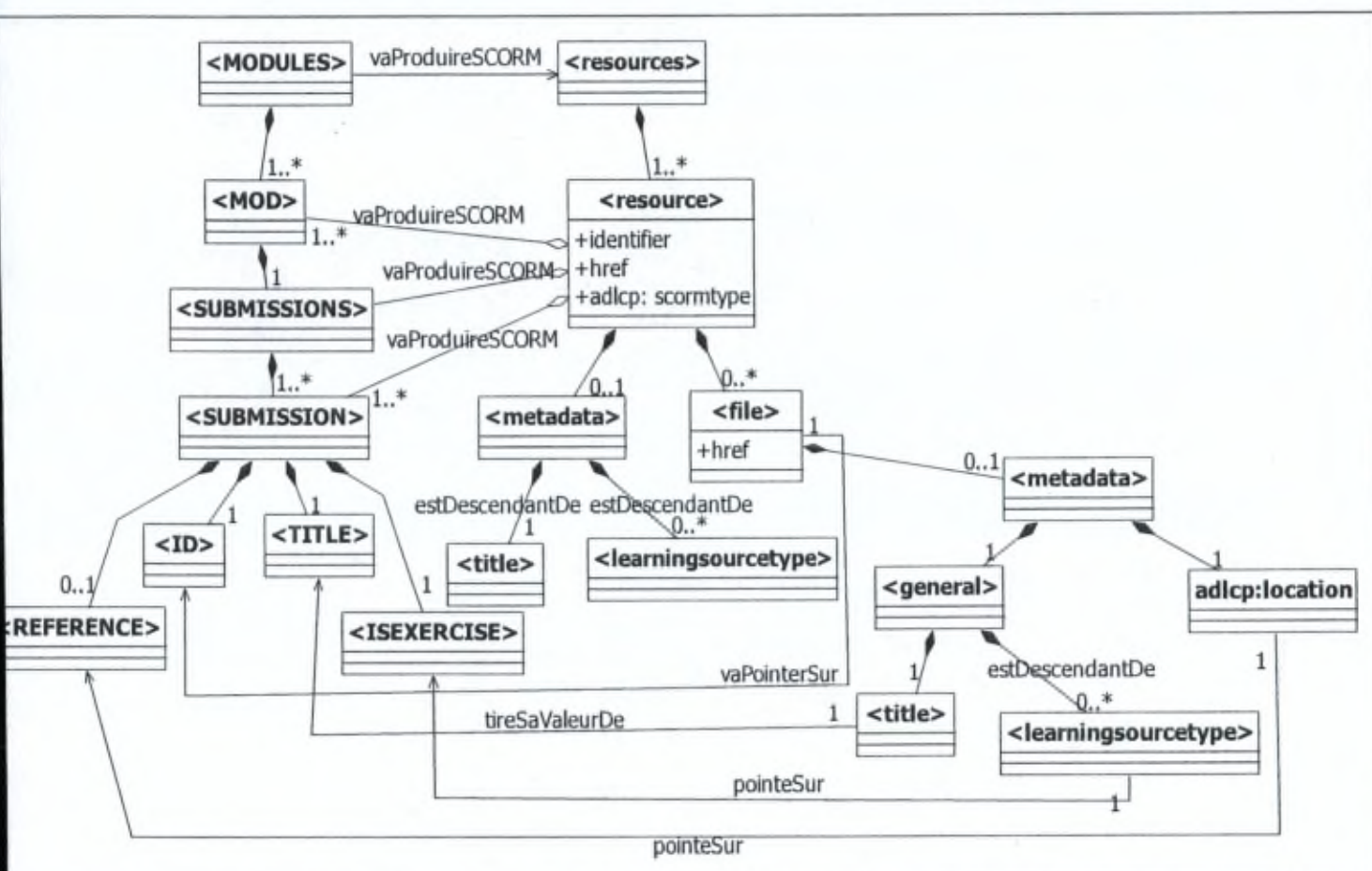


Figure 6.8 : Exemple de Diagramme des classes de la création d'un SCO.

Sur le diagramme de la figure 6.9 nous voyons comment toutes les grandes fonctions qui constitueront le prototype d'interopérabilité sont basées sur DOM (*Document Object Model*)<sup>49</sup>. Comme le diagramme le montre, les manipulations venues de la classe *Traduction\_Moodle\_SCORM* vont alimenter la classe *Moteur\_Transformation\_imsmanifest* qui finalise le processus *XML binding*.

<sup>49</sup> DOM est un standard pour lire et manipuler des documents par le biais d'un programme. C'est une représentation informatique d'un document XML voir section 6.4.2.2

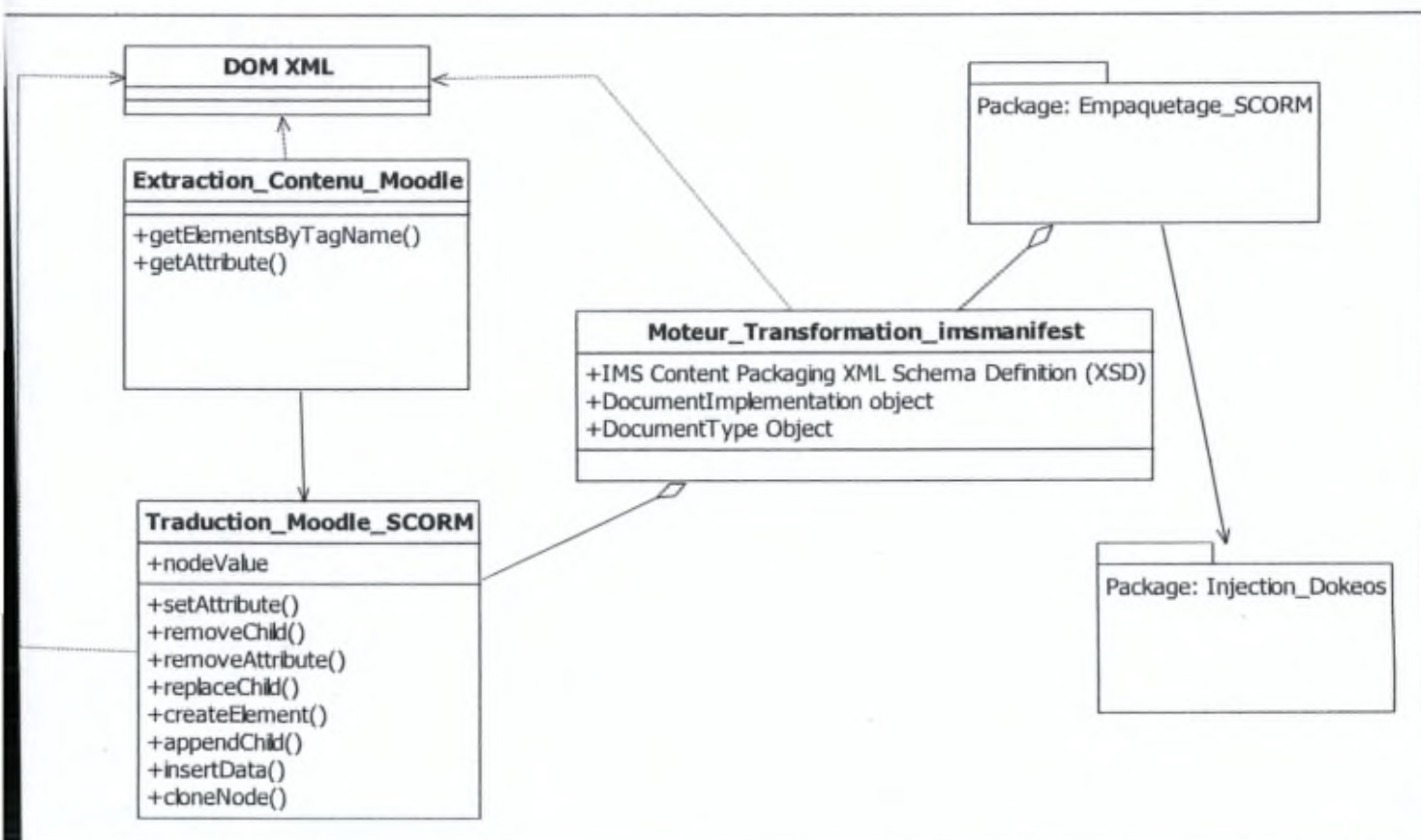


Figure 6.9 : Diagramme des classes des principaux éléments de l'architecture de l'outil d'interopérabilité statique.

## 6.7 Présentation de l'outil et transfert d'un cours

### 6.7.1 Introduction

Le concept d'*implémentation* fait référence en informatique à la mise en œuvre ou la réalisation d'un produit informatique à partir d'une analyse d'un besoin ou d'un problème observé dans un contexte bien précis. Nous avons réalisé dans le cadre de nos travaux un outil pilote (prototype) dont nous allons présenter le fonctionnement global dans le cas d'un transfert des ressources d'un cours. Rappelons que la réalisation de l'outil est le résultat d'un « étalonnage » que nous avons dû mener sur les plates-formes Moodle et Dokeos afin de rendre possible les échanges entre eux.

Pour ce qui est de Dokeos, il exporte un cursus complet, c'est-à-dire le suivi d'un parcours pédagogique, au format SCORM. Ce format SCORM est limité à une conformité que



nous estimons minimale à l'échelle de SCORM. Même des contenus SCORM venus des éditeurs SCORM passent par un « réaménagement » pour être importables dans Dokeos.

Pour sa part, Moodle se veut complet et autonome en ce qui concerne le format zip attendu. Il faut savoir qu'il ne favorise quasiment pas l'usage de packages SCORM pour les cours en ligne dans la plupart de ses sites de production et que la plate-forme Moodle n'en produit pas directement. On peut y importer du SCORM en tant que cours en ligne ou en tant qu'activité d'apprentissage mais on ne peut pas en exporter. On dispose juste de certains formats d'exportation normalisés dans certains contextes tels IMS QTI 2.0 (*Question and Test Interoperability*) ou GIFT (*General Import Format Template*) qui permettent à un utilisateur de se servir d'un éditeur texte pour écrire des questions à choix multiples, vrai-faux, espaces blancs à remplir, etc. Moodle permet tout de même de réaliser un backup complet d'un cours en ligne sous la forme d'un fichier zip contenant les ressources du cours et un fichier XML, du nom de *moodle.xml*, reprenant toute sa configuration comme nous l'avons déjà souligné au chapitre 5. Il accepte l'importation de tout ce qui est SCORM et « refuse » visiblement d'en produire.

Dokeos pour sa part ne pousse pas sa conformité plus loin, et plusieurs acteurs e-learning en font de même. Sur le marché, comme stipulé d'entrée de jeu, c'est la volonté politique des acteurs qui manque sûrement guidés par la concurrence pour arriver à une interopérabilité basée sur le modèle SCORM.

### 6.7.2 Tâches à accomplir par l'outil informatique.

Nous avons développé une application Web qui interagit avec les deux plates-formes conformément au diagramme d'activités donné à la figure 6.10, celui-ci reprend les quatre phases principales d'exécution qui vont de l'adaptation à l'agrégation des contenus d'apprentissage.

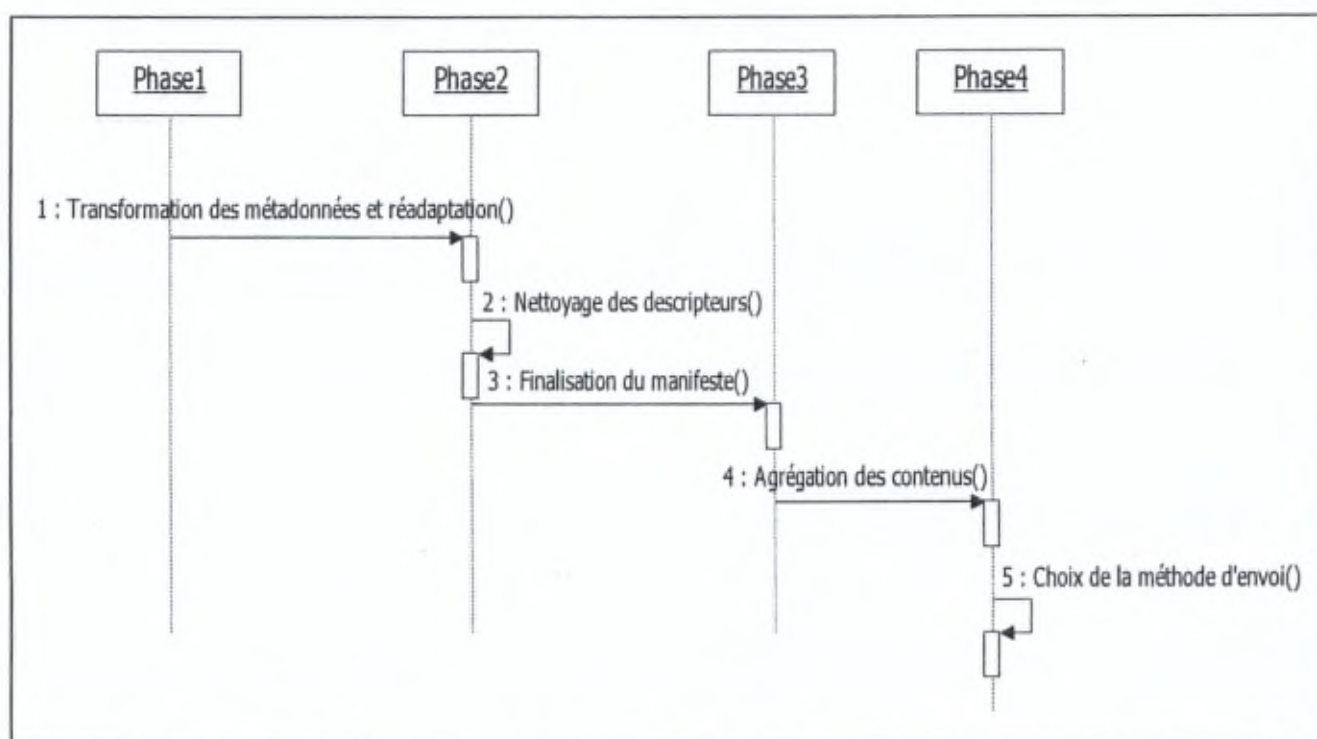


Figure 6.10 : Diagramme de séquence illustrant les phases des opérations principales de Moodle à Dokeos.

La figure 6.11 propose l'algorithme générique de transformation des métadonnées de Moodle à SCORM qui sera utilisé par la suite. Il s'agit de présenter comment le référencement des ressources se fait. L'outil comprend trois grandes parties et travaille en trois phases. L'appel des nœuds par leur nom <MOD> qui contiennent les URI des ressources ou indiquent l'emplacement des fichiers physiques. Ce premier appel se fait au sein du nœud parent <INFO> qui fournit sa liste de nœuds répondant au nom <MOD>. Ensuite, une vérification et un référencement des éléments <NAME> déterminent la nature des éléments (label, forum, chat, ressource, etc.) à extraire comme ressource. L'élément <INCLUDED> reçoit en valeur un booléen *true* ou *false* qui indique si la ressource est présente dans le package ou pas. Dans l'élément <INSTANCE>, on retrouve les descriptifs détaillés de la ressource avec notamment le nœud <ID> dont la valeur nous renvoi à l'identification de la ressource au sein de l'élément <COURSE>. La nature donnée à chaque élément subit un traitement différent pour être transformée en métadonnée SCORM.



```

DEBUT
Var, ValeurNoeud : car ;
Lire listeNoeudsDeNom ;
NombreNoeuds <- 0 ; PositionNoeuds <- 0 ;
Pour PositionNoeuds allant de 0 à NombreNoeuds
Faire
    CreerNoeud('resource') ;
    Le nœud créé est enfant de RendreEnfant('resources') ;
    Lire ListeNoeudEnfants ;
    Si NomNoeud == 'NAME'
        Si ValeurNoeud == 'label' ;
            Lire valeurNoeudEnfant('CONTENT') = ValeurNoeud ;
            Copier ValeurNoeud ;
            CreerNoeud('metadata') ;
            Le nœud créé est enfant de RendreEnfant('resource') ;
            CreerNoeud('description') ;
            Le nœud créé est enfant de RendreEnfant('metadata') ;
            Ecrire ValeurNoeud ;
        FinSi
        Si ValeurNoeud == 'resource' ;
            Lire valeurNoeudEnfant('REFERENCE') = ValeurNoeud ;
            Copier ValeurNoeud ;
            CreerAttribut('href') ;
            href = ValeurNoeud ;
            L'attribut créé est du nœud
RendreAttributDe('resource') ;
            CreerNoeud('file') ;
            Le nœud créé est enfant de RendreEnfant('resource') ;
            Ecrire ValeurNoeud ;
            CreerAttribut('href') ;
            href = ValeurNoeud ;
            L'attribut créé est du nœud
RendreAttributDe('file') ;
        FinSi

    Finsi
    CreerNoeud('educational') ;
    Le nœud créé est enfant de RendreEnfant('metadata') ;
    CreerNoeud('learningsourcetype') ;
    Le nœud créé est enfant de RendreEnfant('educational') ;
    Ecrire ValeurNoeud = 'asset' ;

FinFaire
FIN

```

Figure 6.11 : Algorithme générique de transformation des métadonnées.

Le choix d'une solution Web se justifie aussi par le fait que cela devient avantageux dans la mesure où la quasi-totalité des acteurs de l'e-learning est familiarisée à la navigation sur Internet. De plus la maintenance logicielle est réduite car aucune application n'est à installer sur les postes des utilisateurs. L'utilisation de l'application en déplacement et à distance est un autre avantage à escompter. Sachant qu'une application est un ensemble de codes spécifiques écrits dans un langage informatique dit « *langage de programmation* » qui permet de réaliser une ou plusieurs tâches précises destinées à fonctionner sur un appareil

informatique ou ordinateur dit système informatique. Nous sommes pour notre cas sur une architecture client-serveur dans la mesure où notre prototype sollicitera les services des mêmes logiciels serveur que les plates-formes pour lui fournir les ressources nécessaires à l'accomplissement de ses tâches.

Le langage de programmation choisi pour l'instant est celui des deux plates-formes, c'est à dire PHP. Nous allons voir ici quelques missions assignées à notre outil d'interopérabilité. Notre outil a pour mission d'exécuter les tâches suivantes que nous divisons en deux catégories à savoir les tâches d'agrégation et les tâches de réadaptation adaptées des principes d'agrégation et de réadaptation Steinmetz et Rensing (Steinmetz, et al., 2007) vus au chapitre 5.

1. Création des éléments SCORM (tâche de réadaptation) pour combler les éléments obligatoires non présents de manière explicite ou implicite dans Moodle.

```
$metadataResource = $dom-> createElement("metadata"); // Création de
l'élément metadata et ses descendants.
$general = $dom-> createElement("general");
$identifierResource = $dom-> createElement("identifier");
$titleResource = $dom-> createElement("title");
$descriptionResource = $dom-> createElement("description");
$educationalResource = $dom-> createElement("educational");
$learningResourceType = $dom-> createElement("learning sourcetype");
```

Figure 6.12 : Exemple d'un extrait de code de la tâche 1.

2. Déclaration des relations de filiation (tâche d'agrégation) il s'agit de vérifier de part et d'autre les priorités ainsi que la hiérarchie accordées aux métadonnées de description afin de les réadapter selon les exigences et les règles de conformité érigées par SCORM.

```
$resource-> appendChild($metadataResource); // Fixation des relations entre
les éléments SCORM.
$resource-> appendChild($fileResource);
$metadataResource-> appendChild($general);
$general-> appendChild($identifierResource);
$general-> appendChild($titleResource);
$general-> appendChild($descriptionResource);
$metadataResource-> appendChild($educationalResource);
$educationalResource-> appendChild($learningResourceType);
```

Figure 6.13 : Exemple d'un extrait de code de la tâche 2



3. Détermination des attributs (obligatoires) de l'élément <manifest> selon les règles de conformité de SCORM (tâche de réadaptation).

```
$schema = $dom->createElement("schema");
$valeurSchema = $dom->createTextNode("ADL SCORM");
$schema->appendChild($valeurSchema);
$schemaVersion = $dom->createElement("schemaVersion");
$valeurSchemaVersion = $dom->createTextNode("1.2");
$schemaVersion->appendChild($valeurSchemaVersion);
$metadata->appendChild($schema);
$metadata->appendChild($schemaVersion);
```

Figure 6.14 : Exemple d'un extrait de code de la tâche 3

4. Détermination des attributs de l'élément <resource> (tâche de réadaptation et agrégation). Certains de ses attributs tirent à la fois leurs conditions d'existence et leurs valeurs de certains éléments nœuds du fichier *moodle.xml* qu'il va falloir transformer en attribut dans le fichier *imsmanifest.xml*.

```
//Va suivre la section d'affectation des attributs à l'élément resource
$resource-> setAttribute("identifiant", "");
$resource-> setAttribute("type", "webcontent");
$resource-> setAttribute("href", "");
$resource-> setAttribute("adlcp:scormtype", "asset");
$resource-> setAttribute("xml:base", "");
$fileResource-> setAttribute("href", "");
```

Figure 6.15 : Exemple d'un extrait de code de la tâche 4

5. Détermination des attributs de l'élément <file> (tâche de réadaptation et agrégation). Comme pour l'élément <resource>, certains de ses attributs tirent à la fois leurs conditions d'existence et leurs valeurs de certains éléments nœuds du fichier *moodle.xml* qu'il va falloir transformer en attribut dans le fichier *imsmanifest.xml*.

```
$reference = $dom->getElementsByTagName('REFERENCE');
$nombreReference = count($reference);
echo $nombreReference. "<br />";
foreach($reference as $href)
    //$fileResource = $resources-> createElement("file");
echo $href->nodeValue. "<br />";
$fileResource-> setAttribute("href", $href->nodeValue);
```

Figure 6.16 : Exemple d'un extrait de code de la tâche 4

6. Déplacement des ressources vers un nouvel emplacement (tâche d'agrégation) :

Il s'agit d'un emplacement tampon pour la préparation des données avant leur importation.

7. Importation du contenu vers Dokeos (tâche d'agrégation).

### 6.7.3 Traitement des métadonnées : agrégation et réadaptation des ressources

Le traitement des métadonnées apporte de nouveaux descripteurs issus du LOM qui obéissent à des règles bien strictes de conformité établies par le modèle SCORM. Il est suivi ensuite de l'agrégation des contenus conformément au CAM (*Content Agregation Model*). Nous avons essayé pour amorcer cette tâche complexe de cadrer notre approche itérative en nous focalisant sur le respect de deux étapes principales. La première se situe au niveau de la conformité : on a commencé par une conformité minimum des métadonnées du LOM. La deuxième se situe au niveau de l'agrégation avec le mode « agrégation des ressources » du CAM.

Les métadonnées ont subi les transformations suivantes par l'outil :

1. <MOODLE\_BACKUP> éléments racine de *moodle.xml* de Moodle est remplacé par <manifest> élément racine de *imsmanifest.xml* de SCORM.
2. <INFO> quant à lui constitue l'équivalent de l'élément <metadata>.

La réadaptation l'élément <INFO> se fait à travers son nœud enfant <DETAILS> qui fournit tous les détails nécessaires sur le cours en tant qu'un tout, module par module à travers l'élément <MOD>. C'est ici que nous vérifions la présence d'un outil, d'une ressource ainsi que le profil de ses utilisateurs. Nous spécifions entre autres s'il s'agit d'un forum, d'un chat, d'un test, d'un quiz, d'un workshop, d'un texte, d'un fichier, d'un glossaire, d'un journal, d'une activité, d'un cours, etc.

3. <COURSE> c'est l'élément descripteur du cours en détails nous l'avons remplacé dans son rôle par <resources> de SCORM. Pour une agrégation des ressources de manière directe, c'est l'élément <MOD> nœud enfant de <MODULES> fils de



<COURSE> qui est remplacé par <resource> dans SCORM. Ses attributs sont identifier qui s'autodétermine par le *Run-time Environment* de la plate-forme d'accueil. Les attributs type, href, adlcp :scormtype, trouvent aussi leurs valeurs par l'extraction des valeurs des éléments contenus dans le nœud <MOD> comme le montre le tableau 6.14

Tableau 6.14 : Exemple des correspondances adoptées dans nos travaux des éléments Moodle vs SCORM.

| XPath : COURSE/MODULES/MOD | XPath : resources/resource/metadata/general     |
|----------------------------|---|
| ID                         | identifier                                      |
| NAME                       | title   |
| SUMMARY                    | description                                     |
| XPath : COURSE/MODULES/MOD | XPath : resources/resource/metadata/educational |
| CONTENT                    | description                                     |
| SCALES/SCALE/DESCRIPTION   | description                                     |
| MODTYPE                    | learningsourcetype                              |

Il faut noter que pour une agrégation de ressources, nous avons adopté deux types d'éléments <MOD> différents via la valeur des nœuds enfants <MODTYPE> : soit label, soit resource. La valeur label décrit du texte à afficher soit de manière brute, soit entre les balises HTML (webcontent). Tandis que la valeur resource fait référence à un fichier physique par son URI. L'élément <TYPE> de <MOD> détermine quant à lui l'existence d'un fichier.

Les éléments <SECTIONS> sont similaires à une présentation successive des chapitres. Ils sont les descripteurs d'un chapitre entier d'un cours et indiquent quels sont les outils (forums, chat, etc.) que l'on a déjà utilisé ou que l'on peut utiliser, pour suivre le chapitre.

Les éléments <USERFILES> et <COURSEFILES> signalent par leur valeur booléenne (true ou false), la présence des fichiers accompagnant le cours et indiquent parfois leur emplacement. Pour aller en profondeur dans le transfert des ressources, nous

avons aussi l'élément <SCALES>. À travers son nœud enfant <SCALE>, il produit une ressource SCORM du type asset qui ajoute des informations supplémentaires sur le cursus du cours sans nécessairement avoir un caractère contraignant. Ses nœuds enfants <NAME>, <SCALETEXT>, <DESCRIPTION> fournissent respectivement le nom, le résumé, ainsi que la description de la ressource.

4. Dès le départ, c'est dans l'élément <DETAILS> de <INFO> que se fait la déclaration et l'identification des différents éléments <MOD> qui constituent le cours. Il les groupe selon leurs catégories définies par la valeur de son élément <NAME>. Il peut s'agir de forum, label, resource, etc. Les autres éléments auxquels nous devons faire attention sont : <INCLUDED> et <USERINFO> qui reçoivent en valeur un booléen qui indique si, au sein du cours, on retrouve le module annoncé ou pas. <INSTANCES> identifie, nomme et atteste la présence du module avec respectivement les éléments <ID>, <NAME>, <INCLUDED>, <USERINFO>, les deux derniers étant des booléens.

## 6.7.4 Illustration transformation et transfert du cours

Pour visualiser la transformation et le transfert nous proposons ici quelques captures d'écrans qui vont illustrer notre démarche pour le cours «*Beginners IT*», cours d'initiation à l'informatique.

### 6.7.4.1 Interface de départ dans Moodle

Dans Moodle, le cours se présente sous forme de six sections dont quatre modules de cours dans la colonne centrale de l'interface (figure 6.17). La première section du cours présente brièvement le cours et propose un forum pour recevoir de l'aide des autres membres du cours. Les quatre sections suivantes sont constituées des modules des cours : le premier module décrit les objectifs de la section puis propose un lien vers des différentes ressources groupées selon les thématiques diverses (hardware, software, réseaux, sécurité, droit, etc.) ; le deuxième module traite de l'usage d'un ordinateur et est accompagné de ressources qui traitent des systèmes d'exploitation, de la gestion des fichiers, des périphériques, etc. ; le troisième module traite d'Internet et de la communication et est



accompagné des ressources qui traitent d'Internet, du Web, des navigateurs, de la messagerie électronique, etc.



Figure 6.17 : Interface du cours dans Moodle

#### 6.7.4.2 Interfaces phases de transformations

Nous avons choisi de proposer une interface Web simple pour l'outil qui propose notamment à l'utilisateur les options pour démarrer une transformation, voir les contenus disponibles à transformer, et accéder à des ressources qui fournissent des éléments sur l'usage de l'outil et son projet de développement (voir figures 6.18 et 6.19).



Figure 6.18 : Page d'accueil de l'application.



Figure 6.19 : Interface de la Phase 1 de la transformation.

Une fois que l'on clique sur l'onglet «*démarrer une transformation de contenu*», on accède à la première phase où il est demandé à l'utilisateur via l'onglet «*Choisir et Charger*» de choisir et charger le contenu à transformer (fichier zip exporté depuis Moodle). Une fois



le contenu choisi, on peut alors démarrer les transformations et l'utilisateur est guidé d'interface en interface de la phase 1 à la phase 4 tel qu'illustré sur la figure 6.10 du diagramme des séquences de l'outil, jusqu'au transfert complet des données.

### 6.7.4.3 Interface à l'arrivée sur Dokeos

Sur Dokeos, l'interface du cours se présente tout à fait différemment de Moodle. Lorsqu'on est sur la page d'accueil, parmi les nombreux outils et options que proposent Dokeos, on choisit l'option «*Modules*». Une fois que l'on clique dessus, il nous renvoie à la page des modules du cours ou chaque module est représenté par un rectangle dans lequel se trouve un bouton en forme de fléchette pour «*jouer*» le module. Le module déjà joué voit sa barre colorée pour indiquer que l'apprenant a déjà consulté le module.



Figure 6.20 : Interface du cours dans Dokeos.

### 6.7.5 Bilan de la transformation.

Nous nous sommes arrêtés à la construction des packages de ressources au sens du modèle SCORM. Ceci sous-entend que les contenus sont pris en tant que tels sans être accompagnés des noms d'utilisateurs, de leurs droits et rôles par rapport aux contenus, des

messages et sans les données des parcours personnalisés des apprenants. Il manque aussi les interactions via les différents outils de communication synchrones et asynchrones, la synchronisation dynamique s'en chargera. Mais la méthodologie utilisée est généralisable.

Le transfert des ressources a donc été possible, et on a pu rendre un cours interopérable au niveau de ses ressources et non du cursus entre les plates-formes Dokeos et Moodle. Nous avons retenu deux cours pilotes. D'une part, nous avons choisi le cours « *Beginners IT* » pour les notions de base en informatique<sup>50</sup>. Il est en effet utile à l'Université de Lubumbashi vu qu'un cursus sur les notions de base en informatique est encore nécessaire localement pour toutes les facultés où l'apprentissage de l'informatique s'impose. Le deuxième cours est la « *Description et Construction des Machines* » obtenu de l'Université de Mons (UMONS), une des universités belges partenaires de l'Université de Lubumbashi. Ce cours est issu du service de génie mécanique de la Faculté polytechnique de l'UMONS et pourra servir au département d'électromécanique de l'UNILU. Il a notamment pour objectif de fournir une vue d'ensemble sur les critères limitant l'utilisation des organes de machine ainsi que de fournir une vue d'ensemble des principaux éléments de machine assurant les fonctions de transmission de puissance, de guidage, d'assemblage et d'étanchéité.

Néanmoins cette réussite de transfert s'est faite avec certaines difficultés, dont les principales ont été :

1. La réadaptation des ressources Moodle au format SCORM a posé la difficulté de retrouver les contenus et fichiers strictement dans l'ordre où ils ont été conçus. Un léger réaménagement manuel sur Dokeos est à ce stade encore nécessaire pour éviter d'inverser parfois l'ordre de l'appel des fichiers via leur URI (les modules d'un cours apparaissent parfois en désordre).
2. Nous n'avons pas tenu compte à ce stade de certaines données d'enseignement véhiculées via les forums de Moodle tant l'implémentation

---

<sup>50</sup> Ce cours est proposé sur le site officiel de Moodle comme un des back up utiles à des études portant sur les caractéristiques des contenus Moodle et de la manière de les agréger



des systèmes Moodle et Dokeos est très différente. Ceci pourra faire l'objet d'une étude ultérieure.

3. Le contenu devant être sous format zip, on a été limité par la puissance du langage qui n'implémente pas une classe suffisamment puissante pour la compression des données. Pour zipper les données, nous avons dû le faire manuellement à l'aide des logiciels dédiés. Mais le développement futur de l'outil dans un langage de programmation plus puissant pourrait permettre de zipper automatiquement.
4. Le mode d'envoi des fichiers zip par Internet (par exemple, en FTP) dépend en partie de la configuration de chacun des réseaux locaux sur lesquels sont installées les deux plates-formes et de la politique d'administration des réseaux. Il faut donc des accords entre des universités partenaires.
5. Nous avons trouvé tout à fait absurde le fait que Moodle soit capable d'agréer SCORM que dans un seul sens, c'est-à-dire à l'import, et soit incapable d'agréer ses propres contenus et de les exporter au format SCORM. Nous pensons qu'une bijectivité de l'interface d'interopérabilité entre Moodle et SCORM apporterait une plus-value à l'interopérabilité en e-learning, compte tenu des places de choix qu'occupent ces deux acteurs majeurs dans le domaine.
6. Pour l'instant le prototype se limite au transfert des contenus. Une étude plus approfondie est nécessaire pour assurer un transfert encore plus complet des cursus qui pourrait faire l'objet d'un financement ultérieur.

## CHAPITRE 7 : IMPLEMENTATION DE L'INTEROPERABILITE DYNAMIQUE.

### 7.1 Introduction

Pour atteindre les objectifs de nos travaux tels que retracés dans la chapitre 2, notamment la mise à jour et l'incrémentation du niveau des enseignements donnés à l'Université de Lubumbashi, les outils de communication occupent une place prépondérante et sont appelés à être intégrés dans un processus optimisé d'interopérabilité entre plates-formes. Cela répond aussi à une des exigences obtenues lors de nos expérimentations où nous avons découvert que dans le cadre de l'e-learning à l'UNILU, il fallait prévoir, pour chaque groupe, un forum et un chat thématiques. En effet, les modes de communication synchrone et asynchrone se sont révélés être le gage d'un apprentissage collaboratif réussi. Ayant travaillé sur une méthodologie de transfert des contenus entre Moodle et Dokeos au chapitre précédent, nous voulons maintenant proposer une méthodologie pour synchroniser les outils de communications synchrones et asynchrones de Moodle et Dokeos. Pour ce faire, nous nous focalisons dans un premier temps sur l'outil de communication asynchrone à savoir le forum. Ce dernier a l'avantage de favoriser aussi une communication quasi synchrone avec l'enseignant, ce qui améliore le suivi des étudiants. Le forum, au-delà d'être un outil favorisant le tutorat et le suivi des apprenants, permet la construction d'une véritable communauté d'apprentissage. Il est aussi un outil à multiples usages avantageux pour un enseignement et peut servir comme :

1. Outil d'information : un enseignant à distance peut en effet utiliser un forum pour diffuser une information portant sur un rendez-vous ou sur toute autre activité liée à la tâche d'apprentissage.
2. Outil de communication : un forum synchronisé permet à l'enseignant de communiquer et d'être en interaction tant synchrone qu'asynchrone avec ses apprenants.



3. Outil de collaboration à distance : le forum peut devenir pour les différents groupes d'apprenants un espace de travail partagé et favoriser le travail collaboratif.
4. Outil de dépôt ou de partage des documents : le forum comme outil de partage et d'échange de documents entre apprenants et aussi entre l'enseignant et ses apprenants.
5. Outil favorisant la construction d'une communauté d'apprentissage : le forum a aussi pour vocation pédagogique de construire une communauté d'apprentissage car il permet aux apprenants d'interagir avec les autres plutôt que de rester seul en face d'une matière. Il améliore ainsi la qualité de l'apprentissage, l'efficacité de la formation et les performances des apprenants.
6. Outil favorisant la métacognition : les experts en pédagogie sont tous d'accord pour reconnaître que les outils de communication asynchrone offrent la possibilité à l'apprenant de poser sur son propre parcours de formation un regard réflexif, et d'améliorer l'efficacité de son apprentissage. La métacognition ainsi développée favorise donc l'autonomie et la responsabilisation des apprenants, leur motivation et leurs performances.

Nous allons, au cours de ce chapitre proposer une méthodologie de conception, constituant un avant projet d'un prototype informatique d'interopérabilité dynamique. Nous donnerons parmi les technologies celle qui se prête le mieux à notre cas, en l'occurrence les services Web. Nous posons ici les bases de la construction future d'un tel outil à travers la technologie choisie. Rappelons encore que le cas traité ici concerne l'outil de communication asynchrone, le forum.

## **7.2 Présentation des outils forum de Moodle et Dokeos.**

### **7.2.1 Présentation du forum Dokeos**

Le forum est ici un outil pour des échanges principalement en mode asynchrone. Les échanges sont organisés de façon hiérarchique selon l'arborescence : dossier de forum/Forum/Fil dediscussion/Sujet dedépart/Message(s)/Réponse(s)/Message(s)/Réponse(s)

). Pour simplifier la lecture des messages plusieurs vues peuvent être affichées selon la complexité des échanges sur les forums et les réponses aux articles postés. Ainsi nous distinguons :

1. La vue linéaire : qui présente simplement les articles sous forme chronologique.
2. La vue thématique : qui simplifie grandement l'affichage de l'interface en ne proposant à la lecture qu'un seul article à la fois, selon le thème développé.
3. La vue hiérarchique : qui reprend le principe de la vue thématique, en affichant l'intégralité des articles d'un sujet ou d'un thème, avec une indentation.

Selon les droits que l'on a en tant qu'administrateur, enseignant ou apprenant, on peut notamment effectuer les actions suivantes :

- ajouter un dossier de forum ;
- ajouter un forum ;
- administrer les dossiers des forums ;
- administrer les forums ;
- lancer un nouveau sujet ;
- administrer les sujets ;
- administrer les messages.

On note aussi une évaluation possible des apprenants par leur enseignant lorsque ce dernier choisi de noter un fil de discussion, qui possède alors un score maximum à atteindre. Un titre de colonne et un coefficient apparaissent dans le rapport de compétences. Il devient ainsi possible d'évaluer un apprenant depuis la liste des fils de discussion d'un forum. Au niveau technique le forum est un module constitué de plus d'une vingtaine des scripts PHP, interagissant pour le bon fonctionnement de l'outil.

## 7.2.2 Présentation du forum Moodle

Sous Moodle, le forum est un des modules de l'ensemble de la plate-forme. Dans un cours donné, on peut en avoir plusieurs et ces forums sont organisés selon les en-têtes suivants :



1. Le nom du forum (Forum).
2. L'objectif du forum et le type de sujet traité (Description).
3. Le nombre de discussions démarrées (Discussions).
4. Le nombre de messages non encore lus (Messages non lus).
5. Le suivi des messages qui fournit l'information selon laquelle on a choisi ou non de suivre les messages non lus. En cas de choix négatif, le signe « - » est produit à la place du nombre de messages non lus.
6. L'abonné fournit l'information qui détermine si l'on a choisi de recevoir les messages par courriel ou pas.
7. Le bouton «RSS» *Really Simple Syndication* : qui contient les mises à jour des derniers messages publiés.

Il existe deux catégories principales de forums :

1. Le forum des nouvelles qui est placé dans la section 0 de chaque cours. Ce forum est particulier :
  - il est créé automatiquement avec chaque cours et est unique ;
  - seul l'enseignant peut y écrire ;
  - il est associé à un bloc latéral «dernières nouvelles» qui, comme son nom l'indique, diffuse les derniers messages de ce forum ;
  - quand il est supprimé, il n'est pas possible de le recréer par la méthode traditionnelle (création d'une nouvelle activité forum).
2. Les forums d'apprentissage qui sont placés dans les sections spécifiques du cours auxquelles ils se rapportent. Ils sont organisés et numérotés de la même façon que les sections du cours dans lesquelles ils apparaissent.

Le forum est un outil de communication asynchrone très prisé dans l'usage de la plate-forme. Il permet notamment de classer les messages dans une variété de formats et ils peuvent contenir un fichier attaché. En s'inscrivant à un forum, les participants reçoivent par courriel une copie de chaque nouveau message émis dans ce forum. Les forums peuvent être organisés de différentes façons et peuvent inclure une évaluation par les pairs pour chaque message. Au niveau technique le forum est un module implémenté en plus d'une trentaine des scripts PHP, JavaScript, XML et HTML.

## 7.3 Choix de la technique à appliquer pour une interopérabilité dynamique : les services Web

### 7.3.1 Contexte technique

Les modules forum des deux plates-formes stockent tous les messages dans des bases de données dédiées. Pour des raisons de simplicité, nous optons pour un échange des messages bruts entre les plates-formes et cela indépendamment de SCORM et des plates-formes elles-mêmes. Notre application basée sur le service Web échangera avec Moodle et Dokeos pour extirper de part et d'autre des messages à échanger comme on le voit à la figure 7.1. Notons aussi qu'actuellement Moodle et Dokeos n'ont pas de service Web mais que cela pourrait être implémenté dans les versions à venir.

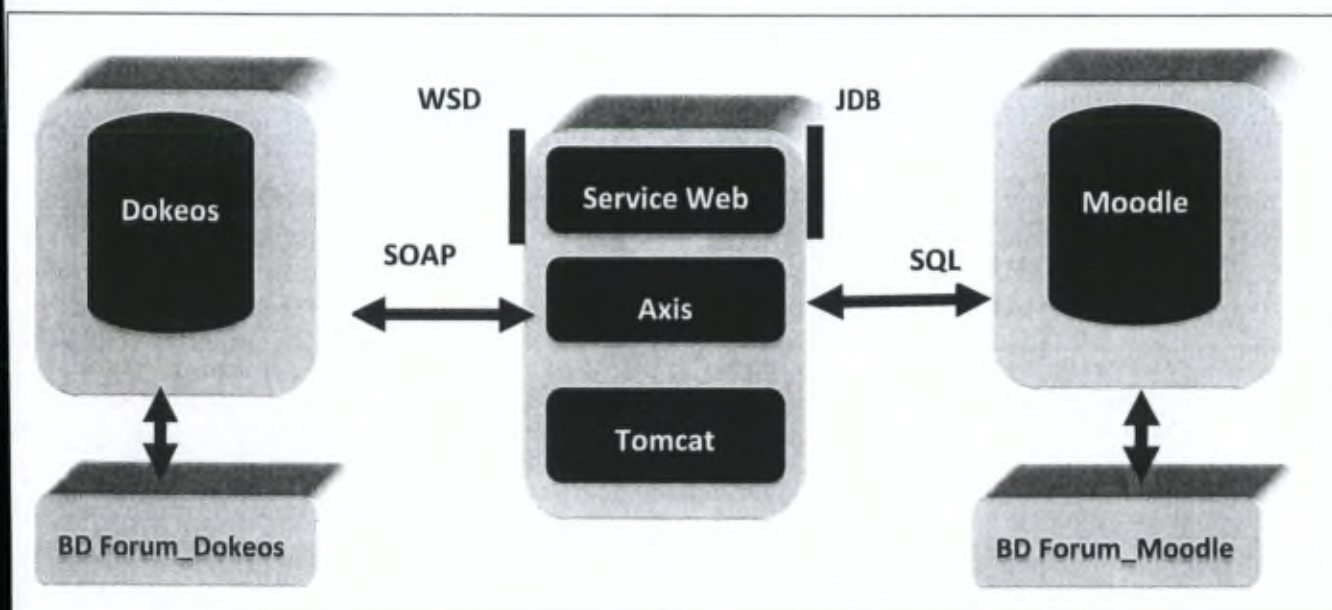


Figure 7.1 : Architecture du service Web des forums (W3C, 2004).

Pour rendre interopérables les forums de Moodle et Dokeos plusieurs possibilités s'offrent à nous. Parmi les nombreuses techniques permettant d'établir des échanges entre les applications informatiques, citons :

- l'échange des données entre applications ;
- la technique d'interopérabilité point à point ;
- les EAI (*Enterprise Application Integration*) ;



- les techniques d'interopérabilité au niveau de la couche des données ;
- le patron de conception.

Notre choix portera dans la cadre de ce travail sur les techniques d'interopérabilité point à point notamment les appels des procédures à distances dont les services Web sont un cas particulier.

### 7.3.2 Justification du choix des services Web

Nous optons pour le service Web pour rendre cette interopérabilité le plus simple possible, en soustrayant à la fois aux utilisateurs finaux et aux développeurs la complexité et la diversité des environnements, mission que les services Web ont la réputation de remplir parfaitement. Notre choix est renforcé par le fait que les services Web se révèlent une technologie émergente permettant à des applications de dialoguer à distance via Internet et ceci, indépendamment des plates-formes sur lesquelles elles reposent. Les services Web visent à réduire les problèmes d'interopérabilité en permettant à des applications de communiquer directement entre elles et d'échanger des données indépendamment de leur localisation, des systèmes d'exploitation ou langages utilisés. Ils sont dotés d'un caractère indépendant, polymorphe et multidimensionnel qui favorise leur adoption par la plupart de grands éditeurs de logiciels.

La problématique de notre travail rencontre parfaitement les trois principales caractéristiques des services Web utiles dans le domaine de l'interopérabilité :

- *des protocoles de transport standard* : les protocoles HTTP et HTTPS sont en effet les implémentations les plus répandues, mais les services Web ont la flexibilité et la capacité d'utiliser n'importe quel protocole de transport ;
- *la définition de type* : les services Web peuvent fournir des données fortement typées. Ils assurent un fonctionnement totalement indépendant de leurs plates-formes ou des langages de programmation de l'environnement qui les supporte, à partir du moment où ils utilisent et comprennent le même type ;

- *de multiples niveaux de support* : le client est épargné du souci de connaître la plate-forme sur laquelle un service particulier s'exécute car les services Web peuvent être implémentés dans n'importe quel langage, n'importe quelle plate-forme, et utiliser n'importe quel outil commercial.

Nous adopterons donc les services Web comme moyen de faire communiquer et d'échanger les messages des forums entre Dokeos et Moodle. Dans les sections suivantes, nous allons brièvement aborder les notions de base des services Web. Ensuite, nous proposerons une méthode pour synchroniser les deux forums afin de permettre un suivi à distance effectif des étudiants de l'Université de Lubumbashi par un enseignant belge utilisant une autre plate-forme (Moodle dans notre implémentation).

### 7.3.3 Principes et fonctionnement des services Web

#### 7.3.3.1 Définition du service Web

De manière simplifiée, les services Web se définissent comme des composants logiciels encapsulant des fonctionnalités métier<sup>51</sup>. Ils sont accessibles via des protocoles standards, basés sur XML, et des protocoles de transport (HTTP, SMTP, etc.). On associe généralement les services Web à deux spécifications du W3C : SOAP (*Simple Object Access Protocol*), protocole d'échange de messages XML permettant d'invoquer des opérations à distance, et WSDL (*Web Service Description Language*), langage de description des services Web. Un annuaire, notamment UDDI (*Universal Description, Discovery and Integration*), est en appui pour permettre une publication et une exploration dynamique des services Web offerts par les entreprises. On voit donc sur la figure 7.2 qu'on est dans un schéma de type RPC (*Remote Procedure Call*). La nouveauté consiste dans le mode d'appel du service : un flux ASCII encadré dans des balises XML et transporté principalement via HTTP. De manière succincte, nous pouvons dire qu'un service Web est principalement composé de trois parties : une interface assumée par WSDL, un nombre non-limité d'implémentations et un protocole

<sup>51</sup> En architecture 3-tiers, elles correspondent à la partie fonctionnelle de l'application, celle qui implémente la logique, et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation.



d'échange des messages assuré par SOAP. La figure 7.2 schématise les principes de fonctionnement d'un service Web.

1. Un client (Moodle ou Dokeos) recherche un serveur qui publie un service Web.
2. L'annuaire UDDI<sup>52</sup> lui retourne la désignation d'un serveur hébergeant le service, lorsque ce dernier a préalablement été publié par le serveur dans l'annuaire.
3. Le client demande au serveur quel format d'appel il implémente.
4. Le serveur lui retourne sa description par son contrat WSDL<sup>53</sup>.
5. Le client comprend le contrat et envoi une requête sous forme XML.
6. Le serveur lui retourne le résultat.

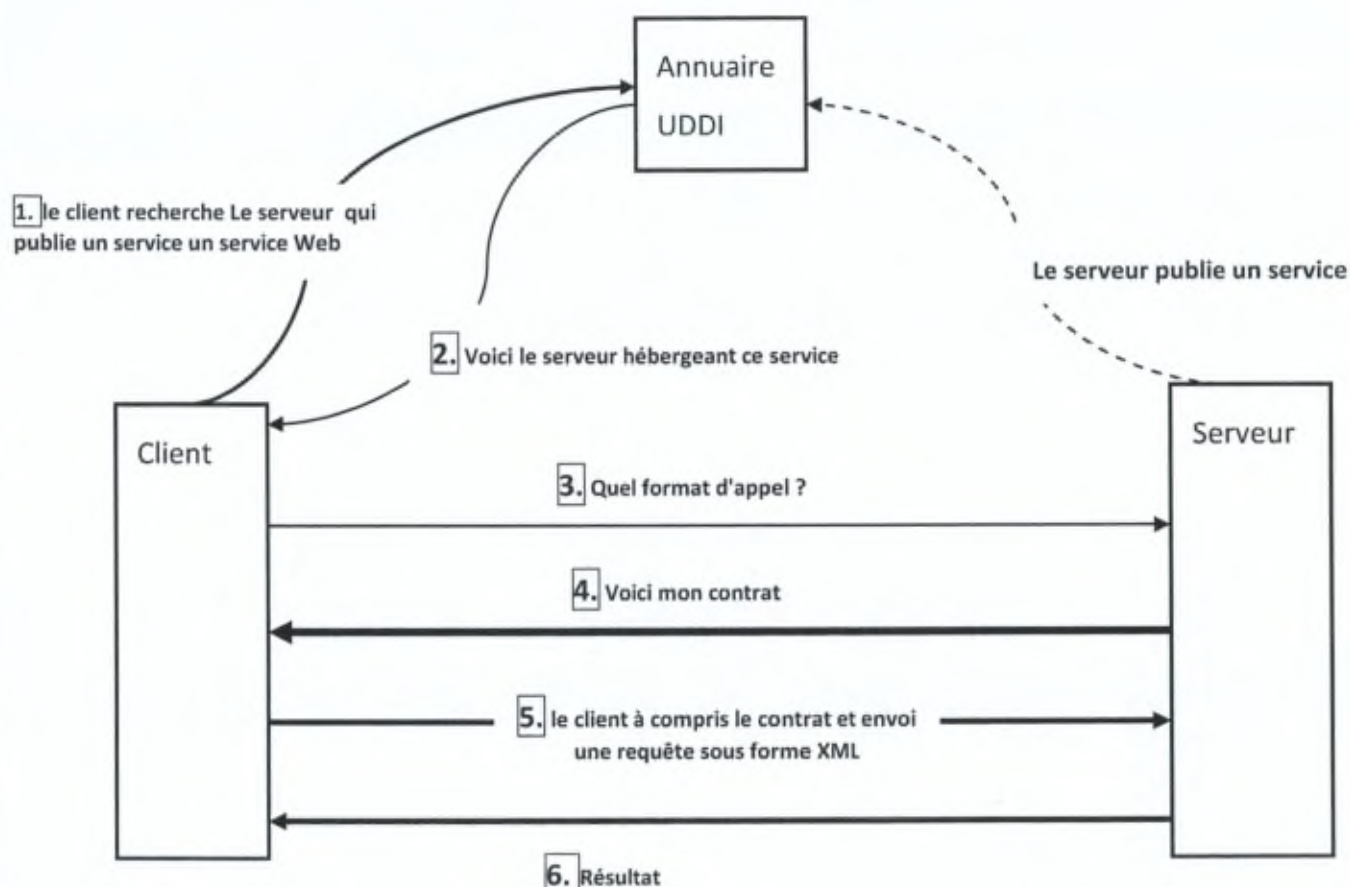


Figure 7.2 : Principe de fonctionnement d'un service Web, inspiré de (Tahé, 2009).

<sup>52</sup> Universal Description Discovery and Integration, annuaire universel des services Web.

<sup>53</sup> Web Services Description Language, langage de description d'un service Web.

### 7.3.3.2 Domaines d'utilisation

Dans leur usage sur le marché du Web, les services Web ont deux domaines principaux d'usage :

1. L'intégration des applications qui dans le monde de l'entreprise, vise à faire collaborer différentes applications dans le but d'atteindre les objectifs métiers de l'entreprise. Le but recherché est la réduction des coûts de transactions en automatisant la majorité des tâches.
2. Les portails car de part leur caractère modulaire, les services Web peuvent s'y intégrer. En effet, un portail est composé de petites applications indépendantes offrant des fonctionnalités à l'utilisateur. Moodle et Dokeos sont au sens Web des portails qui offrent plusieurs fonctionnalités modulaires aux utilisateurs (forum, chat, outils de suivi, etc.).

### 7.3.3.3 Infrastructure des services Web

Le concept de service Web s'appuie sur une infrastructure «en couches». Chaque partie fait désormais l'objet d'une normalisation<sup>54</sup>. Nous représentons à la figure 7.3 la pile structurale des Web Services.

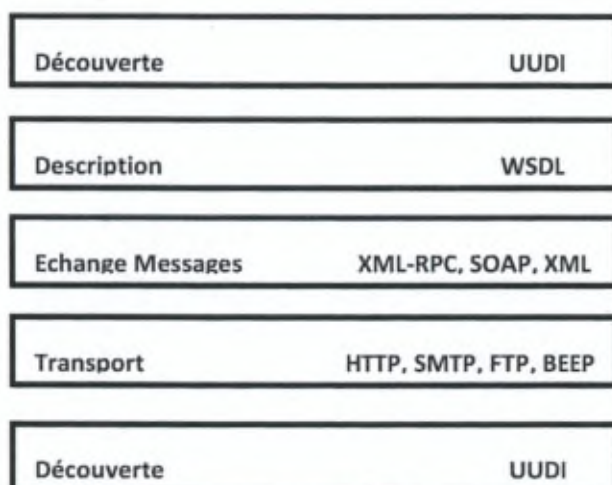


Figure 7.3 : Répartition en couches des protocoles des Web Services (Carbone, 2006).

<sup>54</sup> Les principaux organismes ou consortium sont : W3C, OASIS (*Organization for the Advancement of Structured Information Standards*), IETF (*Internet Engineering Task Force*)



### 7.3.3.3.1 La couche de transport

Cette couche se retrouve en première position de la base de la pile de protocoles nécessaires aux services Web. Elle a en charge le transport des messages XML entre entités communicantes. De nos jours, HTTP 1.1 reste le protocole majoritairement utilisé par les services Web. Des explications détaillées sont prévues en annexe.

### 7.3.3.3.2 La couche d'échange des messages XML

Dans les services Web, la couche d'échange des messages XML peut utiliser deux protocoles d'échanges à savoir XML-RPC et SOAP.

#### 7.3.3.3.2.1 XML-RPC

XML-RPC (*XML Remote Procedure Call*) est un protocole d'échange de messages XML ayant pour vocation d'appeler des méthodes à distance, à travers un réseau. Le serveur retourne une erreur ou une réponse toujours au format XML. En dépit de ces limitations, XML-RPC permet une grande variété d'actions souvent suffisantes à l'implémentation d'un service Web.

#### 7.3.3.3.2.2 SOAP

SOAP (*Simple Object Acces Protocol*) est un protocole gérant l'échange d'informations. Il utilise XML pour formater ses messages et HTTP pour les véhiculer. C'est en fait la spécification du format des messages XML échangés lors des invocations de services Web et de leurs réponses. Dans sa version 1.2, il s'étend au support d'autres protocoles que HTTP (TCP, UDP, etc.) tout en étant capable d'intégrer des fichiers attachés (par le biais de *SOAP Attachment* et *DIME*). A l'initiative de plusieurs acteurs, notamment de Microsoft, le protocole SOAP a été créé en 1999 sur les bases de la spécification XML-RPC. SOAP a été ensuite appuyé et supporté par une recommandation du consortium W3C (W3C, 2002). La version courante est 1.2 et date de 2003. La spécification de SOAP 1.2 se compose de quatre parties :

- la spécification de l'enveloppe SOAP ;
- les règles d'encodage ;

- les conventions d'invocation des méthodes ;
- l'utilisation d'un protocole de transport.

```

POST /forumMoodle HTTP/1.1
Host: www.unilu.ac.cd
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nnnn
<?xml version='1.0' ?>
  <SOAP-ENV:Envelope xmlns: SOAP-ENV
="http://www.w3.org/2001/12/soapenvelope">
    <SOAP-ENV:Body>
      <m:moodleForumPost
env:encodingStyle="http://www.w3.org/2003/05/soap-encoding"
xmlns:m=" "
      <m:messageEntrant xmlns:m=" ">
        <m:temps>201128060900</m:temps>
      </m:messageEntrant>
    </m:moodleForumPost>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Figure 7.4 : Exemple d'une requête HTTP POST transportant un message SOAP.

Les trois premières parties définissent le format du message XML dans la spécification SOAP. Il s'avère nécessaire à ce stade de visualiser et d'examiner la nature d'une requête SOAP transportée par HTTP dans son ensemble. Nous commençons par illustrer par quelques lignes de code un exemple de requête SOAP (figure 7.4) et sa réponse transportée par HTTP (figure 7.5).

```

HTTP/1.1 200 OK
Content-Type : application/soap+xml; charset="utf-8"
Content-Length : nnnn
<? xml version='1.0' ?>
  <SOAP-ENV:Envelope xmlns: SOAP-ENV
="http://www.w3.org/2001/12/soapenvelope">
    <SOAP-ENV:Body>
      <m:messageEntrantResponse xmlns:m=" ">
        <m: Message type="string">Nous avons fini le TP
N°4</m:Message>
      </m:messageEntrantResponse>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>

```

Figure 7.5 : Réponse à la requête HTTP.

### 7.3.3.3.3 La couche de description: WSDL

La couche la plus importante des services Web est sans conteste la couche de description, c'est elle qui décrit en profondeur le mécanisme de fonctionnement d'un



service Web. WSDL est l'acronyme de *Web Services Description Language*, c'est une spécification qui est née des apports conjoints de Microsoft, IBM<sup>55</sup> et leurs partenaires dans le but de réifier la couche de description. WSDL permet d'explicitier «complètement» un service Web à travers les types des données, les messages, les opérations, le *binding* (ou incarnation), ainsi que la méthode d'invocation, dans une grammaire XML commune. WSDL spécifie les quatre parties nécessaires à la mise en œuvre d'un service Web :

1. La description de l'interface aux méthodes publiques disponibles.
2. Le typage des données nécessaires aux requêtes et à leurs réponses.
3. Les informations sur la liaison avec le protocole de transport utilisé.
4. La localisation du service Web demandé.

On voit dans cette énumération que la description WSDL d'un service Web représente un contrat entre le demandeur et le fournisseur du service. Ainsi, WSDL s'apparente au contrat IDL (Khelifi, 2004) que l'on rencontre dans CORBA<sup>56</sup>.

La description avec WSDL se fait en utilisant la grammaire XML et s'articule autour des concepts suivants : un service pour une collection de ports (*port* ou *endpoints*), un port basé sur une adresse réseau et un *binding*. Un *binding* est fondé sur un protocole et un format de données associé à un type de port (*port type*). Un type de port étant un ensemble d'opérations proche d'une interface au sens Java. Une opération, qui représente une action proposée par un service Web, est décrite par ses messages (proche d'une méthode au sens Java), et un message est un ensemble de données. Une donnée est une information typée selon un système de type comme celui des schémas du W3C. Le fichier de description ainsi constitué est donc un fichier XML contenant les six éléments suivants tels qu'illustrés dans la figure 7.6 : <Type>, <Message>, <PortType>, <Service>.

Après avoir parcouru brièvement la couche de description avec WSDL, la littérature sur les services Web fournit, des modélisations UML des applications qui vont utiliser le WSDL dans le but d'implémenter un service Web (Provost, 2003).

<sup>55</sup> International Business Machines

<sup>56</sup> Common Object Request Broker Architecture, est une architecture logicielle, pour le développement de composants et d'Object Request Broker ou ORB.

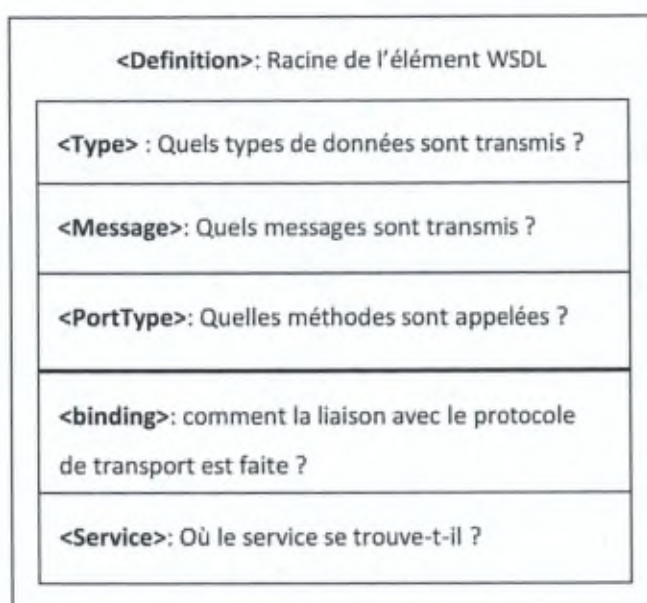


Figure 7.6: Les éléments de la spécification WSDL, extrait de (Carbone, 2006)

UML est donc souvent utilisé pour spécifier un document WSDL, notamment en permettant la modélisation des trois domaines des services Web à savoir : la sémantique des messages (WSDL), la spécification des types (XML Schema), et la conception Orientée-Objet traditionnelle des implémentations. La figure 7.8 propose un diagramme statique de la structure traditionnelle WSDL.

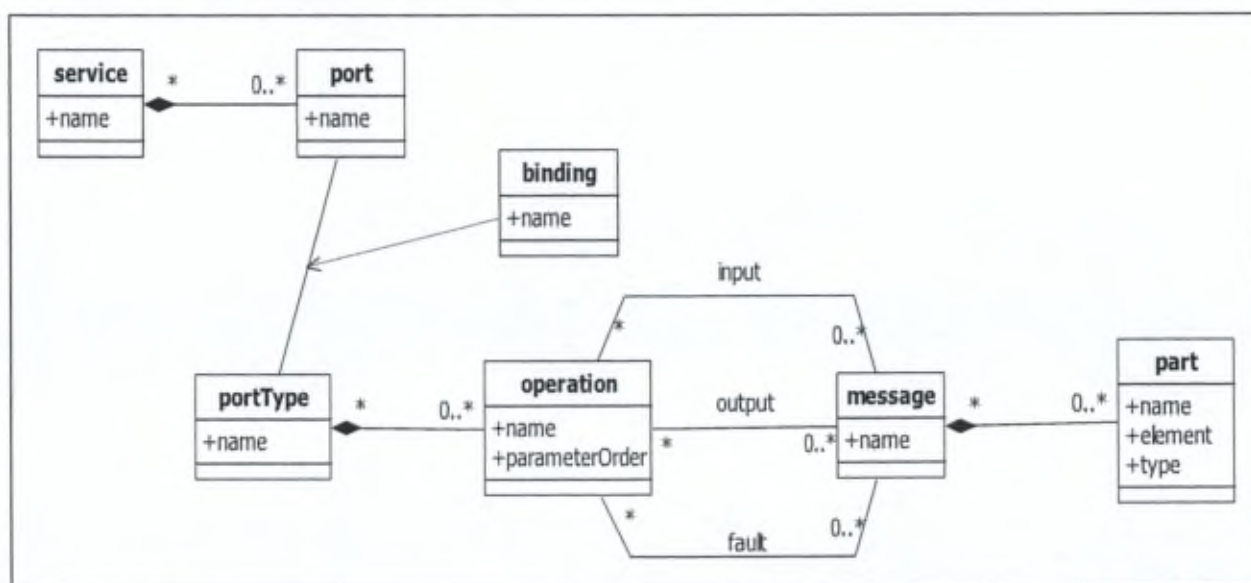


Figure 7.8 : Diagramme statique de la structure WSDL (Tahé, 2009).



## 7.4 Proposition de mise en œuvre du service Web des forums

Le déploiement d'un service Web demande, pour être disponible, un serveur d'applications. On a besoin, pour mettre en œuvre notre service Web faisant interagir les forums de la technologie des *servlets*<sup>57</sup> Java. Pour ce faire nous utiliserons l'environnement des solutions open source Apache :

- Axis1.4 qui est une implémentation java de la spécification SOAP c'est une structure pour construire des processus SOAP comme des clients, des serveurs, des passerelles, etc. ;
- Apache Jakarta Tomcat server 7.0 qui est un serveur d'applications exécutant les *servlets* Java.

Rappelons que notre intention ici est de transférer le contenu des messages de forum postés depuis la plate-forme Moodle vers Dokeos et vice-versa, sachant que chacune des deux plates-formes stocke les contenus des forums dans des bases de données dédiées au service de forum. Parmi les multiples possibilités, nous avons donc choisi d'implémenter un service Web d'accès générique à une base de données. Un service de ce genre peut avoir de multiples utilisations. Tout d'abord, il permet de s'affranchir du type de SGBD<sup>58</sup> utilisé (ici MySQL<sup>59</sup>). Il peut également servir lorsque l'on doit effectuer un transfert de base de données entre deux SGBD hétérogènes qui utilisent parfois des syntaxes SQL spécifiques. En extrayant le contenu d'une base dans un format générique (XML en l'occurrence), on peut effectuer cette traduction de n'importe quel SGBD vers n'importe quel autre et permettre ainsi un transfert des contenus d'une de ses bases des données vers une autre.

L'architecture globale du service Web à développer sera semblable à une architecture 3-tiers classique avec une séparation des données, des traitements et de la présentation des résultats. Nous illustrons la dite architecture par le schéma à la figure 7.9,

<sup>57</sup> Un *servlets* est une classe Java qui s'exécute dans un serveur Web, généralement à la suite d'une requête HTTP.

<sup>58</sup> Système de Gestion des Bases de Données.

<sup>59</sup> Système de Gestion de Base de Données relationnelles, open source voir <http://dev.mysql.com/>

qui montre également le rôle de chacun des outils à utiliser. Ici le client, sachons-le, peut être tour à tour Moodle ou Dokeos, pour une circulation bijective des échanges.

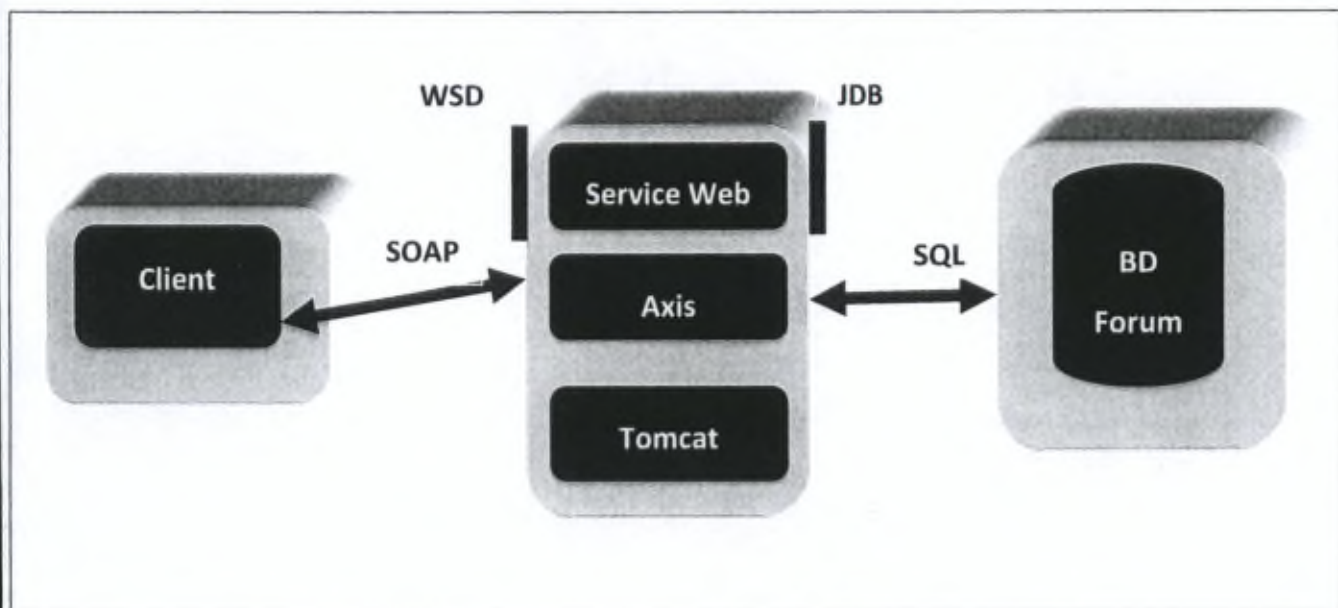


Figure 7.9 : Architecture du service Web des forums. (W3C, 2004)

Enfin, les messages SOAP étant exprimés dans le formalisme XML, nous utiliserons le parseur Java Xerces.

### 7.4.1 Description du processus métier

Pour implémenter un service Web pour l'échange des messages entre les forums des plates-formes, nous avons deux solutions pouvant être mises en œuvre. La première consiste à transmettre les résultats au client sous forme d'une chaîne XML générées dynamiquement et dont la structure (DTD) est connue à l'avance. La seconde consiste à transmettre les résultats sous forme d'objets sérialisés par Axis. Pour ce travail, nous adoptons la solution XML. Dans ce modèle, les données récupérées par le service auprès de la base de données sont intégralement décrites dans un format XML spécifique et transmises au client sous la forme d'une chaîne de caractère contenant ce XML.

Essayons donc pour illustrer l'idée de notre démarche, de définir une classe Java, basée sur JDBC<sup>60</sup>, une API Java pour interagir avec des bases de données relationnelles, qui

<sup>60</sup> Java Data Base Connectivity : API Java pour les bases de données.



va exécuter des requêtes SQL (statiques ou dynamiques), et récupérer ensuite les résultats sous forme de chaîne de caractère. Prenons, par exemple, le cas de la table «*mdl\_forum\_post*» de la base de données «*forum*» dans la plate-forme Moodle qui a la structure suivante décrite au tableau 7.1.

Tableau 7.1 : Structure de la table «*mdl\_forum\_post*» de Moodle

| Colonne    | Type         | Interclassement | Attributs | Null | Default | Extra          |
|------------|--------------|-----------------|-----------|------|---------|----------------|
| <u>id</u>  | bigint(10)   |                 | UNSIGNED  | Non  | Aucun   | AUTO_INCREMENT |
| Discussion | bigint(10)   |                 | UNSIGNED  | Non  | 0       |                |
| Parent     | bigint(10)   |                 | UNSIGNED  | Non  | 0       |                |
| Userid     | bigint(10)   |                 | UNSIGNED  | Non  | 0       |                |
| Created    | bigint(10)   |                 | UNSIGNED  | Non  | 0       |                |
| Modified   | bigint(10)   |                 | UNSIGNED  | Non  | 0       |                |
| Mailed     | tinyint(2)   |                 | UNSIGNED  | Non  | 0       |                |
| Subject    | varchar(255) | utf8_general_ci |           | Non  |         |                |
| Message    | Text         | utf8_general_ci |           | Non  | Aucun   |                |
| Format     | tinyint(2)   |                 |           | Non  | 0       |                |
| Attachment | varchar(100) | utf8_general_ci |           | Non  |         |                |
| Totalscore | smallint(4)  |                 |           | Non  | 0       |                |
| mailnow    | bigint(10)   |                 | UNSIGNED  | Non  | 0       |                |

Nous souhaitons en extraire les messages postés par un utilisateur par une requête (exemple : `SELECT* FROM mdl_forum_post`), ce qui aura pour résultat une chaîne de

caractère qui décrit le résultat de manière précise en intégrant à la fois la structure de la table et les données qu'elle contient. Cette chaîne pourra à son tour servir de manière complètement indépendante du SGBD utilisé, car les types SQL des champs seront décrits sous forme de métadonnées, ce qui constitue une prémisse d'interopérabilité des contenus des messages du forum. La classe java correspondante ressemblera au code de la figure 7.10.

```
public class MoodleForumPost {
    Connection conn =
DriverManager.getConnection("jdbc:mysql:localhost/phpmyadmin/moodle/mdl_forum_posts", "user", "passwd");

    java.sql.Statement st = conn.createStatement();

    ResultSet r=st.executeQuery("SELECT discussion, parent, userid,
created, modified, mailed, subject, message, format, attachment,
totalscore, mailnow FROM mdl_forum_post");

    while (r.next()) {
        // imprime les éléments du tuple

        int ladiscussion = r.getInt("discussion");
        int leparent = r.getInt("parent");
        int luserid = r.getInt("userid");
        int cree = r.getInt("created");
        int modification = r.getInt("modified");
        int envoye = r.getInt("mailed");
        String lesujet = r.getString("subject");
        String lemessage = r.getString("message");
        Int leformat = r.getInt("format");
        String piecejointe = r.getString("attachment");
        int lescore = r.getInt("totalscore");
        int ecriremaintenant = r.getInt("mailnow");

    }
}
```

Figure 7.10 : Extrait de code de la classe java du client.

Cette classe une fois placée dans le domaine applicatif d'Axis, son implémentation SOAP met à disposition un outil permettant de créer une spécification WSDL à partir d'une classe Java :

1. *Java2Wsd* permettant de créer la spécification WSDL du service Web à partir de la classe Java implémentée sur le serveur.
2. *Wsd2Java* qui permet d'implémenter les classes Java clientes à partir du document de description WSDL.



Ainsi la description de notre service Web pourra ressembler à celui de la figure 7.11

```
<wsdl:binding name="moodleForumPostSoapBinding" type="impl:moodleForumPost">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="main">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="mainRequest">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://DefaultNamespace" use="encoded"/>
      </wsdl:input>
      <wsdl:output name="mainResponse">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://127.1.1.0:8080/axis/moodleForumPost.jws" use="encoded"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="moodleForumPostService">
    <wsdl:port binding="impl:moodleForumPostSoapBinding" name="moodleForumPost">
      <wsdlsoap:address location="http://127.1.1.0:8080/axis/moodleForumPost.jws"/>
    </wsdl:port>
  </wsdl:service>
```

Figure 7.11 : Extrait de code de la description wsdl du service

Ce contrat WSDL ainsi obtenu, une fois transmis à l'application cliente, permettra de générer le code d'invocation à travers une classe souche interface (ou *Stub*) (Gibello, 2003).

## 7.4.2 Invocation du service

Pour tester la faisabilité, nous avons implémenté le service Web de recherche des contenus des messages postés dans le forum Moodle afin les envoyer dans le forum Dokeos. Il suffit d'invoquer une méthode particulière pour l'utiliser. Pour cela, il faut que le client puisse se connecter au service Web en s'appuyant sur la description WSDL. Ici il existe deux solutions à envisager :

1. On plante directement les commandes SOAP/HTTP dans le code à partir de l'analyse du document WSDL. Cette méthode est réservée à une utilisation statique, sans respect des patrons de conception<sup>61</sup>.

<sup>61</sup> En génie logiciel, un patron de conception (*Design pattern* en anglais) est une solution générique d'implémentation répondant à un problème spécifique.

2. On crée un proxy SOAP (en s'appuyant sur le design pattern Proxy) sous la forme d'un objet invocable par l'application cliente. C'est cet objet qui sera en relation avec le service Web. Il se chargera de traduire les requêtes émises par le client en requêtes SOAP, qui seront compréhensibles par le serveur. Cet objet proxy (ou *stub*) se chargera aussi de traduire les réponses SOAP renvoyées par le serveur. De nombreux éditeurs de logiciels fournissent les librairies (exemple NuSOAP<sup>62</sup>) nécessaires à la génération dynamique de proxy à partir des contrats WSDL.

Nous avons opté pour la deuxième solution, mais vu que les plates-formes Dokeos et Moodle sont toutes les deux conçues sous PHP, pour intégrer le service Web nous avons recouru à une implémentation en PHP de SOAP, soit NuSOAP. NuSOAP est un bon candidat disponible gratuitement sur Internet. Cet outil est constitué de plusieurs classes permettant la mise en place de clients et de serveurs SOAP en utilisant le langage PHP. Ceci permettra l'accès des deux plates-formes aux services Web.

NuSOAP s'installe simplement dans un serveur Web et est d'une utilisation aisée car il suffit de mettre sur le serveur un simple script PHP pour avoir accès à des services. Nous prévoyons dans les scripts des pages d'accueil des deux plates-formes, un bouton tel qu'une fois que l'on clique dessus, il démarre l'appel au service (). Ceci est faisable d'autant plus que sous NuSOAP la partie serveur est démarrée à chaque appel de l'URL du script implémentant la partie serveur pour SOAP.

```
<?php
require_once('../nusoap/nusoap.php');
$s = new soap_server;
$s->register('moodleForumPost',false,false,"moodleForumPost");
function moodleForumPost ($chaine){
// optionally catch an error and return a fault
if ($chaine == '') {
    return new soap_fault ('Client','', 'La chaine est vide. ');
}
$res="message envoyé:".$chaine;
return $res;
}
$s->service($HTTP_RAW_POST_DATA);
?>
```

Figure 7.12. Script serveur : déploiement du service MoodleForumPost

<sup>62</sup> Site officiel : <http://dietrich.ganx4.com/nusoap/> : site de NuSOAP (consulté en Août 2011)



Au niveau du code il suffira d'un `include` du fichier contenant les classes de NuSOAP pour pouvoir déployer ou invoquer des services comme nous le montrons à la figure 7.12. Voici successivement ce que fait le script serveur pour le déploiement du service :

1. Inclusion des classes NuSOAP.
2. Création d'un objet «serveur».
3. Enregistrement du service MoodleForumPost auprès du serveur.
4. Fonction effectuant le service et qui sera appelée par RPC.
5. Envoie des réponses du serveur au client : Ici, NuSoap va générer le message réponse du serveur en XML et l'envoyer dans la réponse HTTP au client.

En outre, NuSOAP permet, si l'on utilise les descripteurs de services écrits en WSDL comme c'est notre cas, de générer un proxy pour service. Ce proxy permet d'accéder à la méthode distante comme si elle était une méthode d'un objet local. Cela permet d'avoir un code beaucoup plus clair et moins lourd (voir figure 7.13).

```
<?php
require_once('../nusoap/nusoap.php');
$soapclient= new
soapclient(' http://127.1.1.0:8080/axis/moodleForumPost.jws ');
$proxy=$soapclient->getProxy();
$res=$proxy-> moodleForumPost ($lachaine);
echo $res;
```

Figure 7.13 : script client : appel direct service MoodleForumPost avec le fichier WSDL et le proxy

Sur le script de la figure 7.13, on effectue respectivement la création du client à partir d'un fichier WSDL, puis une création du proxy et enfin un appel à la procédure distante via le proxy.

## 7.5 Conclusion du chapitre

Ce chapitre, signalons-le encore une fois, est une étude de faisabilité d'une démarche d'interopérabilité dynamique. Il constitue une prémisse à un projet de développement qui peut être approfondie dans le cadre d'autres travaux et ouvre ainsi des perspectives des

recherches ultérieures. Nous avons identifié une problématique qui caractérise le suivi dynamique à l'aide des outils de communication asynchrone, en vue de permettre à un enseignant belge de communiquer avec ses étudiants de l'Université de Lubumbashi sans que les uns et les autres ne soient obligés de changer leur plate-forme d'apprentissage en ligne. Actuellement, une des technologies qui s'y prête le mieux est celle des services Web, car c'est une technologie Web qui permet de communiquer à travers le Web avec l'objectif notamment de supporter l'interopérabilité. L'interopérabilité entre services doit permettre la communication entre eux tout en gardant l'indépendance du rapport à un environnement technologique donné et la cohérence de chacun d'eux (Sanlaville, et al., 2005).

Nous avons dans ce chapitre rappelé les concepts de base de la technologie des services Web, puis nous les avons adaptés à notre problématique. Nous avons aussi proposé un schéma de conception pour arriver à déployer les services Web entre Moodle et Dokeos pour le cas des forums des deux plates-formes afin de permettre un transfert des contenus des messages entre eux. L'une des difficultés est que les plates-formes elles mêmes ne sont à ce jour pas suffisamment prêtes à implémenter les services Web. Du côté Moodle, il est signalé sur le site officiel<sup>63</sup> qu'à partir de la version 2.0, Moodle supportera plusieurs protocoles des services Web, notamment SOAP, XML-RPC, REST et AMF<sup>64</sup>. Côté Dokeos, aucune publication sur le site officiel, juste une indication<sup>65</sup> d'un groupe de développeur ayant réussi à implémenter un service Web (non flexible selon ses auteurs !), permettant l'importation d'un groupe d'utilisateurs au format CSV dans la plate-forme. Cette réalité oblige pour arriver à déployer un service Web entre Moodle et Dokeos au niveau technique, à prévoir le développement des extensions pour chacune des plates-formes capable d'implémenter les services Web. Ceci est possible car les deux plates-formes étant open source, il est facile d'en étendre les fonctionnalités. Un tel travail pourrait faire l'objet d'un financement supplémentaire.

<sup>63</sup> [www.moodle.org](http://www.moodle.org): site officiel de la plate forme Moodle (consulté en juillet 2011).

<sup>64</sup> REST: Representational State Transfer, AMF: Action Message Format.

<sup>65</sup> <http://beeznest.wordpress.com> (consulté en juillet 2011)



## CHAPITRE 8 : CONCLUSIONS ET PERSPECTIVES

Dans cette thèse nous nous sommes intéressés principalement, dans le cadre de l'e-learning, à mettre en œuvre une méthodologie qui a pour mission d'assurer une interopérabilité entre la plate-forme de l'Université de Lubumbashi et les plates-formes des universités partenaires belges afin de permettre un transfert des contenus d'apprentissage. Notre démarche n'a pas voulu s'arrêter à l'échange des contenus, nous sommes allés jusqu'à proposer des moyens de permettre un suivi dynamique des étudiants via les outils de communication. Nous concluons, maintenant, en présentant un bilan des travaux réalisés, puis, nous mettons en évidence la contribution de notre recherche. Enfin, nous terminons en exprimant et en discutant des perspectives de recherches à venir.

### 8.1 Parcours des réalisations

#### 8.1.1 Une étude du contexte d'application

Le travail a commencé par un chapitre qui a consisté à bien redéfinir les enjeux d'une telle étude en Afrique subsaharienne plus spécifiquement en République démocratique du Congo, pour faire ressortir le besoin de l'e-learning localement et sa portée régionale. Nous nous sommes intéressés à ce que représente l'interopérabilité entre les plates-formes d'apprentissage en ligne. Nous avons émis un avant-projet d'une application informatique permettant une interopérabilité à la fois statique et dynamique de l'Université de Lubumbashi. Puis nous finissons en montrant l'utilité de l'étude par rapport à l'enseignement supérieure dans la région, dans le sens où l'e-learning constitue une plus-value à la fois pour l'enseignement supérieur et pour la recherche scientifique.

Dans le deuxième chapitre nous avons démontré la pertinence du sujet et souligné que c'est la première fois qu'une telle étude était menée à Lubumbashi, et que nous évoluions donc en « terrain inconnu » et qu'il nous fallait caractériser le contexte pour évaluer la faisabilité d'une telle étude. Pour cela nous avons scruté le contexte scientifique de l'étude et le contexte du milieu dans lequel elle s'appliquait pour une première fois. Il est ressorti qu'une telle étude a fait appelle à plusieurs branches scientifiques allant des sciences pédagogiques aux sciences de l'ingénieur, en passant aussi par les théories des

sciences sociales. Pour le contexte du milieu d'application, nous avons mené une enquête pour déterminer le « profil informatique » de la ville de Lubumbashi. Cela afin de dégager les vraies chances de la réussite d'un projet d'apprentissage en ligne. Nous avons fait un inventaire du parc informatique de la ville pour déduire la faisabilité d'un tel projet et nous avons eu des résultats encourageants. Nos analyses ont aussi ressorti une tendance à la progression de l'usage des TIC surtout dans le milieu des étudiants, de la ville de Lubumbashi. On a aussi établi une croissance du nombre d'ordinateurs dans la ville de Lubumbashi allant presque du simple au triple en 3 ans.

Les observations, les analyses et les contextes nous ont permis de définir la problématique principale de nos travaux dont le questionnement essentiel tourne autour de l'apport de l'e-learning dans le contexte local des enseignements ainsi que de l'utilité de l'interopérabilité entre plates-formes d'apprentissage en ligne et son impact sur la coopération universitaire.

Au troisième chapitre nous avons fait un état de l'art sur les avancées accomplies en matière d'interopérabilité par les différentes plates-formes. Cela nous a amené à faire un tour d'horizons de toutes les normes et standards établis en la matière. Nous avons recensé tous les acteurs (organismes) œuvrant dans le domaine, avec une mention spéciale pour le modèle SCORM qui est celui dont la stratégie le propulse au rang de leader en matière standardisation pour l'interopérabilité dans l'e-learning. Il s'impose comme étant un standard de facto vu le nombre d'acteurs du secteur l'ayant déjà adopté.

Vu que depuis 2005, des tentatives d'établissement des projets e-learning ont vu jour à l'Université de Lubumbashi, il nous a paru capital de statuer sur l'état de l'e-learning en son sein. Nous sommes partis des expérimentations menées dans le cadre de la présente thèse avec un échantillon d'étudiants, pour dégager les comportements émergents d'une telle pratique, l'apport de l'encadreur (tuteur) et l'usage des outils. Tout ceci nous a permis d'établir un scénario narratif indiquant un cahier des charges en termes d'outils à utiliser. Cela pour un déroulement optimal d'une session d'apprentissage en ligne sur base de nos analyses des résultats des expérimentations. Nous avons relevé ici surtout la place prépondérante des outils de communication synchrone et asynchrone favorisant la



collaboration, vivifiant les échanges et diminuant le sentiment de solitude ressenti parfois au cours de telles sessions. On a également déterminé que la présence d'un tuteur était rassurante pour les apprenants.

### 8.1.2 Une étude technique

L'aspect technique de nos travaux démarre à partir du chapitre 4 où nous avons mené une étude approfondie du modèle SCORM qui nous servira comme modèle de référence en vue des échanges des contenus entre plates-formes. Le modèle SCORM, nous l'avons établi, repose sur une trilogie visant à donner une « couche informatique » aux travaux d'interopérabilité en e-learning.

Premièrement, il vise à structurer comment agréger et emballer le contenu à travers le CAM (*Content Aggregation Model*). Les concepts principaux étant la constitution d'un fichier zip, contenant toutes les ressources et à la racine duquel est placé un fichier XML où est déclinée toute la structure des fichiers physiques constituant une unité d'apprentissage. La description des contenus se fait à l'aide des métadonnées du LOM, reprises dans le fichier XML, faisant ainsi office de manifeste, et obéissant à des règles et exigences de structuration strictes établies dans le modèle.

Deuxièmement, le modèle SCORM crée un cadre d'exécution d'un dialogue entre un package constitué selon ses règles et une plate-forme qui lui est conforme. Tout ce dialogue de reconnaissance est implémenté dans ce qu'on appelle un RTE (*Run-Time Execution*). Il s'appuie sur un mécanisme standardisé permettant le dialogue et basé sur trois aspects caractéristiques : le lancement, l'interface de programmation d'applications (API) et le modèle de données. Le lancement démarre le dialogue, l'API détermine l'état de la ressource et le modèle de donnée c'est le CAM.

Troisièmement, le Séquencement (*SCORM Sequencing and Navigation*) qui sert à rythmer dans le temps les activités de l'apprenant par rapport aux ressources disponibles pour une formation.

Nous avons noté un succès grandissant de SCORM qui a notamment détrôné son grand rival AICC qui n'a visiblement pas réalisé que le support Web remplaçait irrémédiablement le support CD en matière d'apprentissage assisté par ordinateur. La stratégie remarquable du département de la défense américain (*DOD*) avec les laboratoires chargés de diffuser le plus possible le modèle, est un des facteurs de la progression de SCORM.

Au chapitre 5, nous nous attardions sur les deux plates-formes qui constituent notre cas d'étude, à savoir Moodle et Dokeos. Nous avons commencé par une brève présentation de chacune pour passer ensuite à une étude comparative des contenus de chaque plate-forme. Le constat ici est que Moodle implémente son propre format de cours à exporter uniquement entre plates-formes Moodle (pour autant qu'elles soient de même version d'ailleurs !). Son empaquetage est intrinsèquement défini par Moodle et la documentation sur la question est très peu disponible. Pour Dokeos, l'exportation d'un cours se fait de deux façons : soit il s'agit des fichiers que l'on disponibilise dans un répertoire, soit il s'agit d'un package SCORM pour autant que le cours contienne un « parcours pédagogique » prédéfini. Bien que la conformité SCORM des packages de Dokeos soit imparfaite, cela nous a permis de mettre SCORM au centre comme modèle d'interopérabilité entre Moodle et Dokeos.

Nous avons ensuite comparé les fonctionnalités de Moodle et Dokeos pour détecter les outils qui sont à la fois présents dans les deux plates-formes ainsi que leurs rôles au sein des plates-formes. Ensuite, nous nous sommes appuyés sur le modèle des trois « C », pour déterminer quelles sont, d'un point de vue pédagogique, les tâches que ces outils permettent d'accomplir. Et nous avons classé les outils selon un ordre hiérarchique bâti sur le nombre de tâches qu'ils permettent d'accomplir. A ce niveau, nous avons rejoint notre scénario narratif du chapitre 3 qui a déterminé le cahier des charges des outils nécessaires pour une session d'apprentissage en ligne adaptée à Lubumbashi ; car nous avons su les classer par ordre prioritaire.

Au chapitre 6, nous avons établi une méthodologie de conception d'une application informatique d'interopérabilité permettant un transfert des contenus de Moodle à Dokeos. On a rappelé que les contenus des deux plates-formes constituaient des objets



pédagogiques pouvant être décrits par des métadonnées spécialisées, notamment celles du LOM. Nous avons établi une correspondance entre les métadonnées propre à Moodle et celles définies par le LOM partant des règles d'adaptation et réadaptation. Techniquement, nous nous sommes appuyés sur le DOM pour implémenter la transformation des métadonnées Moodle pour les rendre conformes au SCORM. Puis nous avons présenté l'application informatique ainsi qu'une mise en œuvre de l'interopérabilité statique.

Le chapitre 7 part du souci du suivi à distance des étudiants par leur enseignant belge. Nous nous focalisons pour cela sur l'outil de communication asynchrone, le forum. Nous effectuons à ce stade une étude de faisabilité sur la meilleure implémentation d'un moyen technique d'échange des messages entre les forums de Moodle et ceux de Dokeos. Nous avons opté pour cela pour la solution des services Web et avons montré les moyens de sa mise en œuvre.

## 8.2 Notre contribution

Nos apports à travers ces travaux ont été de répondre en partie à de nombreuses problématiques que pose la question de l'e-learning en Afrique subsaharienne :

1. Cette étude, la première, a permis de ressortir au jour le profil informatique de la ville de Lubumbashi. Une initiative de ce travail qui va sans doute constituer un substrat utile pour des projets informatiques et des études futurs dans la ville de Lubumbashi.
2. Cette étude, la première, a permis de mettre sur pieds des sessions d'apprentissage en ligne dans un véritable cursus académique à l'Université de Lubumbashi lors de nos expérimentations. Elle nous a fourni les comportements émergents, l'usage effectif des outils d'une plate-forme dans le contexte locale et dans quel but.
3. Nous avons fourni une étude détaillée et approfondie du modèle SCORM, permettant d'apporter un plus à la documentation francophone sur SCORM. Cette dernière se limitant souvent à des articles traitant des aspects particuliers de SCORM.
4. Nous avons étudié et créé les moyens techniques de transfert des contenus entre Moodle et Doekos. Cette question n'avait pas été traitée auparavant par les

chercheurs (sauf le cas WebCT et Moodle). La plupart des recherches se focalisent sur le standard SCORM lui-même et sont très peu orientées sur les plates-formes majeures de l'e-learning.

5. Cette thèse introduit la notion d'interopérabilité dynamique qui consiste à faire communiquer entre eux les différents outils et services des diverses plates-formes. Nous proposons un moyen de mise en œuvre de cela à travers les services Web.
6. Cette étude apporte aussi des éléments de réponses aux problèmes que pose l'application de l'e-learning en Afrique subsaharienne, spécialement dans la région des grands lacs. Elle est la première à aller en profondeur de la question pour cette région du monde.
7. Dans le cadre de la coopération universitaire Nord-Sud, ce projet une fois abouti, constituera à coup sûr un facteur multiplicateur des échanges entre les universités.
8. C'est une première aussi que cette étude une fois finie permettra à la Faculté polytechnique de l'UNILU d'ouvrir un Département d'Ingénierie Informatique si nécessaire pour une université qui se veut désenclavée et moderne.

### 8.3 Perspectives des recherches

Les différents travaux de cette thèse participent à un processus itératif de conception ouvrant de nombreuses perspectives d'études futures pouvant être réparties sur le moyen et le long terme. Toutefois, notre contribution doit être suivie dans l'immédiat de deux travaux à courts termes :

- L'achèvement du développement au niveau technique de l'outil d'interopérabilité statique, cette application informatique se développera dans le cadre d'une démarche itérative sur le long terme ;
- L'achèvement de l'outil d'interopérabilité dynamique assurant une communication et un échange des messages entre les outils des différentes plates-formes. Cette méthodologie pourra être appliquée pour différentes plates-formes et garantir ainsi un suivi à distance dans le cas où la plate-forme de l'institution de l'enseignant est différente de la plate-forme des étudiants.



Ce travail ouvre au-delà de cela plusieurs perspectives pluridisciplinaires de recherche. D'abord l'étude menée lors de nos expérimentations pourrait être approfondie en utilisant, les théories des activités des sciences sociales. Elles permettront d'interpréter et de quantifier des ratios nécessaires aux analyses sur l'influence des interactions sociales au cours d'un apprentissage en ligne. De telles études amèneraient, à coup sûr, beaucoup plus d'éléments édifiants pour la réussite d'un projet e-learning en zone d'Afrique subsaharienne.

Le cas d'interopérabilité statique traité entre Moodle et Dokeos pourrait servir comme démarche de base pour des recherches similaires appliquées sur d'autres plates-formes d'e-learning. Cela avec un espoir d'aboutir à un « méta-modèle » qui, on l'espère, sera applicable à la majorité des plates-formes connues, apportant ainsi une contribution dans la résolution de la problématique de l'interopérabilité en e-learning.

Cette thèse a attiré notre attention sur l'apport que pourrait octroyer les services Web. Ceux-ci par leur indépendance vis-à-vis des plates-formes qui les supportent et des langages de programmation, peuvent s'avérer indispensables pour permettre des échanges entre services des plates-formes. Nous remarquons dans la littérature que cette option, pourtant prometteuse en matière d'interopérabilité n'est que rarement évoquée dans la communauté des développeurs des plates-formes. Ce qui, il nous semble, est guidé par la concurrence que se mènent les différents acteurs du secteur.

Notons aussi que l'e-learning quoique comportant plusieurs avantages avérés, n'est pas une sorte de « panacée » pour l'enseignement en Afrique subsaharienne. En effet certains cours ne se prêtent pas totalement à l'e-learning (par exemple les travaux de labo), d'où l'incontournabilité du présentiel qui devra accompagner des solutions e-learning.

Et pour finir, cette thèse s'inscrit dans le cadre d'une recherche pluridisciplinaire qui touche l'ingénierie informatique et plus particulièrement dans le domaine de l'apprentissage en ligne ou l'e-learning. Dès lors nos travaux nous ont offert l'opportunité de nous intéresser à plusieurs domaines allant des sciences sociales et de la pédagogie jusqu'à l'ingénierie informatique. C'est cette pluridisciplinarité qui nous a conduits à établir le profil informatique de la ville de Lubumbashi, à effectuer des expérimentations d'apprentissage en

ligne et d'en dégager les comportements émergents ainsi que les faits saillants. Nous avons eu recours à des sciences informatiques et plusieurs technologies nous ont servi au cours de notre méthodologie. Notre travail en informatique concerne encore plus la communauté de la programmation Web et du Web sémantique notre contribution représente une application de leurs méthodes et de leurs techniques, particulièrement le traitement des métadonnées.



## Bibliographie

**ADL, Advanced Distributed Learning. 2002.** *SCORM Conformance Requirements Version 1.2.*

[Online] Février 15, 2002. <http://www.adlnet.org>.

—. **2001.** *The SCORM Content Aggregation Model*. Sharable Content Object Reference Model(SCORM) version1.2, Octobre 1, 2001.

—. **2009.** *SCORM. ®. 2004 4 th. Edition Sequencing and Navigation (SN).* [Online] 2009. <http://www.adlnet.org>.

—. **2009.** *SCORM. ®. 2004 4 th. Edition Run-Time Environment (RTE).* [Online] 2009. <http://www.adlnet.org>.

**AICC. 2001.** *CMI Guidelines for Interoperability.* Technical Report CMI001, Revision 3.5, Aviation Industry CBT Committee, Avril 2, 2001.

—. **2002.** *Courseware Delivery Stations.* AICC Guidelines & Recommendations AGR-002 version 9.1, Aviation Industry CBT Committee, Février 06, 2002.

—. **1995.** *Courseware Interchange.* AICC Guidelines & Recommendations AGR-007 version1.0, Aviation Industry CBT Committee, Août 29, 1995.

—. **1998.** *Computer Managed Instruction.* AICC Guidelines & Recommendations AGR-006, version 2.0, Aviation Industry CBT Committee, Mai 19, 1998.

**AIF-INTIF. 2002.** *Rapport de l'atelier " Logiciels libres: enjeux stratégiques pour l'Afrique".* Conférence régionale Afrique, Sommet mondial sur la société de l'information, Bamako, 2002.

**AILF. 2002.** *Traduction en français de LOM.* [Online] 2002. <http://www.aillf.net>.

**Arnaud, Michèle. 2002.** *Normes et standards de l'enseignement à distance : enjeux et perspectives.* Actes du colloque Technologies de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie (TICE 2002), Novembre 13 au 15, 2002.

**AUF. 2002.** *Normalisation de la formation en ligne - Enjeux, tendances et perspectives.*

[Online] Février 2002. Agence Universitaire de la Francophonie, Bureau Amérique du nord.  
<http://ameriquenord.auf.org>.

**Baumgartner, Peter. 2004.** *Didaktik und Reusable Learning Objects (RLO's).* In: Campus 2004  
 - Kommen die digitalen Medien an den Hochschulen in die Jahre? , 2004.

**Bernd, Amann. 2001.** *Données Semistructurées et XML.* Conservatoire National des Arts et  
 Métiers. Paris : s.n., 2001.

**Bersini, H. 2009.** *La programmation orientée objet: Cours et exercices en UML2 avec Java 5,  
 C#2, C++, Python, PHP5 et LINQ.* Paris : Eyrolles, 2009.

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

**Betbeder, Marie-Laure and Tchounikine, P. 2003.** *Symba: a Framework to Support Collective  
 Activities in an Educational Context.* International Conference on Computers in Education  
 (ICCE 2003 ), Décembre 2-5, 2003.

**Bourda, Yolaine and Hélier, Marc. 2003.** *Métadonnées et XML.Applications aux "Objets  
 pédagogiques".* Supélec, Plateau de Moulon, F-91192 Gif-sur-Yvette cedex France, 2003.

**Burgos, D, et al. 2005.** *IMS Learning Design:la flexibilité pédagogique au service des besoins  
 de l'e-formation.* Publications and Preprints Keur der Wetenschap., Octobre 27, 2005.

**Candace Chou, C. 2002.** *A comparative content analysis of student interaction in  
 synchronous and asynchronous learning networks.* HICSS 2002:134, 2002.

**Carbone, Cedric. 2006.** *Cours de Web dynamique.* [Online] 2006. IUT Evry, Licence  
 Professionnelle IRS2I. <http://www.iut.univ-evry.fr>.

**Cardinaels, K, Meire, M and Duval, E. 2005.** *Automating Metadata Generation: the Simple  
 Indexing Interface.* 14th International Conference on World Wide Web, 2005.



**Charlier, B. 1998.** *Apprendre et changer sa pratique d'enseignement : expériences d'enseignants*. Bruxelles : De Boeck, 1998.

**Charlier, B, Daele, A and Dschryver, N. 2002.** *Apprendre en collaborant à distance: Ouvrons la boîte noire*. in Guir, R Pratiquer les TICE : Former les enseignants et les formateurs à de nouveaux usages, 2002.

**Chikh, A. 2008.** *Modélisation avec IMS-LD des scénarios d'apprentissage*. 2008, Vol. CEMAFORAD4.

**Cojean, F. 2007.** *Manuel du programmeur Moodle debutant*. [Online] 2007.  
<http://docs.moodle.org>.

**Crepuq. 2002.** *Les normes et standards de la formation en ligne - Etat des lieux et enjeux*. [Online] Septembre 2002. Conférence des Recteurs et des Principaux des Universités du Québec. <http://profetic.org>.

**CSC. 2003.** *Les learning management systems (LMS) - Solutions logicielles de gestion de la formation- étude annuelle*. [Online] 2003. Technical report. <http://csc.com>.

**Daft, R L and Lengel, L H. 1984.** *Information richness: a new approach to managerial behavior and organizational design*. Research in organizational behavior 6, 1984.  
Homewood, IL: JAI Press.

**David, Bertrand. 2001.** *IHM pour les collecticiels*. Hermès, Les télé-applications, 2001, Vol. 13.

**DCMI, Dublin Core Metadata Initiative. 1995.** *The Dublin Core Metadata Initiative Limited*. [Online] 1995. Copyright © 1995-2011 DCMI. <http://dublincore.org>.

—. **2004.** *Politique de nommage des termes du DCMI*. [Online] 04 05, 2004.  
<http://dublincore.org/documents/naming-policy/>.

**De la Passardière, Brigitte and Monique, Grandbastien. 2007.** *Présentation de LOM v1.0, standard IEEE*. Revue Sciences et techniques éducatives Hors série, Mars 2007.

**De Lièvre, Bruno and Depover, Christian. 2001.** *Apports d'une modalité de tutorat proactive ou réactive sur l'utilisation des aides dans un hypermédia de formation à distance.* Actes du Ve Colloque Hypermédias & Apprentissages, Mai 9-11, 2001.

**De Lièvre, Bruno, Depover, Christian and Acierno, Melanie. 2006.** *Analyse du soutien fourni aux apprenants par les tuteurs à l'aide d'outils synchrones et asynchrones.* Colloques JOCAIR 06, Première journée communication et Apprentissage instrumentés en réseaux, Juillet 2006.

**De Lièvre, Bruno, Depover, Christian and Dillenbourg, P. 2006.** *The relationship between tutoring mode and learners' use of help tools in distance education.* Instructional Science, 2006, Vol. 34.

**Degoulet, P, Fieschi, M and Attali, C. 1997.** *Les enjeux de l'interopérabilité sémantique dans les systèmes d'information de santé.* Informatique et Santé, 1997, Vol. 9.

**Depover, Christian and De Lièvre, Bruno. 2005.** *Analyse des usages des outils de communication médiatisés par ordinateur dans le cadre de deux scénarios de formation à distance.* Colloque Symfonic, 2005.

**Dorel, Gorga. 2003.** *Exposé 2: Métadonnées et XML – une initiation.* TECFA, 2002 – 2003 Taf 14, Juin 2003.

**Duval, E, et al. 2002.** *Metadata principles and practicalities.* D-Lib Magazine, 2002.

**Ecoutin, Eric. 2001.** *Fiche pratique no.1 : les utilisations d'une plate-forme.* [Online] Mars 2001. Technical report. <http://capfoad.chez.com>.

**Ellis, C and Wainer, J. 1994.** *A conceptual model of groupware.* In Proceedings of CSCW'94.ACM Press, 1994, pp. 79-88.

**Fage, C. 2005.** Vous avez dit « SCORM »? elearning Agency, 2005.



**Ferber, J and Gutknecht, O. 1998.** *A meta-model for the analysis and design of organisations in multi-agents systems.* Third International Conference on Multi-Agent Systems(ICMAS'98) Proceedings, 1998, pp. 128-135. IEEE Computer Society.

**Ferber, J. 1995.** *Les systèmes multi-agents : vers une intelligence collective.* s.l. : Interditions., 1995.

**Ferraris, C and Martel, C. 2000.** *Regulation in groupware: The example of a collaborative drawing tool for young children.* Proceedings of CRIWG'2000, 6th international workshop on groupware., 2000, pp. 119-127.

**Friesen, N. 2004.** *Three Objections to Learning Objects and E-learning Standards.* [ed.] R McGreal. Online Education Using Learning Objects. London : Routledge, 2004. pp. 59-70.

**Fyama, Blaise. 2006.** *Conception d'une plate-forme open source d'e-learning en R.D Congo: cas de l'université de Lubumbashi.* Mémoire de DEA, Université Libre de Bruxelles, Septembre 2006.

**George, W, Gagnon, Jr and Michel, C. 2005.** *Constructivist Learning design- Key Questions for teaching to Standards.* s.l. : Corwin Press, 2005.

**Gibello, Pierre-Yves. 2003.** *ICAR'03.* INRIA Rhône Alpes, 2003.

**Glikman, V. 1999.** *Formation à distance: au nom de l'usager.* DistanceS. 3.2, 1999, pp. 101-117.

**Gorga, Dorel. 2003.** *Exposé 2: Métadonnées et XML – une initiation.* TECFA, 2002 – 2003 Staf 14., Juin 2003.

**Gounon, P. 2005.** *Encadrement d'apprenants à distance: Etude du soutien informatique à la conception d'une formation en ligne fondé sur un modèle d'organisation du tutorat.* s.l. : Thèse de doctorat,LIUM, Université du Maine, 2005.

**Gounon, Patricia, Dubourg, X and Leroux, P. 2004.** *Un modèle d'organisation du tutorat pour la conception de dispositifs informatiques d'accompagnement des apprenants.* Actes de TICE'2004, Octobre 20-22, 2004. Édité par Université de Technologie de Compiègne.

**Gounon, Patricia, Leroux, P and Dubourg, X. 2005.** *Décrire l'accompagnement des apprenants - Proposition d'une extension du langage de modélisation pédagogique IMS-Learning Design.* EAIH'05, Mai 27, 2005.

**Grandbastien, M. 2002.** [ed.] Baron G.-L. et Bruillard E. *Quelques questions à propos de l'indexation et de la recherche de ressources pédagogiques sur le Web.* Les technologies en éducation : Perspectives de recherche et questions vives, 2002, pp. 211-220.

**Greenberg, J. 2002.** *Metadata and the World Wide Web.* Encyclopedia of Library and Information Science, 2002.

**Guéraud, V, et al. 2004.** *L'exploitation d'Objets Pédagogiques Interactifs à distance: Le FORMID.* Revue STICEF, 2004, Vol. 11, pp. 109-163. en ligne sur [www.sticef.org](http://www.sticef.org).

**Henri, F and Lundgren-Cayrol, K. 2001.** *Apprentissage collaboratif à distance: Pour comprendre et concevoir les environnements d'apprentissage virtuels.* Sainte-Foy : Presses de l'université du Québec., 2001.

**Hodgins, H W. 2002.** *The future of learning objects.* ECI Conference on e-Technologies in Engineering Education: Learning Outcomes Providing Future Possibilities, Août 11-16, 2002. Retrieved September 29, 2003, from <http://www.coe.gatech.edu>.

**IEEE. 2001.** *IEEE Learning Object Metadata.* [Online] Septembre 26, 2001. <http://ltsc.ieee.org>.

—. **2002.** *Draft Standard for Learning Object Metadata.* [Online] Juillet 15, 2002. LTSC. <http://ltsc.ieee.org>.

**IMS, Global Learning Consortium. 2001.** *IMS Content Packaging XML Binding Version 1.1.2.* [Online] Août 2001. Final Specification. <http://www.imsglobal.org>.



- . **2004.** *IMS Vocabulary Definition Exchange*. Best Practice and Implementation Guide, Fevrier 2004.
- . **2006.** *IMS Question and Interoperability*. [Online] 2006. Public Draft(revision2) Specification. <http://www.imsglobal.org>.
- . **2007.** *IMS Content Packaging Specification Primer, version 1.2*. [Online] Mars 2007. <http://www.imsglobal.org>.
- . **2003.** *IMS Learning Design, Information model, version1.0*. [Online] Janvier 20, 2003. Final Specification. <http://www.imsglobal.org>.
- . **2002.** *IMS Question & Test Interoperability*. [Online] 2002. ASI Best Practice & Implementation Guide, IMS Global Learning Consortium, Inc. <http://www.imsglobal.org>.
- . **2003.** <http://www.imsglobal.org>. *IMS Learning Design Best Practice and Implementation*. [Online] 2003.
- Johnson, D and Johnson, R. 1998.** *Cooperative learning and social interdependence theory*. Cooperative learning, 1998.
- Khelifi, Karim. 2004.** *Cours de Systèmes informatiques répartis*. Université de Laval, 2004.
- Koper, R. 2003.** [ed.] A. Littlejohn. *Combining re-usable learning, resources and services to pedagogical purposeful units of learning*. Reusing Online Resources: A sustainable approach to eLearning, 2003, pp. 46-59.
- Koper, R. 2001.** *Modelling units of study from a pedagogical perspective. The pedagogical meta-model behind EML*. [Online] 2001. <http://eml.ou.nl>.
- Laforcade, P. 2006.** *Représentation graphique des scénarios pédagogiques abstraits: expérimentation entre IMS-LD et UML*. TICE, 2006.
- Leclercq, Eric, Savonnet, Marinette and Terrasse, Marie-Noëlle. 2003.** *Adaptation d'une plateforme d'e-learning à un modèle pédagogique*. Katholieke Universiteit Leuven, 2003. Conference, Proceedings of 3rd Annual Ariadne.

**Lejeune, 2006.** *IMS Learning Design. Etude d'un langage de modélisation pédagogique.* Formamente, 2006, pp. 29-58.

**Lejeune, A. 2004.** *IMS Learning Design.* Distances et Savoirs, 2004, Vol. 2, pp. 409-450.

**Leroux, P. 2002.** *Machines partenaires des enseignants et des apprenants – Etude dans le cadre des environnements supports de projets pédagogiques.* Université du Maine, 2002. Habilitation à Diriger les Recherches de l'Université du Maine.

—. **1995.** *Conception et réalisation d'un système coopératif d'apprentissage-Etude d'une double coopération : maître/ordinateur et ordinateur/groupe d'apprenants.* Thèse de doctorat en informatique, Université Paris 6, 1995.

**Mason, R. 1992.** *Application of electronic communication for distance education in the third world.* BANCMC; University of Calgary, 1992. Bangkok Project.

**Mbala, A. 2003.** *Analyse, conception, spécification et développement d'un système multi-agents pour le soutien des activités en formation à distance.* Thèse de doctorat, Université de Franche-Comté, Octobre 2003.

**Miao, Yongwu. 2000.** *Design and Implementation of a Collaborative Virtual Problem-Based Learning Environment.* PhD thesis, Die Dissertationsschrift Vorgelegt am Fachbereich Informatik Der Technischen Universität Darmstadt von, 2000.

**Millet, P and Vaudron, D. 2009.** *Transfert Dokeos-Moodle .* MATICE de Montpellier, Mai 2009.

**Morel-Pair, Catherine. 2007.** *Prépublication de « Métadonnées et XML : des standards efficaces de l'environnement numérique ».* Ingénierie des systèmes d'information, 2007, Vol. 12, 2, pp. 9-39.

**Negroponte, N. 1997.** *Agents : From Direct Manipulation to Delegation, In Software Agents.* [ed.] J.M.Bradshaw. Menlo Park, Californie : AAAI Press, 1997.

—. **1995.** *Being Digital.* New York : Alfred Knopf, 1995.



**Nelson, Ted. 1965.** *Complex information processing: a file structure for the complex, the changing and the indeterminate.* ACM '65 Proceedings of the 1965 20th national conference, 1965.

**Newcomer, Eric. 2002.** *Understanding Web Services: XML, WSDL, SOAP, and UDDI.* s.l. : Addison-Wesley Professional, 2002. 0201750813.

**Nilsson, M, et al.** *Expressing Dublin Core metadata using the Resource Description Framework (RDF).* [Online] <http://dublincore.org>.

**NISO, National Information Standards Organisation. 2004.** *Understanding Metadata.* [Online] 2004. <http://www.niso.org>.

**Nodenot, T. 2007.** *Scénarisation pédagogique et modèles conceptuels d'un EIAH: Que peuvent apporter les langages visuels?* International Journal of Technologies in Higher Education, 2007, Vol. 2, 4.

**OMG. 2003.** *Unified Modeling Language v1.5 Specification.* [Online] Mars 2003. Report formal/03-03-01. <http://www.omg.org>.

**Oubahssi, Lahcen. 2005.** *Conception de plates-formes logicielles pour la formation à distance, présentant des propriétés d'adaptabilité à différentes catégories d'utilisateurs et d'interopérabilité avec d'autres environnements logiciels.* Thèse de doctorat, Université René Descartes, Paris V, 2005.

**Paquette, G. 1998.** *L'ingénierie des interactions dans les systèmes d'apprentissage.* Revue des sciences de l'éducation, Septembre 1998.

**Pecquet, E. 2007.** *Développer des cours en ligne avec Dokeos 1.8.* [Online] Avril 2007. <http://www.dokeos.org>.

**Pernin, Jean-Philippe, Prieur, Michèle and Sanchez, Eric. 2007.** *Stratégies d'élaboration, de réutilisation et d'indexation de scénarios.* Colloque SCENARIOS, 2007.

**Pernin, Jean-Philippe. 2003.** *Objets pédagogiques : unités d'apprentissage, activités ou ressources ?*. Revue "Sciences et Techniques Educatives", Hors série 2003 " Ressources numériques, XML et éducation", Avril 2003, pp. 179-210.

—. 2003. Ressources numériques, XML et éducation. Revue "Sciences et Techniques Educatives", avril 2003, pp. 179-210.

**Pernin, J-P and Lejeune, A. 2004.** *Modèles pour la réutilisation des scénarios d'apprentissage*. Actes du colloque TICE méditerranée, 2004.

—. 2004. *Dispositifs d'apprentissage instrumentés par les technologies : vers une ingénierie centrée sur les scénarios*. TICE 2004, Octobre 2004, pp. 407-414.

**Piombo, C. 2007.** *Modélisation probabiliste du style d'apprentissage et application à l'adaptation de contenus pédagogiques indexés par une ontologie*. Thèse de doctorat, Université de toulouse, Octobre 2007.

**Polsani, Pithamber R. 2003.** *Use and Abuse of Reusable Learning Objects*. Journal of Digital Information, 2003, Vol. 3, 4. Learning Technology Center, University of Arizona, USA.

**Provost, Will. 2003.** *UML for Web services*. [Online] Mai 08, 2003. <http://XML.com>.

**Psyché, V. 2007.** *Rôle des ontologies en ingénierie des EIAH: Cas d'un système d'assistance au design pédagogique*. Thèse de doctorat, Université du Québec, Juillet 2007.

**Pujolle. 2008.** *Les réseaux*. Eyrole. Paris : s.n., 2008.

**Rodet, J. 2000.** *La rétroaction, support d'apprentissage Distance? DistanceS, 4,2, 2000.*

**Salomon, G. 1993.** *Distributed cognitions : Psychological and educational Considérations*. New York : Cambridge University, 1993.

**Sanlaville, S and Estublier, J. 2005.** *Un environnement de modélisation et de coordination de services*. RSTI-ISI, 2005, Vol. 10, 3.

**Schneider, D. 2006.** *La norme Learning design, Technologie Internet et Education*. TICE, 2006.



- Simard, Cyrille. 2002.** *Normalisation de la formation en ligne, Enjeux, tendances et perspectives*. [Online] Février 2002. AUF, BAN. <http://NordSud.org>.
- Smith-Yoshimura, Karen and Cellentani, Diane. 2007.** *RLG Programs Descriptive Metadata Practices*. [Online] 2007. Report produced by OCLC Programs and Research. <http://www.oclc.org>.
- Steinmetz, R, Rensing, C and Lehmann, L. 2007.** [ed.] C. Montgomerie & J. Seale. *Lifecycle Information Management and Utilization in an Authoring by Aggregation Environment*. Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007, 2007, pp. 3142-3149.
- Strijker, A. 2004.** *Reuse of Learning Objects in Context: Technical and Human Aspects*. Faculty of Behavioural Sciences, University of Twente, Enschede, Netherlands., 2004. PhD dissertation.
- Tahé, Serge. 2009.** *Construire un service web Java EE avec l'IDE Netbeans 6.5 et le serveur Java EE Glassfish*. [Online] Février 2009. <http://developpez.com>.
- Tarpin-Bernard, Franck. 1997.** *Travail coopératif synchrone assisté par ordinateur : Approche AMFC*. Thèse de doctorat, Doctorat de l'Ecole Centrale de Lyon, spécialité Ingénierie Informatique, 1997.
- Uyttebrouck, Eric. 2002.** *WebCT et la normalisation*. Centre des Technologies pour l'enseignement (CTE), Université Libre de Bruxelles. 2002.
- Valayer, C. 2004.** *E-learning...Une introduction. Définitions, outils et enjeux*. Séminaire e-learning, mars 2004.
- Vantroys, Thomas. 2003.** *Du langage métier au langage technique, une plate-forme flexible d'exécution des scénarios pédagogiques*. Université des Sciences et Technologies de Lille, Décembre 16, 2003. Thèse de doctorat.
- W3C. 1999.** *Transformation XSLT(XSLT) version1.0*. [Online] 1999. Recommandation W3C du 16 novembre 1999. [www.w3.org/TR/WD-xsl](http://www.w3.org/TR/WD-xsl).

—. 1999. *Langage XML Path (XPath) version 1.0*. [Online] 1999. Recommandation du W3C du 16 novembre 1999. [xmlfr.org/W3C/TR/xpath](http://xmlfr.org/W3C/TR/xpath).

—. 2002. *SOAP Version 1.2, Préliminaires*. [Online] 2002. Nilo Mitra. <http://www.w3.org>.

**W3C, Consortium.** 2009. *Namespaces in XML 1.0*. [Online] December 8, 2009. W3C recommendation. <http://www.w3.org>.

**W3C, World Wide Web Consortium.** 2004. *Web Services Architecture*. [Online] Février 2004. W3C Working Group Note 11. <http://www.w3.org/TR/ws-arch/>.

**Wiley, David A.** 2000. *Learning Object design and sequencing theory*. Department of Instructional Psychology and Technology, 2000. Dissertation.

**Wisc-Online.** 2007. *Wisc-Online*. [Online] Février 1er, 2007. <http://www.wisc-online.com>.



## ANNEXE A : PRESENTATION DES SITES

Nous signalons que les sites ont été obtenus par un maillage carré de la ville de Lubumbashi et les dénominations ont été choisies librement par l'équipe d'enquête. Elles sont inspirées le plus souvent soit par le nom des avenues délimitant la maille, soit carrément par le nom de la commune dans laquelle l'enquête est menée, nous en reprenons six seulement.

### A.1. Campus de la Kasapa

Le premier site à être exploré est le site du campus universitaire de la Kasapa, de près de neuf km<sup>2</sup>, il est le plus proche des étudiants et tous les types d'entités informatiques définis dans ce travail s'y retrouvent. Nous avons notamment : sept Cybercafés, six bureautiques et deux centres de formation. Pour les cybercafés ils possèdent en moyenne 10 ordinateurs chacun principalement de processeur Intel Pentium 3 et 4. Ils n'utilisent pas d'onduleurs, tout le matériel de bureautique (imprimantes, photocopieuses, graveurs, etc.) est également présent. Tous utilisent comme antivirus AVIRA antivir gratuit sous licence GPL. Quatre Fournisseurs d'Accès Internet (FAI) donnent des services (Comax, Vodanet, UNILU, Microcom). Les heures d'ouvertures sont en moyenne de 13 heures par jour. Les bureautiques et centres de formation sont huit au total et sont concentrés principalement dans les homes<sup>66</sup> du campus : ils possèdent en moyenne six ordinateurs chacun, et gardent globalement les mêmes caractéristiques que les cybercafés. Avec les heures d'ouverture moyenne de 13 heures et demi, elles fonctionnent un peu plus longtemps que les bureautiques.

### A.2. Quartier Golf

Notre deuxième site est situé au sud-ouest de la ville de Lubumbashi et est adjacent au site de la Kasapa pour une superficie de près de 144 km<sup>2</sup>. On y trouve deux types d'entités définies ici, les cybercafés et les bureautiques, les centres de formation y sont absents. Pour ce qui est des cybercafés il y en a quatre avec en moyenne sept ordinateurs chacun, l'unique système d'exploitation est Windows. La majorité ici utilise des onduleurs,

<sup>66</sup> Home: Logement des étudiants sur le site de Kasapa

tout le matériel de bureautique classique (Imprimante, Photocopieuse, Graveur, Scanner etc.) y est présent. Notons ici l'usage de quatre antivirus différents à savoir Norton, NOD 32, AVIRA, Kaspersky. Deux FAI se partagent la zone, il s'agit de Vodanet et Comax. Les heures d'ouvertures sont en moyenne de 11 heures par jour. Les bureautiques sont trois dans ce site et possèdent en moyenne deux ordinateurs et ont à peu près les mêmes caractéristiques techniques que les cybercafés avec un fonctionnement moyen de 13 heures par jour nettement supérieur à celui des cybercafés.

### **A.3. Quartier Carrefour**

Ce site est situé entre le campus universitaire de la Kasapa, et le centre Ville et a une superficie de près de 64 km<sup>2</sup>. Il possède deux types d'entités informatiques pour quatre cybercafés et deux bureautiques. Les cybercafés ont en moyenne 10 ordinateurs et le système d'exploitation est toujours Windows. On note aussi ici l'usage des onduleurs, un nombre assez élevé d'ordinateurs et un matériel relativement neuf, tout le matériel bureautique s'y trouve (imprimantes, photocopieuses, scanner, graveurs, etc.). Les logiciels de sécurité utilisés ici sont tous payants (Norton, Karspersky, BitDefender) avec une licence valide. Deux fournisseurs d'accès se partagent ce marché, Vodanet et Microcom. Les heures d'ouverture sont en moyenne de 13 heures par jour. En ce qui concerne les bureautiques, avec nettement moins d'ordinateurs, soit cinq en moyenne, elles gardent néanmoins le même profil technique que les cybercafés et travaillent en moyenne sur la même durée, soit 13 heures d'ouverture par jour.

### **A.4 Kimbangu-L.D Kabila**

Ce site qui amorce le centre de la ville de Lubumbashi par L'Ouest a une superficie de plus ou moins 36 km<sup>2</sup>. Il est aussi doté de plusieurs entités informatiques de deux catégories, les cybercafés et les bureautiques. Les cybercafés sont au nombre de 11 ce qui montre une concentration assez élevée un cybercafé pour trois km<sup>2</sup>, soit deux cybers en moyenne par avenue. On notera tout de même que cinq cybercafés se trouvent sur une même avenue, il s'agit de l'avenue Kasai. Les ordinateurs se répartissent en moyenne de neuf ordinateurs par cybercafé, le système d'exploitation utilisé est Windows. Ici, en



majorité, il y a usage des onduleurs et tout l'attirail des périphériques de bureautique y est présent notamment des imprimantes, scanners, photocopieuses, graveurs, etc. Notons que la grande majorité n'utilise que des antivirus gratuits pour la sécurité, quatre fournisseurs d'accès Internet occupent cet espace avec plus de 50% du marché pour Microcom. Les heures d'ouverture sont en moyenne de 12 heures par jour. Les bureautiques quant à elles sont au nombre de cinq avec aussi cinq ordinateurs en moyenne pour chaque bureautique, les caractéristiques techniques restant pour la plupart similaires à celles des cybercafés. Leurs heures d'ouvertures sont en moyenne légèrement inférieures pour atteindre 12h.

### **A.5. Révolution-Kimbangu-Lumumba-Kamanyola.**

Cette zone est située entre le site Quartier-Golf et le site Kimbangu-L.D Kabila. Comme son nom l'indique, elle est délimitée par quatre grandes artères de la ville de Lubumbashi (Avenue de la Révolution, Kimbangu, Lumumba, Kamanyola). Avec ses près de 49 km<sup>2</sup>, elle compte toutes les catégories des entités informatiques à savoir les cybercafés, les bureautiques et les centres de formation. Nous y dénombrons six cybercafés ce qui fait un cyber pour huit km<sup>2</sup> avec en moyenne six ordinateurs, le système d'exploitation y est toujours Windows. Quatre cybercafés sur six utilisent des onduleurs, mais tous proposent les dispositifs et périphériques de bureautique (imprimantes, graveurs, photocopieuses, scanner etc.). Tous les cybers sont ici dotés d'antivirus sous licence légale (Kaspersky et Norton) à l'exception d'un seul qui aurait un crack d'AVIRA version professionnelle. Deux fournisseurs d'accès se partagent cet espace (Comax et Vodanet), et les heures d'ouvertures sont évaluées à 10h30 par jour. Les bureautiques sont au nombre de cinq, avec cinq ordinateurs en moyenne, leurs caractéristiques demeurent similaires à celles des cybercafés. On notera ici que toutes sont protégées par des antivirus sous licence légale et les heures d'ouvertures sont en moyennes de 12 heures par jour. Dans ce site comme nous l'avons dit plus haut, nous rencontrons à nouveau des centres de formation aux caractéristiques techniques similaires à celles des deux entités précédentes, mais elles possèdent exceptionnellement une densité élevée en nombre d'ordinateurs, soit 39 ordinateurs par centre. Ce qui est remarquable car c'est un chiffre qui est environ quatre fois supérieur à la moyenne des autres sites toutes entités confondues. Leurs heures d'ouvertures sont de 10 heures par

jour. Notons que pour ce site, le personnel d'encadrement est en grande partie qualifié c'est à dire de formation technique d'informaticien contrairement au reste des sites où le personnel est souvent formés sur le tas. Les exceptions notées pour ce site sont dues au fait que tous les centres de formation appartiennent à des institutions universitaires, qui sont relativement dotées des moyens de se fournir en matériel et personnel d'appoint. Contrairement aux entités des autres sites qui sont majoritairement des initiatives privées de particuliers moins nantis.

## A.6 L.D Kabila-Likasi

Ce site situé en plein centre de la ville de Lubumbashi est d'une superficie de 25 km<sup>2</sup>. On y rencontre donc sans surprise toutes les trois catégories d'entités informatiques. Les cybercafés y sont au nombre de 11 soit un cybercafé pour environ deux km<sup>2</sup> avec en moyenne neuf ordinateurs. Le système d'exploitation est encore une fois Windows, avec une négligence remarquable en terme d'onduleurs pour la sécurité des PC (sans doute que le courant électrique y est plus stable). Les dispositifs et périphériques de bureautique y sont quasi complets (imprimantes, photocopieuses, scanner, graveurs, etc.). Dans la plupart des cybercafés on retrouve aussi un espace réservé à la vente de matériels informatiques divers vu qu'on est en plein centre commercial de la ville. Sept cybers ont un antivirus gratuit (AVIRA) sous licence GPL, tandis que trois seulement possèdent un antivirus payant. Cinq fournisseurs d'accès se partagent cet espace, les heures d'ouverture sont ici en moyenne de 12h30 par jour. Les bureautiques sont au nombre de 10. Cependant, malgré leur nombre assez significatif dans le contexte de la ville de Lubumbashi, nous remarquons une présence importante de matériels vétustes avec, par exemple, une absence marquée de photocopieuses, des imprimantes le plus souvent en panne. Les conditions de travail y sont relativement modestes, et en moyenne nous comptons quatre ordinateurs par bureautique. Les heures d'ouvertures sont aussi estimées à 11 heures par jour. Quant aux centres de formation, nous en dénombrons quatre avec une moyenne de 15 ordinateurs par centre. Notons aussi au passage qu'il s'agit ici des centres qui ont en moyenne aussi près de 20 ans d'existence, et où la formation en informatique est souvent couplée avec celle de l'anglais.



Le personnel de formation est aussi relativement bien formé et expérimenté, et ces centres fonctionnent en moyenne 11h 30 par jour.

## ANNEXE B : TABLEAUX DE COMPARAISON DES OUTILS

Les tableaux non repris ici sont en entièreté dans le texte du présent travail.

### B.1 Outils de contrôle

Tableau B.1 : Comparaison des outils de contrôle.

| Plate-forme                     | Moodle   | Dokeos |
|---------------------------------|--|--------|
| QCM (Question à Choix Multiple) | QCM mais aussi des Questions à appariements, des questions à réponses numérique, des questions à réponses courtes, des tests limités dans le temps (date limite de rentrée, date de disponibilité) | Oui    |
| Résultats et notes              | Oui  | Oui    |
| Statistiques de suivi des cours | Oui  | Oui    |
| Contrôle des connexions         | Oui, accompagné d'un journal détaillé.   | Oui    |

### B.2 Outils de partage.

Tableau B.2 : Comparaison des outils de partage.

| OUTILS DE PARTAGE  |   |   |
|--|---|---|
| Plate-forme  | Moodle  | Dokeos  |
| Dépôts de documents  | Oui   | Oui   |
| Mise en ligne de cours   | Oui, structuration du contenu par rubriques :<br>- banque d'exercices ;<br>- intégration multimédia aux questions ;<br>- audio/vidéo ;<br>- flash ;<br>- import/export d'exercices d'autres systèmes. | Oui   |
| Salle de cours (partage tous les documents avec tous les apprenants)   | Oui   | Oui   |
| Salle de TP (partage des documents limités à des groupes d'apprenants) | Il existe trois modalités de fonctionnement de groupes : aucun groupe, groupes séparés, groupes visibles.<br>1. Dans le mode séparé, chaque   | Oui, les membres du groupe ne se voient qu'entre eux. |

|                              |  |                              |
|------------------------------|--|------------------------------|
|                              | <p>membre d'un groupe ne voit que les membres de son groupe, les activités des autres groupes sont masquées et non accessibles pour l'observation.</p> <p>2. Dans le mode visible, chaque membre peut voir les activités de son groupe au sein d'un forum par exemple et observer les activités des autres groupes sans y participer.</p> <p>3. Il n'existe pas de groupe du tout.</p> |                              |
| Version des documents        | Tous les formats de fichiers   | Tous les formats de fichiers |
| Historique                   | Apprendre c'est aussi une activité d'observation. Un étudiant peut ainsi observer l'évolution d'un autre.  | Oui                          |
| Cours (audio, visuel...)     | Oui  | Oui                          |
| Tableau blanc                | Non  | Non                          |
| Ouverture, alertes, flux RSS | Possibilité d'afficher des flux RSS de l'extérieur   | Oui                          |

## B.3 Aspects pédagogiques

Tableau B.3 : Comparaison des aspects pédagogiques des outils.

| ASPECTS PEDAGOGIQUES                           |   |   |
|--|---|---|
| Plate-forme                                    | Moodle  | Dokeos  |
| Nombre d'utilisateurs maximum de la plateforme | Illimité.   | Illimité.   |
| Profils et rôles                               | <p>Les administrateurs peuvent tout faire sur le site, dans tous les cours. Les créateurs des cours peuvent créer de nouveaux cours et y enseigner.</p> <p>Les enseignants peuvent tout faire dans un cours, y compris ajouter et modifier les activités et donner des notes aux étudiants.</p> <p>Les enseignants non éditeurs peuvent enseigner dans leur cours et donner des notes aux étudiants, mais ne peuvent ni ajouter, ni modifier des activités.</p> <p>Les étudiants ont en général moins de privilèges dans un cours. Les visiteurs anonymes ont</p> | <p>Le formateur gère les outils pour les rendre visibles et exploitables par les apprenants, qui suivent les cours.</p> <p>L'administrateur modifie les droits et les profils des utilisateurs si nécessaire.</p> |



|                                  |  |   |
|----------------------------------|--|---|
|                                  | très peu de privilèges et ne peuvent normalement saisir de texte à aucun endroit.  |   |
| <b>Évaluation et contrôle</b>    | Oui  | Oui   |
| <b>Fonctions de l'enseignant</b> | <p>Plate-forme mise en place en se fixant pour objectif le socioconstructivisme.</p> <p>Mais les formateurs ont les rôles suivants :</p> <ul style="list-style-type: none"> <li>- rédiger la description du cours ;</li> <li>- publier le cours ;</li> <li>- administrer des forums de discussion ;</li> <li>- gérer une liste de liens ;</li> <li>- créer des groupes de participants ;</li> <li>- composer des exercices ;</li> <li>- publier des annonces ;</li> <li>- consulter des statistiques de fréquentation et de réussite aux exercices.</li> </ul> | <ul style="list-style-type: none"> <li>- rédiger la description du cours ;</li> <li>- publier le cours ;</li> <li>- administrer des forums de discussion ;</li> <li>- gérer une liste de liens ;</li> <li>- créer des groupes de participants ;</li> <li>- composer des exercices ;</li> <li>- publier des annonces ;</li> <li>- consulter des statistiques de fréquentation et de réussite aux exercices.</li> </ul> |

## B.4 Aspects techniques

Tableau B.4 : Comparaison des aspects techniques des outils

| ASPECTS TECHNIQUES                |   |  |
|-----------------------------------|---|--|
| Plate-forme                       | Moodle  | Dokeos   |
| Format propriétaire               | Oui   | Non  |
| Personnalisation de la plateforme | Oui   | Oui  |
| Ergonomie de la plateforme        | Menu à droite et cours au centre, navigation facile car l'arborescence de tous les cours apparaît sur la page principale. | L'administrateur peut manipuler l'ergonomie de toute la plate forme, tandis que l'enseignant ne peut configurer que l'ergonomie de son propre cours. |
| Maintenance et mise à jour        | Oui, hebdomadaire.  | Oui  |
| Environnement informatique        | Fonctionne avec de nombreux systèmes d'exploitation.  | Fonctionne avec de nombreux systèmes d'exploitation.   |
| Langues                           | Multilingue   | Multilingue (supporte 34 langues à ce jour).   |
| Sécurité                          | Oui, identifiant + mot de passe.  | Oui, identifiant + mot de passe.   |

## ANNEXE C : QUELQUES NOTIONS UTILES SUR XML

### C.1 Structure d'un document XML

#### C.1.1 L'arbre d'éléments

Un document XML peut se représenter sous la forme d'une arborescence d'éléments qui comporte une racine unique, des branches et des feuilles. Ceci pour exprimer des relations de filiation, relations parentales, fraternelles, etc., entre les différents éléments.

##### C.1.1.1 Élément racine

L'élément racine est la base du document XML. Il est unique et englobe tous les autres éléments. Il s'ouvre juste après le prologue, et se ferme à la toute fin du document. C'est l'exemple de l'élément `<manifest>` dans un fichier *imsmanifest.xml*.

##### C.1.1.2 Les éléments

Les éléments forment la structure même du document : ce sont les branches et les feuilles de l'arborescence. Ils peuvent contenir du texte, ou bien d'autres éléments, qui sont alors appelés « éléments enfants ». L'élément contenant étant quant à lui appelé logiquement « élément parent », deux éléments de même niveau dans l'arborescence sont des éléments frères ou *siblings*, L'ensemble de tous les éléments contenus dans un autre élément constitue sa descendance.

Exemple d'élément contenant du texte : `<module> Base de données 1</module>`.

Exemple d'élément contenant d'autres éléments à la figure C.1

```
<module>
  <chapitre> Le modèle conceptuel</chapitre>
  <auteur> Blaise Fyama</auteur>
  <apprenant statut="stagiaire" />
</module>
```

Figure C.1 : exemple d'élément container.



Un élément peut être aussi vide c'est-à-dire se retrouver sans aucun enfant ni descendant. Nous en donnons un exemple à la figure C.2.

```
<musique genre="rumba" />
```

Figure C.2 : exemple d'élément vide.

### C.1.1.3 Les attributs

Un attribut est une caractéristique définie de l'élément, tous les éléments peuvent contenir un ou plusieurs attributs. Un attribut ne peut figurer qu'une seule fois au sein d'un élément. Un attribut est composé d'un nom et d'une valeur. On ne le place que dans la balise ouvrante de l'élément.

```
<cours type="informatique">Programmation 1</cours>
```

Figure C.3 : exemple d'un attribut.

```

```

Figure C.4 : exemple d'utilisation d'un élément vide avec attributs.

### C.1.1.4 Les entités

Il existe des entités définissables et définies. Elles peuvent être analysables ou non, internes ou externes. La déclaration des entités s'effectue au sein de la DTD. Elles peuvent être utilisées aussi bien dans la DTD que dans le document XML. Pour certains caractères ayant un sens précis en XML (par exemple "<"), il est nécessaire de leur trouver un remplaçant lorsque l'on a besoin de les insérer dans un document. On a recours dans ce cas à des entités prédéfinies données au tableau C.1.

Tableau C.1 : Liste des entités prédéfinies

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

### C.1.1. Les sections CDATA

Une section CDATA (*Characters Data*) est une section pouvant contenir toutes sortes de chaîne de caractères, mêmes les caractères réservés au langage. Une section CDATA permet de définir un bloc de caractères ne devant pas être analysés par le processeur XML. Cela permet, entre autres, de garder dans un bloc de texte un extrait de code à afficher tel quel.

```
<![CDATA [Une balise commence par un < et se termine par un >]]>.
```

Figure C.5 : exemple d'utilisation de CDATA.

### C.1.2 Modèles de documents : DTD et schémas

XML propose l'implémentation de modèles de documents décrivant la structure, les éléments et attributs, et même les valeurs autorisées. Le parseur<sup>67</sup> XML effectue la validation de chaque instance de document par rapport au modèle déclaré. Ces modèles sont de deux types : la *Document Type Description* (DTD) vue précédemment et le schéma XML (*XML Schema*).

La DTD, issue du monde SGML, a été le modèle le plus utilisé au départ. Sa syntaxe est relativement simple et ramassée conformément à l'esprit XML, mais sa puissance reste limitée. Le schéma XML, de syntaxe plus complexe, présente de nombreux avantages :

- c'est lui-même un document XML ;

<sup>67</sup> Logiciel spécialisé pour parcourir, lire et analyser un document XML.



- il permet d'imposer des contraintes plus précises sur l'occurrence des éléments et sur le modèle des valeurs attendues ou des expressions régulières par exemple ;
- il permet de déclarer des espaces de nom (ensembles d'éléments fermés décrits de manière formelle). La déclaration de chaque espace de nom inclut son adresse et un préfixe suivi de deux points qui sera précisé devant les éléments correspondants (ceux-ci restent toujours reliés à leur origine) ;
- il autorise l'utilisation de schémas XML externes et leur adaptation à l'application locale par divers mécanismes ;

Le schéma XML permet ainsi de créer des « profils d'application » spécifiques adaptés aux applications locales tout en respectant une interopérabilité de base. La majorité des projets XML sont basés sur les schémas standards desquels on modifie les contraintes ou on y ajoute des éléments issus d'autres espaces de nom ou locaux. Le XML Schéma est maintenu par le Consortium W3C.

### C.1.4 XPath

XPath est un premier langage de requête qui permet de désigner des nœuds de l'arbre en fonction de leur position, du nom et du contenu des éléments, des attributs et des valeurs d'attribut. C'est un module de base utilisé par tous les autres pour retrouver des éléments à travers un arbre XML. Il implémente notamment des axes (ou chemins) de recherche que nous présentons sommairement dans le tableau C.2.

### C.1.5 XSL et XSLT

Le processus de transformation XSLT (*XML StyleSheet Transformation*) fait appel à des feuilles de style XSL qui s'appliquent aux documents XML pour créer un résultat, soit à la volée et lisible par un navigateur, soit en dur, cas le plus fréquent pour les métadonnées. Les résultats sont variés : fichiers de diffusion, notamment pages HTML et documents PDF, nouvelles ressources issues de l'extraction de parties du document ou de la concaténation de plusieurs fichiers ou parties de fichiers, ou encore changement des noms des éléments et attributs pour correspondre à un autre modèle de document.

XSL/XSLT constitue un véritable langage de programmation dans la syntaxe XML. Il comprend des tests et des boucles imbriqués, des compteurs et variables, et une manipulation évoluée des chaînes de caractères. Des processeurs XSLT sont intégrés aux éditeurs XML et aux divers environnements informatiques. Les programmes peuvent leur passer des paramètres et des variables, notamment pour traiter des lots de fichiers ou concaténer des données. Une feuille de style XSLT consiste en un ou plusieurs ensembles de règles appelées « *Templates* » à appliquer à un nœud spécifique lorsqu'il est rencontré par le parseur.

Tableau C.2 : Axes XPath et résultats des recherches<sup>68</sup>.

| Nom de l'axe       | Résultats  |
|--------------------|--|
| ancestor           | Sélectionne tous les ancêtres (parents, grands parents, etc.) du nœud courant.                               |
| ancestor-or-self   | Sélectionne tous les ancêtres (parents, grands parents, etc.) du nœud courant ainsi que le nœud lui-même.    |
| attribute          | Sélectionne les attributs du nœud courant.   |
| child              | Sélectionne tous les enfants du nœud courant.  |
| descendant         | Sélectionne tous les descendants (enfants, petits enfants, etc.) du nœud courant.                            |
| descendant-or-self | Sélectionne tous les descendants (enfants, petits enfants, etc.) du nœud courant ainsi que le nœud lui-même. |
| following          | Sélectionne tous les éléments qui dans le document viennent après la balise de fermeture du nœud courant.    |
| following-sibling  | Sélectionne tous les frères (enfants du même parent) du nœud courant.  |
| namespace          | Sélectionne tous les nœuds espaces de nom du nœud courant.   |
| parent             | Sélectionne le parent du nœud courant  |
| preceding          | Sélectionne tous les éléments du document placés avant la balise d'ouverture du nœud courant.                |
| preceding-sibling  | Sélectionne tous les frères (enfants du même parent) du nœud courant.  |
| self               | Sélectionne le nœud courant.   |

<sup>68</sup> Tiré de <http://www.w3schools.com>: site officiel des tutoriels des standards et spécifications du W3C (consulté en mai 2011).



Nous donnons quelques éléments importants de XSLT :

`<xsl :template>`. Cet élément est utilisé pour construire les modèles (templates).

`<xsl :value-of>`. Cet élément peut être utilisé pour extraire la valeur d'un élément XML et l'ajouter dans le flux de sortie de la transformation.

`<xsl :for-each>`. Cet élément peut être utilisé pour sélectionner chaque élément d'un ensemble spécifique des nœuds.

`<xsl:sort>`. Cet élément une fois ajouté dans `<xsl :for-each>` du fichier XSL permet de trier les éléments à la production.

`<xsl:if>`. Cet élément permet de poser un choix conditionnel sur un fichier XML lorsqu'on l'intègre dans un fichier XSL destiné à une production.

`<xsl :choose>`. Cet élément est utilisé en conjonction avec les éléments `<xsl :when>` et `<xsl :otherwise>`. Pour exprimer un test conditionnel à choix multiple.

`<xsl:apply-template>`. Cet élément est utilisé pour appliquer un template (un modèle) à l'élément courant ou aux éléments enfants du nœud courant.

## ANNEXE D : QUELQUES NOTIONS DE SERVICES WEB

### D.1 La couche de transport

Elle a en charge le transport des messages XML entre entités communicantes. De nos jours, HTTP 1.1 reste le protocole majoritairement utilisé par les services Web.

#### D.1.1 Le protocole HTTP

HTTP est l'acronyme de *HyperText Transfert Protocol*. C'est un protocole réputé simple, stable et largement déployé. Sa structure autorise le transport de messages XML-RPC ou SOAP, au même titre que de simples messages HTTP. Il se base principalement sur le modèle client/serveur, mais en le limitant à la sémantique Requête/Réponse. Un client HTTP émet une requête à un serveur HTTP pour demander une ressource et ce dernier retourne un message réponse, la connexion est toujours amorcée par le client tandis qu'elle est fermée par le serveur.

#### D.1.2 Structure des messages HTTP

Que ce soit en requête ou en réponse, les formats sont similaires et sont composés de :

1. Une ligne initiale composée de 3 parties : le nom de la méthode, l'identifiant unique de la ressource et la version HTTP utilisée, c'est une ligne obligatoire.
2. Une ou plusieurs lignes d'en-tête optionnelles. Seul l'en-tête «*Host*», représentant l'adresse du serveur est devenue obligatoire depuis la version 1.1 du protocole.
3. Une ligne blanche obligatoire.
4. Un corps de message, qui peut être optionnel.
5. Toutes les lignes doivent être terminées par CR/LF (caractères Ascii 13 et 10). On remarque que seul le contenu de la ligne initiale permet de différencier la requête HTTP de sa réponse.



Le protocole HTTP fait recours à un certain panel des méthodes pour fonctionner et nous citerons à titre indicatif les méthodes suivantes:

- la méthode GET, elle est utilisée pour obtenir une ressource identifiée par son URI (*Uniform Resource Identifier*) ;
- la méthode POST, elle est utilisée pour envoyer des données au serveur et déclencher un traitement. Un corps de message est joint à la requête. Il contient les données nécessaires au traitement ;
- la méthode OPTIONS, elle permet de connaître les options de communication utilisables pour obtenir une ressource.

Il existe de nombreuses autres méthodes utilisées par le protocole HTTP notamment HEAD, PUT, DELETE, TRACE, CONNECT. Traditionnellement le serveur envoie une réponse HTTP à une requête HTTP et cette réponse a une ligne initiale appelée la ligne de statut. Elle est également composée de trois parties : le code statut de la réponse, une phrase explicative de ce code ainsi que la version HTTP. Dans le cas d'une requête réussie, la ressource est retournée dans le corps du message.

### D.1.3 La couche d'échange des messages XML

Dans les services Web, la couche d'échange des messages XML, peut utiliser deux protocoles d'échanges à savoir XML-RPC et SOAP.

#### D.1.3.1 XML-RPC

XML-RPC (*XML Remote Procedure Call*) est un protocole d'échange de messages XML ayant pour vocation d'appeler des méthodes à distance, à travers un réseau. Il a été créé en 1998, c'est un protocole simple qui fonctionne par appel de procédures paramétrées au formalisme XML. Le serveur retourne une erreur ou une réponse toujours au format XML. Le typage des paramètres est limité. Les structures et les tableaux sont les types les plus complexes que propose XML-RPC. En dépit de ces limitations, XML-RPC permet une grande variété d'actions souvent suffisantes à l'implémentation d'un service Web. Notons tout de

même qu'à ce jour cette technologie quoique plausible est de moins en moins utilisée et ne connaît pas une grande diffusion.

### D.1.3.2 SOAP

SOAP (*Simple Object Access Protocol*) c'est un protocole gérant l'échange d'informations. Il utilise XML pour formater ses messages et HTTP pour les véhiculer.

### D.1.3.3 L'enveloppe SOAP

L'enveloppe représente l'unité de communication qui s'établit entre les parties utilisant un service Web. Quel que soit le type du contenu (RPC ou Document) ou le modèle de messages utilisés (Requête-Réponse, Message, Requête), sa structure est constante et est décrite dans les illustrations des figures D.1 et D.2 représentant respectivement la structure de l'enveloppe SOAP et son diagramme des classes UML. Classiquement dans le code d'une spécification SOAP 1.2, l'enveloppe est définie par l'élément XML : `<SOAP-ENV:Envelope>`, et dans notre exemple à la section 7.3.3.3.2.2, le corps de l'enveloppe encapsule l'invocation d'une méthode distante. La méthode `messageEntrant` appartient au service `moodleforumPost`. Le paramètre `temps` est passé à cette méthode. Dans la spécification SOAP, chaque paramètre doit être nommé, typé et ordonné comme dans la signature de la méthode invoquée.

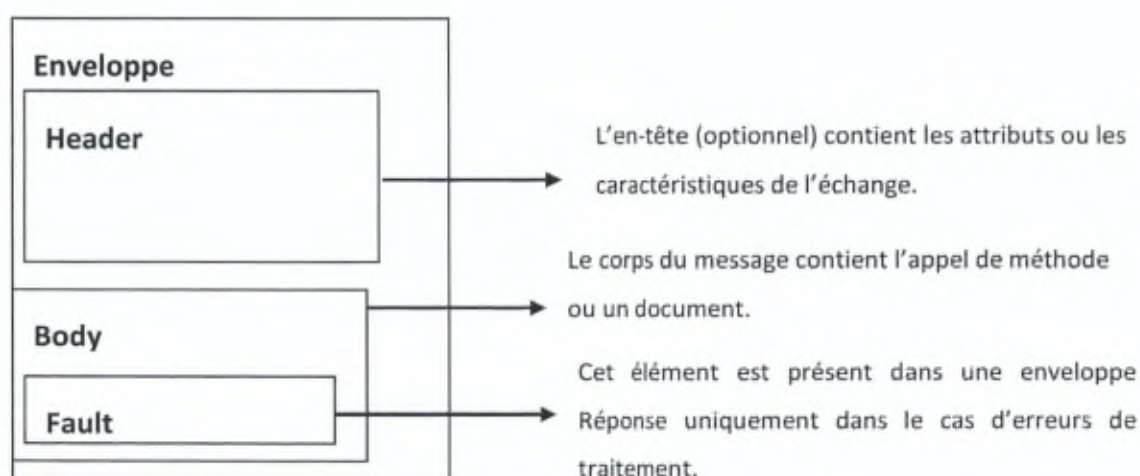


Figure D.1 : structure de l'enveloppe SOAP, inspiré de (W3C, 2002).



Il est aussi par ailleurs possible de représenter sous forme d'un diagramme UML, la structure d'une enveloppe SOAP.

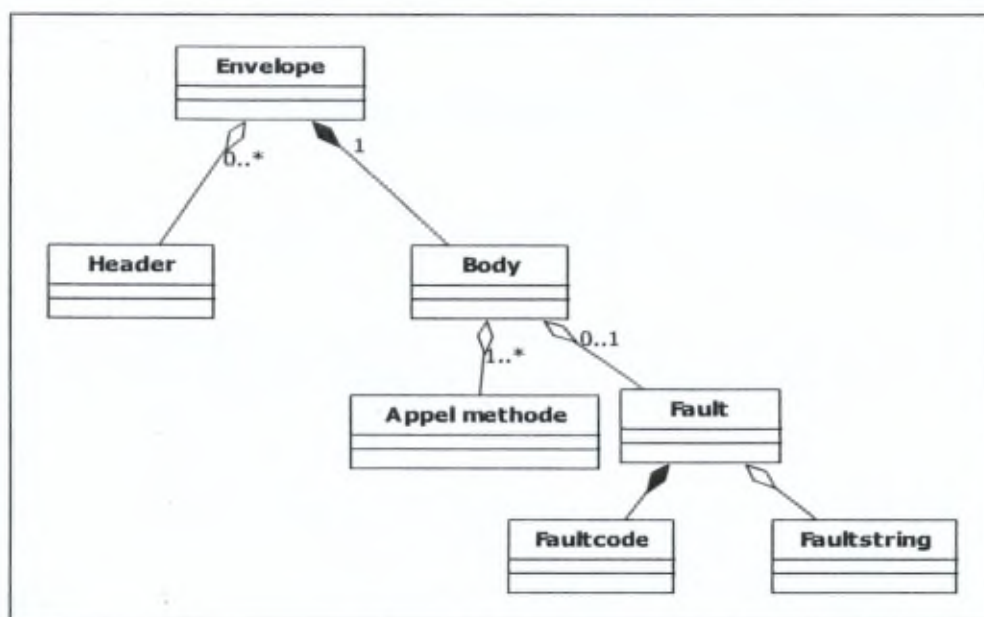


Figure D.2 : diagrammes des classes représentant la structure de l'enveloppe SOAP, inspiré de (Tahé, 2009).

Notons que le corps (*Body*) de l'enveloppe SOAP est un ensemble d'éléments XML. Il n'existe pratiquement aucune restriction à son contenu. Les formats des données quant à eux devront respecter le modèle d'encodage. Les règles de typage des données s'appuient sur les spécifications *XML Schema* et utilisent un grand nombre de types de données. Dans une optique exclusivement d'appel à procédure à distance des messages RPC (*Remote Procedure Call*), le corps de l'enveloppe ne contient typiquement que deux types d'informations :

1. Les appels de procédures distantes. Informations nécessaires à RPC.
2. Les réponses des procédures appelées ou les informations d'erreurs (élément *fault*).

Sans entrer dans le vif du sujet précisons ici les modèles d'utilisation de SOAP :

- Le modèle *RPC* qui utilise SOAP comme protocole d'invocation de méthodes à distances. Dans ce modèle, le contenu du corps de l'enveloppe devra se conformer aux conventions RPC, il s'agit du modèle d'utilisation de SOAP le plus couramment rencontré ;
- Le modèle *Document* consiste à utiliser SOAP pour véhiculer des données non-normalisées en XML. Seul le destinataire et l'émetteur peuvent interpréter le contenu du message.

#### D.1.3.4 Convention d'appel des procédures à distance (RPC)

La spécification SOAP définit la méthode d'encapsulation des appels de procédures distantes. Parmi les modèles d'utilisation de SOAP si l'on considère le modèle RPC, cette convention est le fondement même de SOAP. C'est en effet le moyen de standardiser les invocations de méthodes entre les différents partenaires d'un service Web. Selon la spécification, les éléments du message SOAP requis pour RPC sont les suivants :

- l'URI du nœud SOAP destinataire ;
- le nom de la méthode invoquée ;
- la signature de la méthode invoquée (optionnel) ;
- les paramètres de la méthode invoquée. Dans l'ordre et le typage requis ;
- un en-tête (facultatif) qui peut contenir des informations de sécurité, de transaction, etc.

En outre, l'appel de la méthode doit être décrit dans le corps du message selon la convention de définition d'une structure (struct). L'invocation de la méthode distante comportera un accesseur pour chaque paramètre. Cette structure doit avoir le même nom que la méthode invoquée.



## D.2 La couche de description : WSDL

### D.2.1 L'élément "definitions"

Il est l'élément racine du document WSDL, il permet de définir le nom du service Web et, surtout, les espaces de nommage XML associés. Notamment, l'attribut "targetNamespace" qui est une convention *XML Schema* qui permet au document de faire référence à lui-même. De plus, un espace de nommage par défaut est spécifié. Tous les éléments fils de <definitions> n'ayant pas d'espaces de nommage spécifique se référeront à l'espace de nommage par défaut.

```
<definitions name="forumMoodlePost"
targetnamespace=http://localhost/moodle/forumMoodlePost.wsdl
xmlns=http://schemas.xmlsoap.org/wsdl/ >
```

Figure D.3 : exemple d'un élément <definitions>

### D.2.2 L'élément "message"

En abstraction si l'on entend qu'un service Web consiste en un document XML émis ou reçu en communication avec une application distante, il devient important de définir clairement et sans ambiguïté les types de données échangées. Ces données échangées constituent dans leur ensemble le message. En fait, WSDL étant conçu sur base de la grammaire *XML Schema*, le typage des données respecte donc sa spécification (scalaires, tableaux et structures). Tout élément "message" possède un et un seul enfant "part" qui à son tour possède deux attributs "name" et "type" voir exemples des figures D.4 et D.5.

```
<types>
  <schema targetNamespace="rechercheMessage-xsd"
xmlns=http://www.w3c.org/2001/XMLSchema>
    <complexType name="Dokeos_forum_posts">
      <all>
        <element name="userid" type="xsd:int"/>
        <element name="title" type="xsd:string"/>
        <element name="content" type="xsd:string"/>
      </all>
    </complexType>
  </schema>
</types>
```

Figure D.4 : exemple d'un message partant d'une structure WSDL "Dokeos\_forum\_posts".

Et un message de recherche de l'identifiant de l'utilisateur qui poste une nouvelle sur un forum peut avoir la forme :

```
<message name="rechercheMessage">
  <part name="userid" type="xsd:int"/>
</message>
```

Figure D.5 : définition de l'élément message

### D.2.3 L'élément "portType"

L'élément `<portType>` est un container d'éléments `<opération>`. Ce concept est ici analogue à une classe Java, un objet IDL (CORBA) ou une librairie en .NET. En effet, si une opération est similaire à une méthode d'un objet et que les messages sont considérés comme ses arguments, alors l'élément `<portType>` peut être comparé à la définition d'un objet contenant de multiples méthodes (Newcomer, 2002). L'élément `<portType>` peut en outre être rapproché d'une interface Java car il ne définit que des méthodes appelées. En outre WSDL définit quatre modèles d'opérations pour un `<portType>`, il s'agit des modèles One-Way, Request-response, Solicit-Response, notification.

```
<portType name="rechercheMessage_PortType">
  <operation name="trouveMessage">
    <input message="tns:rechercheMessage"/>
    <output message="tns:rechercheMessageReponse"/>
  </operation>
</portType>
```

Figure D.6 : Code de définition d'un portType pour rechercher un message posté dans un forum.

### D.2.4 Élément "binding"

Conceptuellement il s'agit ici de « relier » les opérations définies au protocole de transport qui se charge de les véhiculer ou encore l'élément `<binding>` permet de spécifier la façon dont les opérations déclarées dans `<portType>` seront transportées sur le réseau. La spécification WSDL prévoit actuellement une liaison avec de multiples protocoles de transport tels que HTTP GET, HTTP POST ou SOAP. Il est à noter qu'un `<portType>` peut être



associé à plusieurs éléments `<binding>`, et c'est par son attribut `type` que ce dernier permet de définir le `<portType>` associé.

```
<binding name="rechercheMessage_Binding"
type="rechercheMessage_PortType">
  <soap:binding style="rpc"
transport="http://schema.xmlsoap.org/soap/http/">
    <operation name="trouveMessage">
      <soap:operation soapAction="trouveMessage"/>
      <input>
        <soap:body
encodingStyle="http://schema.xmlsoap.org/soap/encoding/"
        </input>
      <output>
        <soap:body
encodingStyle="http://schema.xmlsoap.org/soap/encoding/"
        </output>
      </operation>
    </binding>
```

Figure D.7 : Illustration d'un moyen d'associer l'appel du service d'un message du forum au protocole SOAP.

Après le *binding* il nous faut rechercher le service à appeler, en retrouvant sa localisation via un indicateur c'est le rôle de l'élément `<service>`.

## D.2.5 L'élément "service"

Cet élément sert à la localisation du service de manière simple par une adresse IP, une URI (ou URL) de la méthode à appeler. Par conséquent, chaque élément `<binding>` doit être associé à un élément `<service>`. L'élément `<service>` a comme fils l'élément `<port>`, et son occurrence est illimitée au sein de l'élément `<service>` ceci en vue de permettre un découplage entre une abstraction du service Web (éléments `<message>` et `<portType>`). Les protocoles de transports utilisés (élément `<binding>`). Ainsi à titre d'exemple dans le cas où un élément `<binding>` définissant une transaction SOAP synchrone va spécifier l'envoi en modèle *RPC* vers une adresse. Un autre élément `<binding>` définira la même transaction mais en modèle *Document* asynchrone qui sera émise vers une autre adresse.

## D.3 Couche de découverte des services web

Un service Web pour être découvert demande de recourir à une fonction de découverte dynamique nommée UDDI (*Universal Description, Discovery and Integration*).

UDDI est donc un annuaire des services Web rendus disponibles par quelques organisations et entreprises. Il se révèle lui-même comme étant un service Web et se veut d'une dimension universelle, sa version actuelle est UDDI 3.x depuis 2004. Cependant nous n'en avons pas eu besoin pour notre proposition d'implémentation.

## D.4 Codes de l'implémentation du service

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;

/**
 *
 * @author Blaise FYAMA
 */
public class moodleForumPost {

    public static void main(String[] arg){
        Connection conn = null;
        Statement st = null;
        Statement stmt = null;
        ResultSet r = null;
        try {
            conn =
DriverManager.getConnection("jdbc:mysql:localhost/phpmyadmin/moodle"
,"root","");

            st = conn.createStatement();

            r = st.executeQuery("SELECT discussion, parent, userid, created,
modified, mailed, subject, message, format, attachment, totalscore,
mailnow FROM mdl_forum_posts");

            while (r.next()) {
                // imprime les éléments du tuple
                int ladiscussion = r.getInt("discussion");
                int leparent = r.getInt("parent");
                int luserid = r.getInt("userid");
                int cree = r.getInt("created");
                int modification = r.getInt("modified");
                int envoye = r.getInt("mailed");
                String lesujet = r.getString("subject");
                String lemessage = r.getString("message");
                int leformat = r.getInt("format");
                String piecejointe = r.getString("attachment");
                int lescore = r.getInt("totalscore");
                int ecriremaintenant = r.getInt("mailnow");

                System.out.println ("Sujet: " + lesujet + " message: " +
lemessage);
            }
        }
    }
}
```



```

    }
    catch (SQLException ex){
    // handle any errors
    System.out.println("SQLException: " + ex.getMessage());
    System.out.println("SQLState: " + ex.getSQLState());
    System.out.println("VendorError: " + ex.getErrorCode());
    }
}
}

```

Figure D.8 : Code de la classe java du client.

## D.5 Codes de description WSDL du service

```

<wsdl:definitions xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://127.1.1.0:8080/axis/moodleForumPost.jws"
xmlns:intf="http://127.1.1.0:8080/axis/moodleForumPost.jws"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://127.1.1.0:8080/axis/moodleForumPost.jws">
<!--
WSDL created by Apache Axis version: 1.4 Built on Apr 22, 2006
(06:55:48 PDT)
-->
<wsdl:types>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://127.1.1.0:8080/axis/moodleForumPost.jws">
<import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
<complexType name="ArrayOf_xsd_string">
<complexContent>
<restriction base="soapenc:Array">
<attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string[]"/>
</restriction>
</complexContent>
</complexType>
</schema>
</wsdl:types>
<wsdl:message name="mainRequest">
<wsdl:part name="arg" type="impl:ArrayOf_xsd_string"/>
</wsdl:message>
<wsdl:message name="mainResponse"></wsdl:message>

```

```

<wsdl:portType name="moodleForumPost">
  <wsdl:operation name="main" parameterOrder="arg">
    <wsdl:input message="impl:mainRequest" name="mainRequest"/>
    <wsdl:output message="impl:mainResponse" name="mainResponse"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="moodleForumPostSoapBinding"
  type="impl:moodleForumPost">
  <wsdlsoap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="main">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="mainRequest">
      <wsdlsoap:body
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://DefaultNamespace" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="mainResponse">
      <wsdlsoap:body
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://127.1.1.0:8080/axis/moodleForumPost.jws"
        use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="moodleForumPostService">
  <wsdl:port binding="impl:moodleForumPostSoapBinding"
    name="moodleForumPost">
    <wsdlsoap:address
      location="http://127.1.1.0:8080/axis/moodleForumPost.jws"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Figure D.9 : code de description WSDL du service moodleForumPost