



HAL
open science

Approche d'assistance aux auteurs pour la réutilisation d'objets d'apprentissage

Ramzi Farhat

► **To cite this version:**

Ramzi Farhat. Approche d'assistance aux auteurs pour la réutilisation d'objets d'apprentissage. Autre [cs.OH]. Institut National des Télécommunications; Université de la Manouba (Tunisie), 2010. Français. NNT : 2010TELE0034 . tel-00589601

HAL Id: tel-00589601

<https://theses.hal.science/tel-00589601v1>

Submitted on 29 Apr 2011

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

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



Telecom SudParis



Université d'Evry Val
d'Essonne



Ecole Nationale des Sciences de
l'Informatique

**Thèse de doctorat de Telecom & Management SudParis dans le cadre de l'école
doctorale SITEVRY en co-accréditation avec l'Université d'Evry Val d'Essonne et en
co-tutelle avec l'Ecole Nationale des Sciences de l'Informatique**

Spécialité informatique

Par :

Ramzi FARHAT

**Thèse présentée pour l'obtention du grade de Docteur de Telecom &
Management SudParis et du grade de Docteur de l'Ecole Nationale
des Sciences de l'Informatique**

Approche d'assistance aux auteurs pour la réutilisation d'objets d'apprentissage

Soutenue le 17/12/2010 devant le jury composé de :

Madame Anne Doucet
Madame Amel Bouzeghoub
Madame Yolaine Bourda
Monsieur Serge Garlatti
Monsieur Bruno Defude
Monsieur Mohamed Jemni

Professeur, UPMC-LIP6, Présidente
Professeur, Télécom SudParis, Examinatrice
Professeur, Supélec, Rapporteur
Professeur, Télécom Bretagne, Rapporteur
Professeur, Télécom SudParis, Co-directeur de thèse
Professeur, Université de Tunis, Co-directeur de thèse

Remerciements

Ce mémoire de thèse représente l'achèvement de plusieurs années de travail et qui certes n'aurait pu voir le jour sans la participation, l'aide, les conseils, le soutien et l'encouragement de nombreuses personnes.

Je tiens tout d'abord à remercier mes co-directeurs de thèse Mr Bruno DEFUDE et Mr Mohamed JEMNI qui ont accepté de superviser mon travail de thèse au long de ces années. Leurs qualités scientifiques et humaines ont joués un rôle crucial pour que je puisse avancer dans mon travail de thèse et pour que je rédige ce mémoire. Je leur serai toujours reconnaissant pour toute l'aide qu'ils m'ont apportée.

Mes remerciements s'adressent aux membres de jury :

Mme Yolaine BOURDA et Mr Serge GARLATTI qui ont accepté de rapporter ce mémoire de thèse. Leurs remarques, suggestions et commentaires m'ont été très précieux et ont permis d'améliorer ce manuscrit.

Mme Anne DOUCET et Mme Amel BOUZEGHOUB m'ont fait honneur en acceptant de faire partie de mon jury de thèse. Leur participation dans l'évaluation de ce travail est d'une très grande valeur.

Je voudrais adresser un remerciement particulier à tous les membres de l'unité de recherche UTIC à l'École Supérieure des Sciences et Techniques de Tunis. L'ambiance au sein de l'unité de recherche, la complicité et la serviabilité de mes amis m'ont permis de travailler en des bonnes conditions. S'il m'est difficile de ne citer que quelques noms, je ne peux pas ne pas citer Oussama ELGOUL, Fathi ESSALMI et Mohsen LAABIDI pour leur soutien et pour leurs encouragements.

Je voudrais également adresse mes vifs remerciements pour les nombreuses personnes que j'ai croisées lors de mes séjours à l'Institut Télécom SudParis et qui m'ont apporté beaucoup d'aide et des moments inoubliables. S'il est également difficile de ne citer que quelques noms, je tiens à remercier en particulier Samir TATA qui a su m'apporter l'aide nécessaire aux moments difficiles, à mes chers amis Nomane OULD AHMED M'BAREK, Mahmoud GHORBEL, Mohammed SELAMI, Jérôme SICARD et Yassin CHABEB.

Je remercie les membres du département informatique de l'Institut Télécom SudParis pour leur accueil très convivial pendant les multiples stages que j'ai passés avec eux. J'adresse un remerciement particulier à Brigitte HOUASSINE, secrétaire du département,

qui a allié sympathie et compétence. Elle a su toujours me simplifier les procédures administratives liées à mes doubles inscriptions dans le cadre de la cotutelle.

Je remercie également tous mes collègues à l'Ecole Supérieure des Sciences et Techniques de Tunis, où j'ai débuté ma carrière dans l'enseignement supérieur, ainsi que mes collègues à l'Institut Supérieur des Langues Appliquées et d'Informatique de Béja, où j'exerce actuellement mes fonctions en tant qu'assistant d'enseignement supérieur, pour l'ambiance sereine et amicale qu'ils m'ont fait sentir.

Je tiens à remercier ma famille pour son soutien, sa compréhension, ses encouragements et son aide. Mes parents, Khemaïes et Fawzia, tout d'abord qui ont eu confiance en mes choix et qui m'ont toujours poussé vers l'avant. Tout ce que j'ai pu réaliser est certes grâce à eux. Ma femme Fatma qui a su comment m'aider, m'encourager et me soutenir avec plein d'amour et de délicatesse pendant les moments difficiles de stress et de fatigue durant la rédaction de ce mémoire. Je tiens à remercier également mes frères Anis et Haïthem et ma sœur Imen.

Table des matières

INTRODUCTION GENERALE	1
1. CONTEXTE GENERAL : LA FORMATION EN LIGNE	1
2. PROBLEME POSE : CONCEPTION D'OBJETS D'APPRENTISSAGE COMPOSES.....	2
3. APPROCHE PROPOSEE : ASSISTANCE AUX AUTEURS POUR LA REUTILISATION	3
4. ORGANISATION DU RAPPORT	5
OBJET D'APPRENTISSAGE	7
1. INTRODUCTION.....	7
2. DEFINITIONS ET PROPRIETES	7
2.1. Définitions	7
2.2. Propriétés	8
2.3. Synthèse	10
3. METADONNEES EDUCATIVES.....	11
3.1. Métadonnées LOM	11
3.2. Profils d'application LOM	13
4. INTEROPERABILITE	14
4.1. Normes, standards et spécifications	15
4.2. SCORM.....	15
5. REUTILISABILITE	16
5.1. Granularité.....	17
5.2. Assemblage.....	18
6. CYCLE DE VIE D'OBJETS D'APPRENTISSAGE.....	20
6.1. Modèles de cycle de vie	20
6.2. Modèle de Rensing.....	20
6.3. Modèle de Strijker	21
6.4. Synthèse	21
6.4.1. Phase d'utilisation.....	22
6.4.2. Phase de publication	23
6.4.3. Phase de conception.....	23
7. DEFIS DE LA COMPOSITION DE NOUVEAUX OBJETS D'APPRENTISSAGE PAR REUTILISATION.....	24
7.1. Défis de la composition par réutilisation	24
7.2. Etat de l'art de la composition par réutilisation et problématiques ouvertes	26
8. CONCLUSION	26
COMPOSITION ET PERSONNALISATION.....	29
1. INTRODUCTION.....	29
2. COMPOSITION ET PERSONNALISATION DE DOCUMENTS NUMERIQUES.....	29
2.1. Documents virtuels adaptatifs	30
2.1.1. Définitions et principes.....	30

2.1.2. Composition et personnalisation.....	30
2.2. <i>Hypermédiats adaptatifs</i>	31
2.2.1. Définitions et principes.....	32
2.2.2. Composition et adaptation.....	32
2.3. <i>Synthèse</i>	34
3. APPROCHE SIMBAD.....	35
3.1. <i>Vue d'ensemble</i>	35
3.2. <i>Modèle de domaine</i>	37
3.3. <i>Modèle de composition</i>	38
3.4. <i>Modèle de l'apprenant</i>	39
4. OBJETS D'APPRENTISSAGE DANS SIMBAD.....	40
4.1. <i>Modèle d'objet d'apprentissage</i>	40
4.2. <i>Graphes de composition</i>	42
5. PROBLEMATIQUES.....	45
5.1. <i>Conception d'objets d'apprentissage composés</i>	45
5.2. <i>Assistance et guidage</i>	46
5.3. <i>Vérification de la cohérence</i>	46
5.4. <i>Métadonnées</i>	47
5.5. <i>Etude de conformité</i>	47
6. CONCLUSION.....	47
VUE D'ENSEMBLE DE NOTRE APPROCHE D'ASSISTANCE.....	49
1. INTRODUCTION.....	49
2. VUE D'ENSEMBLE.....	49
2.1. <i>Présentation</i>	49
2.2. <i>Déroulement de l'assistance</i>	51
3. PRESENTATION DES PHASES RECHERCHE ET CALCUL DE METADONNEES.....	52
3.1. <i>Assistance à la recherche</i>	52
3.2. <i>Calcul de métadonnées</i>	53
4. TRANSFORMATION D'OBJETS SCORM EN OBJETS SIMBAD.....	56
4.1. <i>Objectif de la transformation</i>	56
4.2. <i>Modèle d'agrégation de contenu</i>	57
4.3. <i>Algorithme de transformation</i>	59
4.4. <i>Exemples de transformation</i>	60
5. CONCLUSION.....	62
VERIFICATION DE LA COHERENCE.....	63
1. INTRODUCTION.....	63
2. VERIFICATION DE LA COHERENCE STRUCTURELLE.....	63
2.1. <i>Règles de cohérence structurelle</i>	63
2.2. <i>Algorithme de vérification de la première règle</i>	67

2.3. Algorithme de vérification de la deuxième et troisième règle.....	68
2.4. Algorithme de vérification de la quatrième règle.....	68
2.5. Algorithme de vérification de la cinquième règle.....	68
3. VERIFICATION DE LA COHERENCE SEMANTIQUE.....	70
3.1. Présentation.....	70
3.2. Satisfaction des pré-requis.....	70
3.2.1. Principe de satisfaction des pré-requis.....	70
3.2.2. Exemple.....	71
3.2.3. Algorithme de vérification de la satisfaction des pré-requis.....	72
3.3. Cohérence sémantique du contenu.....	73
3.3.1. Principe de la cohérence sémantique du contenu.....	73
3.3.2. Exemple.....	74
3.3.3. Algorithme de vérification de la cohérence sémantique du contenu.....	75
4. CONCLUSION.....	75
ANALYSE DE L'OBJET D'APPRENTISSAGE.....	77
1. INTRODUCTION.....	77
2. METRIQUES.....	77
2.1. Métriques Structurelles.....	78
2.2. Métriques Sémantiques.....	78
2.3. Métadonnées.....	79
2.4. Niveaux d'abstraction.....	80
2.5. Synthèse.....	80
3. INDICATEURS.....	81
3.1. Concordance structurelle.....	82
3.2. Capacité à être adapté.....	84
3.3. Capacité à être réutilisé.....	84
3.4. Consistance didactique.....	85
3.5. Exemple illustratif.....	86
4. ANALYSE DE L'ESPACE SEMANTIQUE.....	88
4.1. Notion d'espace sémantique.....	88
4.2. Calcul de l'espace sémantique.....	90
4.3. Composante sémantique.....	91
4.4. Complexité sémantique du contenu.....	92
4.5. Continuité sémantique du contenu.....	94
4.6. Déséquilibre sémantique du contenu.....	96
5. APPORTS DE L'ANALYSE.....	98
6. CONCLUSION.....	99
ANALYSE DE LA CONFORMITE.....	101
1. INTRODUCTION.....	101
2. PRINCIPES DE LA VERIFICATION DE LA CONFORMITE.....	101

2.1. Introduction	101
2.2. Règles de conformité	102
2.3. Notation graphique pour les règles	103
2.4. Vérification automatique des règles	104
3. REGLES DE CONFORMITE	104
3.1. Règles pour les métadonnées.....	104
3.1.1. Catégories LOM obligatoires.....	104
3.1.2. Eléments LOM obligatoires.....	105
3.1.3. Règles quantitatives	105
3.2. Règles structurelles.....	105
3.2.1. Forme du graphe.....	105
3.2.2. Dimensions et densité du graphe de composition	106
3.2.3. Niveaux d'abstraction.....	107
3.2.4. Similarité des graphes affichables	108
3.3. Règles sémantiques.....	108
3.3.1. Contenu	109
3.3.2. Pré-requis.....	109
3.3.3. Fonction d'acquisition	110
3.4. Règles hybrides.....	110
3.4.1. Rôles du sommet racine et des sommets feuilles	111
3.4.2. Patron de conception	111
4. LANGAGE FORMEL POUR LA DESCRIPTION DES REGLES	111
4.1. Eléments du langage.....	111
4.2. Fonctions	113
4.2.1. Fonctions pour les règles de métadonnées.....	113
4.2.2. Fonctions pour les règles structurelles	114
4.2.3. Fonctions pour les règles sémantiques.....	116
4.2.4. Formulation des règles hybrides	117
4.3. Expressivité du langage.....	118
5. NOTATION GRAPHIQUE	120
5.1. Objectif.....	120
5.2. Conteneur pour les règles.....	120
5.3. Patrons de conception	121
5.4. Expressivité de la notation graphique	123
6. EXEMPLE ILLUSTRATIF	123
6.1. Règles en langue naturelle.....	124
6.2. Règles en langage formel.....	125
6.3. Notation graphique pour les règles	126
6.4. Vérification automatique des règles	126
7. CONCLUSION	128
MODELISATION ET PROTOTYPE.....	129

1. INTRODUCTION.....	129
2. SYSTEME COLORS.....	129
2.1. Contexte.....	129
2.2. Présentation.....	131
2.3. Coordinateur de banques COLORS.....	132
2.4. Assistance aux auteurs via COLORS.....	134
2.5. Extension de COLORS aux objets SIMBAD.....	135
3. LOGIQUE DES FRAMES.....	136
3.1. Introduction.....	136
3.2. Présentation de la F-logique.....	137
3.3. Langage F-logique.....	137
4. MODELISATION DES OBJETS D' APPRENTISSAGE SIMBAD EN F-LOGIQUE.....	138
4.1. Description conceptuelle.....	138
4.2. Objet d' apprentissage.....	139
4.3. Graphe de composition.....	141
4.4. Exemple illustratif.....	142
5. ANALYSE DES OBJETS SIMBAD.....	143
5.1. Architecture.....	143
5.2. Implémentation.....	145
6. VERIFICATION DE LA COHERENCE.....	147
6.1. Expérimentation.....	147
6.2. Test et résultats.....	147
7. ASSISTANCE A LA COMPREHENSION.....	149
7.1. Analyse générale.....	149
7.2. Tests et résultats.....	149
7.3. Analyse spécifique de la sémantique du contenu.....	152
7.4. Tests et résultats.....	153
8. VERIFICATION DE LA CONFORMITE.....	154
8.1. Expérimentation.....	154
8.2. Tests et résultats.....	156
9. CONCLUSION.....	157
CONCLUSION GENERALE.....	159
1. BILAN DES TRAVAUX.....	159
2. RESULTATS OBTENUS.....	160
3. PERSPECTIVES.....	161
ANNEXE A : EXTRAIT DE LA BASE DE CONNAISSANCES.....	179
ANNEXE B : PREDICATS DE HAUT NIVEAU.....	185
ANNEXE C : SYSTEME COLORS.....	189

Table des figures

Figure 1 : Propriétés caractéristiques d'un objet d'apprentissage	11
Figure 2 : Organisation du schéma de métadonnées LOM [Passardière et Jarraud, 2004].....	13
Figure 3 : Les quatre documents de référence de la spécification SCORM	16
Figure 4 : Niveaux de granularité selon LOM et SCORM.....	18
Figure 5 : Modèle d'agrégation de contenu [ADL, 2004b].....	19
Figure 6 : Modèle de cycle de vie d'un objet d'apprentissage [Rensing et al., 2005].....	21
Figure 7 : Cycle de vie d'un objet d'apprentissage [Farhat et al., 2007b]	22
Figure 8 : Méthodes d'adaptation	34
Figure 9 : Niveaux de modélisation de l'approche SIMBAD.....	36
Figure 10 : Extrait d'un modèle de domaine	37
Figure 11 : Graphe de composition d'un objet d'apprentissage composé	38
Figure 12 : Vue globale d'un objet d'apprentissage	40
Figure 13 : Différents points de vue d'un graphe de composition	43
Figure 14 : Vue d'ensemble de notre approche d'assistance	50
Figure 15 : Diagramme d'activité relatif à l'approche d'assistance.....	51
Figure 16 : Vue d'ensemble d'un paquetage SCORM.....	58
Figure 17 : Exemple de transformation d'un contenu SCORM simple en SIMBAD.....	61
Figure 18 : Exemple de transformation d'un contenu SCORM complexe en SIMBAD.....	61
Figure 19 : Transformation d'un contenu SCORM en SIMBAD	62
Figure 20 : Exemple de violation de la première règle de cohérence SIMBAD	64
Figure 21 : Exemple de violation de la deuxième règle de cohérence SIMBAD.....	64
Figure 22 : Exemple de violation de la troisième règle de cohérence SIMBAD.....	65
Figure 23 : Exemple de violation de la quatrième règle de cohérence SIMBAD	65
Figure 24 : Exemple de violation de la cinquième règle de cohérence SIMBAD	66
Figure 25 : Exemple de violation de la sixième règle de cohérence SIMBAD.....	67
Figure 26 : Graphe de composition abstrait de l'objet d'apprentissage composé lo_903	71
Figure 27 : Graphe de composition abstrait de l'objet d'apprentissage composé lo_904	74
Figure 28 : Génération des métriques relatives aux objets d'apprentissage composés.....	82
Figure 29 : Structure de l'objet d'apprentissage lo_400.....	86
Figure 30 : Vue en mode radar des résultats d'analyse d'un objet d'apprentissage composé.....	87
Figure 31 : Modèle de domaine (simplifié).....	89
Figure 32 : Représentation graphique d'un espace sémantique	90
Figure 33 : Composantes sémantiques d'un espace sémantique	92
Figure 34 : Etiquetage des composantes sémantiques par les objets d'apprentissage.....	96
Figure 35 : Différents formats de description des règles de conformité.....	102
Figure 36 : Conteneur utilisé dans la notation graphique.....	121
Figure 37 : Exemple de notation graphique pour les règles de conformité.....	122

<i>Figure 38 : Notation graphique de règles de conformité.....</i>	<i>126</i>
<i>Figure 39 : Architecture logicielle du système COLORS.....</i>	<i>132</i>
<i>Figure 40 : Gestion des requêtes au niveau d'un réseau de banques COLORS.....</i>	<i>133</i>
<i>Figure 41 : Interface de gestion des banques d'un réseau COLORS par le coordinateur</i>	<i>134</i>
<i>Figure 42 : Interfaces COLORS relatives à l'analyse des propriétés des objets d'apprentissage</i>	<i>135</i>
<i>Figure 43 : Architecture de l'extension de COLORS.....</i>	<i>136</i>
<i>Figure 44 : Différents éléments relatifs aux objets d'apprentissage SIMBAD</i>	<i>138</i>
<i>Figure 45 : Exemple d'un graphe de composition abstrait.....</i>	<i>142</i>
<i>Figure 46 : Architecture expérimentale</i>	<i>143</i>
<i>Figure 47 : Console d'administration OntoBroker 5.0.....</i>	<i>144</i>
<i>Figure 48 : Diagramme de classe du prototype implémenté.....</i>	<i>146</i>
<i>Figure 49 : Vérification de la conformité structurelle de l'objet LO_902</i>	<i>147</i>
<i>Figure 50 : Vérification de la conformité structurelle de l'objet LO_1007</i>	<i>148</i>

INTRODUCTION GENERALE

1. Contexte général : la formation en ligne

L'e-learning ou la formation en ligne est une variante de la formation à distance qui se caractérise par l'utilisation de l'Internet comme média de diffusion. Plusieurs tentatives de définition existent telles que celle de l'Union Européenne : « l'e-learning est l'utilisation des nouvelles technologies multimédias de l'Internet pour améliorer la qualité de l'apprentissage en facilitant d'une part l'accès à des ressources et à des services, d'autre part les échanges et la collaboration à distance ». Ce mode de formation suscite de nos jours beaucoup d'intérêt dont témoignent les normes, les standards et les spécifications liés à la formation en ligne, le nombre d'acteurs impliqués, l'abondance des outils et des plateformes associés, ainsi que les efforts de recherche scientifique en relation. Grâce à ce mode de formation certaines barrières ont été surmontées telles que celles liées au temps ou à l'espace. Si ces deux aspects sont spécifiques à la formation à distance d'une façon générale, d'autres sont intrinsèques à la formation en ligne. En effet, avec l'utilisation des technologies de l'information et de la communication au profit de la formation un énorme potentiel a été mis à disposition permettant de rendre la formation plus attractive, effective et efficace. Dans ce contexte plusieurs travaux ont été menés tels que ceux concernant l'individualisation de la formation en ligne.

En effet, l'individualisation de la formation en ligne est certes l'un des aspects les plus prometteurs mais aussi les plus difficiles à mettre en œuvre en pratique. Les approches associées permettent la personnalisation et l'adaptation du contenu de la formation pour répondre mieux aux attentes de l'apprenant. L'idée principale est de capturer dans un premier temps certains paramètres liés à l'apprenant. Ceux-ci peuvent varier selon l'approche des simples préférences (en terme de nature du média, de langue, etc.) à la prise en compte des styles d'apprentissage (Sensitif/Intuitif, Visuel/Verbal, etc.). Ces informations doivent être conservées dans un modèle d'apprenant adéquat. Dans un deuxième temps il faut prévoir un modèle de contenu supportant l'adaptation. Ceci doit se traduire au moins à deux niveaux. Le premier correspond aux métadonnées éducatives qui doivent permettre la découverte d'un contenu donné au sein d'un corpus tel qu'il soit adéquat par rapport au profil de l'apprenant et à des objectifs pédagogiques. Le deuxième correspond à la structure interne du contenu qui

doit pouvoir supporter l'adaptation nécessaire dans le contexte d'individualisation de la formation.

Disposant d'un modèle de l'apprenant et d'un modèle de contenu, une stratégie de gestion de la formation en ligne individualisée doit être définie. Celle-ci doit pouvoir offrir les moyens nécessaires pour la gestion du profil de l'apprenant tout au long de la formation, en particulier sa mise-à-jour. En plus, elle doit offrir à l'apprenant les moyens nécessaires pour interagir avec les ressources éducatives afin d'exprimer ses besoins pédagogiques et d'accéder aux ressources disponibles. Cette stratégie doit également supporter une approche d'adaptation du contenu fourni à l'apprenant en se basant d'une part sur son profil et d'autre part sur les possibilités offertes par le modèle de contenu.

2. Problème posé : Conception d'objets d'apprentissage composés

Le contenu éducatif joue le plus souvent, si ce n'est pas toujours, un rôle essentiel dans tout processus de formation et d'apprentissage. Par conséquent, la communauté de la formation en ligne lui a accordé une grande importance qui s'est manifestée entre autres en termes de spécifications, de standards et de normes. Plusieurs aspects ont été étudiés dont l'un des aboutissements est la notion d'objet d'apprentissage. En dépit de la diversité des définitions de la notion objet d'apprentissage, qui ne sont pas forcément convergentes, un consensus est établi sur les caractéristiques minimales qu'il doit avoir, à savoir : l'accessibilité, l'interopérabilité et la réutilisabilité.

L'accessibilité est la capacité de trouver et identifier un objet donné dans un grand corpus d'objets d'apprentissage. Ce sont les métadonnées éducatives qui sont utilisées pour assurer cette caractéristique d'accessibilité. Ainsi, il est possible de chercher et de s'informer sur le contenu éducatif d'un objet d'apprentissage sans avoir besoin de le consulter réellement. Une métaphore souvent utilisée est celle du livre et de la couverture du livre où le livre représente les données (dans notre cas un objet d'apprentissage) et la couverture les métadonnées (dans notre cas les métadonnées éducatives).

L'interopérabilité, dans le cas des objets d'apprentissage, consiste à assurer l'indépendance vis-à-vis des plateformes et des outils de gestion de contenu. L'intérêt est double. En effet, pour les producteurs de contenu cela doit entraîner une diminution des coûts notamment ceux relatifs aux efforts d'adaptation des objets aux outils. Pour l'utilisateur l'interopérabilité lui évite de se trouver limité à un seul éditeur et de se soucier du problème de compatibilité.

La réutilisabilité d'un objet d'apprentissage est sa capacité à être utilisé et réutilisé plusieurs fois dans des contextes différents et par des utilisateurs différents. Un objet est soit réutilisé par les apprenants dans différentes situations d'apprentissage, soit par les auteurs pour les assembler afin de concevoir de nouveaux objets d'apprentissage. C'est le deuxième cas qui nous intéresse vu que c'est le plus délicat à mettre en œuvre. Plusieurs études ont essayé de contribuer à cette problématique en étudiant en particulier des aspects relatifs à la granularité et à la neutralité pédagogique. En fait, il est souvent avancé que la réutilisation des objets d'apprentissage est favorisée par une granularité fine associée à une neutralité pédagogique.

Nous nous intéressons au cas où nous disposons d'objets d'apprentissage réutilisables et qu'un auteur souhaite les assembler pour produire un nouvel objet d'apprentissage que nous allons désigner par le terme « objet d'apprentissage composé ». Ce processus est le plus souvent introduit via la métaphore des « LEGOs ». En possédant des pièces de LEGO élémentaires ou combinées, il est toujours possible de les emboîter ensemble pour former de nouveaux objets. Toutefois, cette métaphore comme le confirme Willey [Willey 2002] est loin d'être fidèle à la réalité. Ainsi, il propose une autre métaphore, qui nous semble plus fidèle au cas des objets d'apprentissage composés, celle des molécules de chimie. En fait, il est possible d'assembler des atomes ou des molécules mais tout en respectant certaines règles et avec un nombre limité de combinaisons cohérentes. Cette métaphore souligne deux points importants relatifs à l'assemblage des objets d'apprentissage. Le premier est que cela nécessite des compétences (contrairement à la métaphore des LEGOs) et que c'est loin d'être intuitif. En effet, plusieurs règles doivent être respectées afin d'assurer la cohérence de l'assemblage.

La problématique à laquelle nous nous sommes intéressés peut être introduite par la question suivante : Comment peut-on assister les auteurs des objets d'apprentissage composés à surmonter les difficultés inhérentes de la réutilisation d'autres objets ? Cette question se décline en plusieurs sous questions relatives à la nature de l'approche à proposer et aux mécanismes à mettre en place pour l'implémenter.

3. Approche proposée : Assistance aux auteurs pour la réutilisation

Notre contribution consiste à définir une approche automatisée pour assister les auteurs des objets d'apprentissage composés. En fait, nous distinguons deux types d'objets d'apprentissage : les objets d'apprentissage simples (ou atomiques) et les objets d'apprentissage composés (ou complexes). Ces derniers sont conçus par assemblage d'autres objets d'apprentissage (simples ou composés). C'est à la conception de ces derniers que nous

nous intéressons. Notre approche comporte cinq phases, à savoir : l'assistance à la recherche des objets d'apprentissage, l'assistance à la composition, l'assistance à la compréhension, le calcul automatique de métadonnées éducatives et l'analyse de la conformité par rapport à un patron de conception.

Un environnement auteur en conformité avec notre approche doit fournir une assistance pour les auteurs pour concevoir un objet d'apprentissage composé selon une démarche itérative et incrémentale. Au départ l'environnement doit inviter l'auteur à préciser la sémantique de l'objet qu'il souhaite mettre en place en terme de contenu, de pré-requis et de fonction d'acquisition. Par exemple, l'auteur peut spécifier qu'il souhaite concevoir un chapitre sur le modèle relationnel pour un public qui connaît déjà la notion de bases de données et le modèle entité association. L'environnement doit le guider ensuite pour choisir parmi les objets disponibles ceux qui peuvent être intégrés dans son objet. Par exemple, l'environnement va lui proposer une liste comportant des exercices, des exemples, des descriptions, etc. qui portent sur l'algèbre relationnelle, les formes normales, etc.

L'environnement doit offrir ensuite une interface permettant de composer le nouvel objet en définissant sa structure et sa sémantique. L'auteur peut vérifier à tout moment la cohérence structurelle ou sémantique de l'objet composé. Par exemple, l'auteur doit être alerté au cas où le graphe de composition comporte un opérateur mal placé.

Une fois que l'objet a pris forme l'environnement procède à une analyse qui va permettre à l'auteur de comprendre les propriétés de l'objet conçu. Le système doit établir une cartographie qui résume ce qu'est l'objet au niveau structure et au niveau sémantique. L'intérêt de cette phase est de permettre à l'auteur de vérifier l'adéquation entre ce qu'il a eu l'intention de composer et ce qu'il a composé réellement. Ceci va lui permettre de mettre en question sa conception et il peut vouloir améliorer certains aspects. Il peut à ce moment revenir à la première ou à la deuxième phase pour effectuer les modifications adéquates. Par exemple, l'auteur peut découvrir au niveau de cette phase que son objet n'a pas la capacité à être adapté aux profils des apprenants.

Une fois que l'auteur a approuvé la composition, l'environnement auteur procède au calcul automatique de métadonnées éducatives. Cette étape se base sur la réutilisation des métadonnées éducatives des objets réutilisés. Le calcul automatique de métadonnées suit un certain nombre de règles en fonction principalement de la structure de l'objet. C'est à l'auteur d'approuver les métadonnées calculées automatiquement.

L'environnement permet ensuite de vérifier automatiquement la conformité de l'objet à des patrons de conception. Ceux-ci peuvent être définis par une ou plusieurs autorités et permettent de décrire des propriétés qui doivent être vérifiées par l'objet composé. Par exemple, un patron de conception peut refléter une approche pédagogique donnée.

Notre approche n'impose pas un ordre particulier pour appliquer les différentes phases d'assistance proposées. En fait, son apport est d'offrir les moyens nécessaires aux auteurs pour relever les défis relatifs à la composition par réutilisation des objets d'apprentissage.

4. Organisation du rapport

Ce présent rapport est organisé en sept chapitres. Dans le premier chapitre nous présentons l'état de l'art concernant le concept d'objet d'apprentissage. Nous élaborons par la suite une synthèse des propriétés caractéristiques d'un objet d'apprentissage afin d'en déduire les propriétés qualifiées de fondamentales. Par la suite nous allons introduire les défis relatifs à la composition par réutilisation d'objets d'apprentissage et poser la problématique générale.

Le chapitre suivant introduit l'approche SIMBAD qui représente d'une part le cadre théorique de nos recherches et d'autre part un cas de mise en œuvre pratique du concept d'objets d'apprentissage composés. Le chapitre est conclu par l'énoncé des problématiques détaillées auxquelles nous nous intéressent dans cette thèse.

Au niveau du troisième chapitre nous détaillons notre approche d'assistance aux auteurs d'objets d'apprentissage composés. Notre approche comporte cinq phases, à savoir : l'assistance à la recherche, l'assistance à la composition, l'assistance à la compréhension, le calcul automatique de métadonnées éducatives et la vérification de la conformité de l'objet composé à un patron de conception. Les phases recherche et calcul automatique de métadonnées sont introduites au niveau de ce chapitre.

Le quatrième chapitre est réservé à la vérification de la cohérence de l'objet que ce soit d'un point de vue structurel ou d'un point de vue sémantique par rapport au modèle sous-jacent qui est SIMBAD dans notre cas. Nous allons étudier et détailler les types d'incohérences qui peuvent apparaître lors de la composition. Puis, nous allons proposer les algorithmes nécessaires pour automatiser la vérification de la cohérence.

Le cinquième chapitre est consacré à l'assistance à la compréhension de l'objet d'apprentissage composé. Il est organisé en deux grandes parties. Dans la première partie nous allons introduire les métriques relatives à un objet d'apprentissage composé et nous

allons proposer des indicateurs métiers afin d'aider les auteurs à la compréhension de leurs objets composés. La deuxième partie du chapitre concerne la compréhension de la sémantique du contenu. C'est à ce niveau que nous allons introduire les notions d'espace sémantique et de composante sémantique. Ces notions seront ensuite utilisées pour en déduire de nouvelles métriques. Celles-ci seront utilisées à leur tour pour la définition d'indicateurs métiers sur la sémantique du contenu des objets composés.

Le sixième chapitre concerne la vérification de la conformité de l'objet d'apprentissage composé à un patron de conception. Nous allons ainsi commencer par étudier les règles de conformité selon leurs catégories : structurelles, sémantiques, d'annotation ou hybrides. Ensuite, nous allons proposer un langage formel pour la description de ces règles. Puis, nous allons introduire une notation graphique permettant de faciliter la communication de ces différentes règles aux auteurs des objets d'apprentissage.

Le septième et dernier chapitre couvre les aspects relatifs à l'expérimentation et la validation de notre approche. Tout d'abord nous allons introduire le système COLORS qui présente une banque d'objets d'apprentissage capable d'intégrer les différentes phases de notre approche. Ensuite, nous allons introduire le prototype d'expérimentation réalisé en se basant sur la F-logique et le moteur d'inférence OntoBroker. Les résultats des simulations sur des objets d'apprentissage particuliers seront présentés afin de comparer les résultats obtenus aux résultats attendus.

Chapitre I :

OBJET D'APPRENTISSAGE

1. Introduction

Ce chapitre est consacré à l'état de l'art concernant le concept d'objet d'apprentissage. La première partie est consacrée aux efforts de définition de ce concept et d'identification de ses propriétés caractéristiques. Au niveau de la deuxième partie nous allons expliquer trois des propriétés caractéristiques qualifiées de fondamentales, à savoir : l'accessibilité, l'interopérabilité et la réutilisabilité. Ensuite nous allons souligner l'importance de ces propriétés au niveau de chaque phase du cycle de vie d'un objet d'apprentissage.

2. Définitions et propriétés

2.1. Définitions

Selon Polsani [Polsani, 2003] le terme « Learning Object » a été popularisé pour la première fois par Wayne Hodgins en 1994. Toutefois, depuis cette date à nos jours on est loin d'avoir un consensus concernant sa définition. En effet, il y en a presque autant de définitions que de communautés de praticiens.

D'après l'IEEE Learning Technology Standard Committee [IEEE, 2002] un objet d'apprentissage est toute entité numérique ou non qui peut être utilisée pour l'apprentissage, l'éducation ou la formation. Cette définition est souvent critiquée pour son aspect trop général et son manque de précision. Par exemple, dans [Bourda, 2001] cette définition est considérée comme inutilisable en pratique et illustre toute la difficulté à définir clairement ce qu'est un objet d'apprentissage.

Pour Wiley [Wiley, 2002] un objet d'apprentissage est toute ressource numérique qui peut être réutilisée pour supporter l'apprentissage. Dans cette définition Wiley souligne à la fois la nature numérique de l'objet d'apprentissage et sa capacité d'être réutilisé.

Le Wisconsin Online Resource Center [Wisconsin 2008] définit un objet d'apprentissage comme étant des petites unités d'apprentissage nécessitant une durée de consultation comprise entre deux et quinze minutes. Cette définition se distingue par une volonté de se baser sur un aspect concret et mesurable qui est le temps.

L'Allier [L'Allier, 1997] (cité par [Polsani, 2003]) définit un objet d'apprentissage comme la plus petite expérience structurée et indépendante qui contient un objectif, une activité d'apprentissage et une évaluation. Cette définition fait référence à des aspects liés à l'anatomie de l'objet d'apprentissage et fait référence à des aspects qui relèvent de la pédagogie.

Polsani [Polsani, 2003] définit, quant à lui, un objet d'apprentissage comme une unité de contenu pour l'apprentissage, à la fois indépendante et autonome, qui est prédisposée à la réutilisation dans de multiples contextes éducatifs. Cette définition essaie de dégager clairement l'aspect objet et l'aspect apprentissage qui figurent dans le terme « objet d'apprentissage ». Les termes « indépendance », « autonomie » et « réutilisabilité » font référence à l'aspect objet, alors que les termes « contenu d'apprentissage » et « contextes éducatifs » font référence à l'aspect apprentissage.

2.2. Propriétés

Malgré la multitude des définitions, il existe un certain consensus concernant les propriétés caractéristiques d'un objet d'apprentissage qui sont d'ordre fonctionnel [Polsani, 2003]. Pour mettre en évidence ce consensus nous allons d'abord citer certains travaux qui ont essayé d'énumérer ces propriétés.

Robert [Robert et al., 2003] propose les propriétés caractéristiques suivantes :

- Interopérabilité : caractère de ce qui permet l'utilisation des ressources d'enseignement et d'apprentissage développées par une organisation dans un environnement technologique donné par d'autres organisations dans d'autres environnements technologiques.
- Accessibilité : caractère de ce qui permet la recherche, l'identification et la livraison de ressources d'enseignement et d'apprentissage de façon distribuée.
- Durabilité : caractère de ce qui permet aux ressources d'enseignement et d'apprentissage d'affronter les changements technologiques en minimisant la réingénierie ou le re-développement.
- Pertinence pédagogique : caractère de ce qui permet d'identifier les contextes pédagogiques des ressources d'enseignement et d'apprentissage.
- Collaboration : caractère de ce qui favorise l'échange, la coproduction et l'enrichissement de ressources d'enseignement et d'apprentissage.

- Reconnaissance de la propriété intellectuelle : caractère de ce qui permet de documenter et de reconnaître la propriété intellectuelle et de respecter les droits d'auteur.
- Réutilisabilité et adaptabilité : caractère de ce qui permet la réutilisation de ressources d'enseignement et d'apprentissage à différentes fins, dans différentes applications, dans différents produits, dans différents contextes et par différents modes d'accès.

Dans [Simard, 2002] on trouve les propriétés caractéristiques suivantes :

- Accessibilité : permettre la recherche, l'identification, l'accès et la livraison de contenus et composantes de formation en ligne de façon distribuée.
- Interopérabilité : permettre l'utilisation de contenus et composantes développés par une organisation sur une plate-forme donnée par d'autres organisations sur d'autres plates-formes.
- Réutilisabilité : permettre la réutilisation des contenus et composantes à différentes fins, dans différentes applications, dans différents produits, dans différents contextes et via différents modes d'accès.
- Durabilité : permettre aux contenus et composantes d'affronter les changements technologiques sans la nécessité d'une réingénierie ou d'un re-développement.
- Adaptabilité : permettre la modulation sur mesure des contenus et des composantes.

L'ADL (Advanced Distributed Learning) [ADL, 2004a] propose les propriétés caractéristiques suivantes pour les objets d'apprentissage :

- Accessibilité : la capacité de localiser et accéder à des composants éducatifs dans un emplacement donné et de les délivrer à d'autres emplacements distants.
- Adaptabilité : la capacité d'adapter la formation aux besoins des individus et des organisations.
- Pérennité : la capacité d'améliorer l'efficacité et la productivité par la réduction du temps et des coûts relatifs à la production du contenu éducatif.
- Durabilité : la capacité de résister à l'évolution des technologies et aux changements sans un grand investissement pour refaire la conception, la configuration et le codage du contenu.

- Interopérabilité : la capacité de prendre un composant éducatif développé dans un endroit avec des outils ou des plateformes et l'utiliser dans un autre endroit avec d'autres outils ou plateformes.
- Réutilisabilité : la flexibilité permettant d'incorporer des composants éducatifs dans de multiples applications et contextes.

Polsani [Polsani, 2003] quant à lui propose les propriétés caractéristiques suivantes :

- Accessibilité : il faut marquer les objets d'apprentissage avec des métadonnées pour qu'ils puissent être stockés et référencés.
- Interopérabilité : l'objet d'apprentissage doit être indépendant du média de diffusion et des systèmes qui vont servir à le manipuler.
- Réutilisabilité : une fois créé, un objet d'apprentissage doit être fonctionnel dans des contextes éducatifs variés et différents.

2.3. Synthèse

Sans essayer de proposer une nouvelle définition de la notion d'objet d'apprentissage, nous considérons qu'un objet d'apprentissage est un contenu numérique destiné à l'apprentissage et la formation et décrit par des métadonnées éducatives.

Concernant les propriétés qui doivent caractériser un objet d'apprentissage, la figure suivante résume les différentes propositions abordées. Trois propriétés d'entre elles sont omniprésentes, à savoir : l'accessibilité, la réutilisabilité et l'interopérabilité. Ainsi nous allons par la suite présenter les travaux qui ont été faits pour permettre aux objets d'apprentissage d'être accessibles, réutilisables et interopérables.

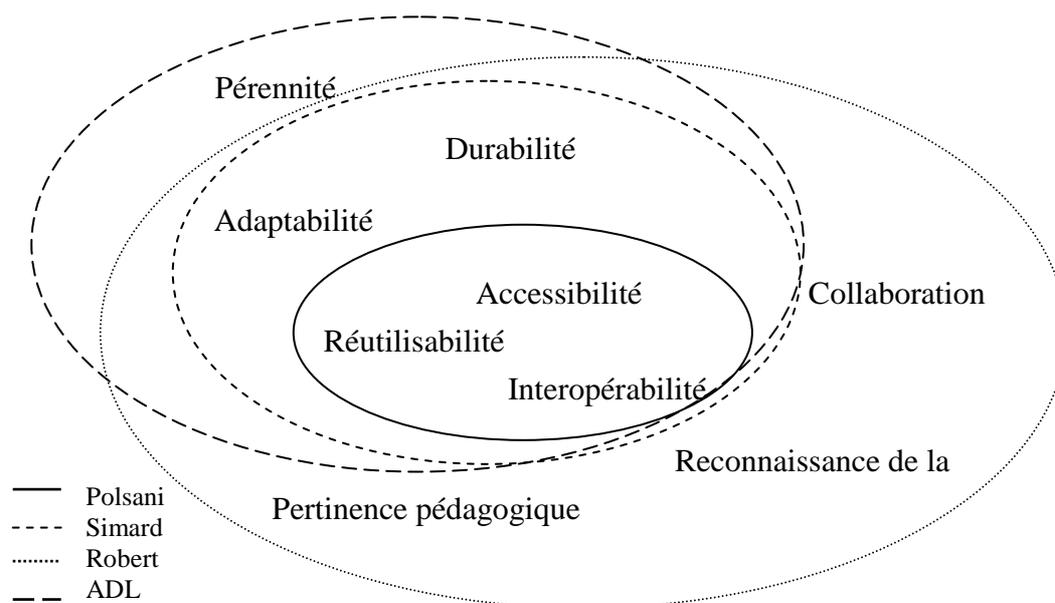


Figure 1 : Propriétés caractéristiques d'un objet d'apprentissage

3. Métadonnées éducatives

Pour permettre la recherche, l'indexation, l'identification, le référencement et l'accès aux objets d'apprentissage dans un contexte centralisé ou distribué il est impératif d'utiliser des métadonnées éducatives. Le standard LOM 1484.12.1 de l'IEEE [IEEE, 2002] fait autorité à ce niveau.

Dans le cadre de ce standard on a souligné que son rôle est de faciliter la recherche, l'évaluation, l'acquisition et l'utilisation des objets d'apprentissage par les apprenants, les instructeurs ou les processus logiciels automatisés. Autrement dit, son rôle est de rendre les objets d'apprentissage accessibles.

3.1. Métadonnées LOM

Les métadonnées les plus utilisées actuellement dans le domaine de la formation en ligne sont celles définies par le standard LOM 1484.12.1 de l'IEEE [IEEE, 2002]. Dans l'introduction du standard il est spécifié que les métadonnées sont des informations à propos d'un objet, qu'il soit physique ou numérique. Leur intérêt est également expliqué comme suit : Comme le nombre d'objets augmente exponentiellement et que notre besoin pour ces objets augmente également d'une façon importante, le manque d'information ou de métadonnées à propos de ces objets instaure une contrainte critique et fondamentale concernant notre capacité à les

découvrir, les gérer et les utiliser. Ainsi, le standard de l'IEEE résout ce problème par la définition d'une structure pour une description interopérable des objets d'apprentissage.

C'est également dans le cadre de ce standard que la définition suivante d'un objet d'apprentissage a été proposée : un objet d'apprentissage est défini comme toute entité – numérique ou non numérique- qui peut être utilisée pour l'apprentissage, l'éducation ou la formation.

Ce standard comporte neuf catégories dont voici les désignations ainsi que leurs rôles :

- La catégorie *général* (general) groupe les informations générales qui décrivent l'objet dans son ensemble.
- La catégorie *cycle de vie* (lifecycle) groupe les caractéristiques liées à l'histoire et l'état courant de l'objet d'apprentissage ainsi que ceux ou celles qui ont affectés cet objet d'apprentissage durant son évolution.
- La catégorie *méta-métadonnées* (meta-metadata) groupe des informations concernant l'instance des métadonnées elle-même.
- La catégorie *technique* (technical) groupe les exigences ainsi que les caractéristiques techniques de l'objet d'apprentissage.
- La catégorie *éducative* (educational) contient les caractéristiques éducatives et pédagogiques de l'objet d'apprentissage.
- La catégorie *droits* (rights) groupe les droits de propriété intellectuelle et les conditions d'utilisation de l'objet d'apprentissage.
- La catégorie *relation* (relation) groupe les traits qui définissent la relation entre cet objet et d'autres objets d'apprentissage.
- La catégorie *annotation* (annotation) offre des commentaires sur l'utilisation éducative de l'objet d'apprentissage ainsi que des informations sur quand et par qui les commentaires ont été créés.
- La catégorie *classification* (classification) décrit l'objet d'apprentissage par rapport à un système de classification.

Pour être strictement conforme à LOM il faut que l'instance LOM ne soit composée que d'éléments de données LOM. Une instance LOM conforme peut contenir des éléments en

extension. En plus, une instance LOM qui ne contient aucune valeur pour les éléments LOM est une instance conforme d'après le standard LOM de l'IEEE.

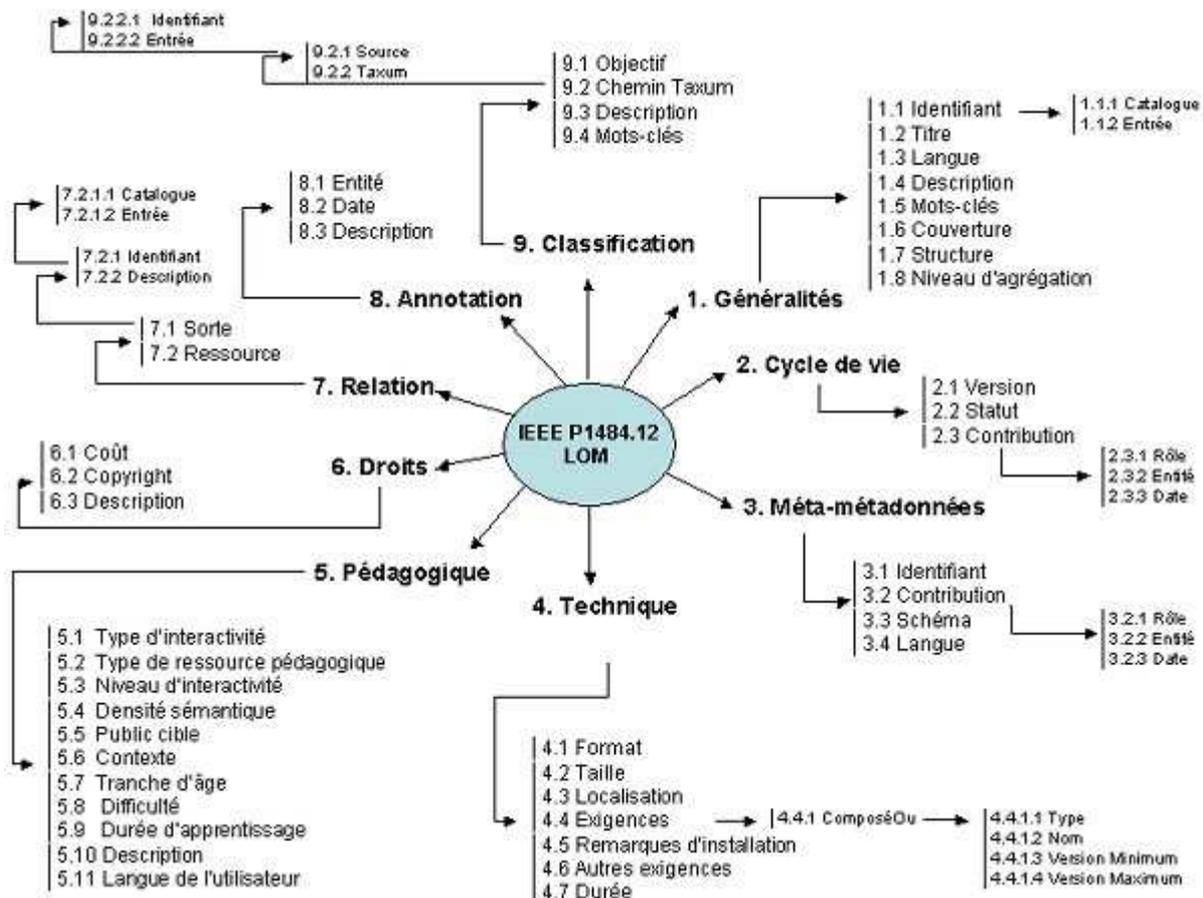


Figure 2 : Organisation du schéma de métadonnées LOM [Passardièrre et Jarraud, 2004]

3.2. Profils d'application LOM

Le schéma de métadonnées LOM de l'IEEE est souvent jugé comme compliqué et difficile à utiliser [Xiang et al., 2003]. Ceci a poussé certaines communautés de praticiens à définir leurs propres variantes appelées profils d'application LOM. D'une façon générale « un profil d'application est une sélection d'éléments d'une norme, d'un standard ou d'une spécification formant ainsi un sous-ensemble adapté aux besoins des groupes qui l'utilisent » [Robert et al., 2003].

Les profils LOM se distinguent par le fait de rendre certains éléments obligatoires, d'autres optionnels ou encore de définir des contraintes sur les valeurs que peuvent prendre certains éléments. Il est également possible d'ajouter des éléments supplémentaires. Toutefois il est

important de garder toujours un haut niveau d'interopérabilité avec le standard LOM et les profils d'applications en relation [Duval et al., 2002].

C'est dans ce cadre que plusieurs profils LOM ont été définis tels que : le profil CanCore [CANCORE, 2003], le profil NORMETIC [CREPUQ et Novasys, 2003], l'UK LOM Core [UK LOM Core, 2003], le profil CELTS (Chinese e-Learning Technology Standard) [CELTS, 2003], le profil d'application LOM de l'ARIADNE [Najjar et Duval, 2003], le profil Celebrate [Celebrate, 2003], le LOM-INSA [DOC'INSA, 2004], le profil ManUeL [De La Passadière et Jarraud, 2004] et le LOM-FR [AFNOR, 2006].

[Duval et al., 2006] propose une démarche en dix étapes pour mettre en place de tels profils d'application. Au niveau de la première étape il faut étudier et identifier ses besoins. La compréhension de ses propres besoins est fondamentale pour pouvoir choisir au niveau de la deuxième étape les éléments à retenir du standard. Une fois que ses éléments sont identifiés, la troisième étape est consacrée à la modification éventuelle du nombre d'occurrences autorisées de chaque élément. Au niveau de la quatrième étape, il s'agit d'adopter certains espaces de noms existants au niveau de certains éléments. Si nécessité, il faut définir son propre espace de noms pour certains éléments au niveau de la cinquième étape. La sixième étape est prévue pour décider pour chaque élément s'il est obligatoire, conditionnel, recommandé ou optionnel. Au niveau de l'étape sept, il est possible d'ajouter des restrictions sur l'espace de valeurs permises au niveau du standard. La huitième étape concerne la définition des relations et des dépendances entre les éléments. L'étape neuf concerne la possibilité de raffiner les types de données retenus dans le standard. La dernière étape concerne la définition du schéma RDF ou XML permettant de décrire le profil d'application LOM conçu.

4. Interopérabilité

Plusieurs travaux ont été menés pour définir des spécifications, des standards et des normes afin de promouvoir l'interopérabilité au niveau de la formation en ligne. Plusieurs aspects ont été couverts par cet effort, tels que les métadonnées, le modèle de l'apprenant et le format de livraison des objets d'apprentissage. Nous allons tout d'abord rappeler le rôle des spécifications, des standards et des normes dans la formation en ligne, puis nous allons présenter un support d'interopérabilité des objets d'apprentissage : la spécification SCORM.

4.1. Normes, standards et spécifications

Les normes, les standards et les spécifications jouent un rôle central pour garantir l'interopérabilité dans la formation en ligne. En fait, comme l'affirme Ely [Ely, 2005] le besoin d'interopérabilité est un besoin connu et justifié. Ces retombées certes sont aussi bénéfiques pour le consommateur que pour le producteur et ceci quelque soit le domaine concerné. Côté consommateur, celui-ci ne risque plus d'être prisonnier d'un vendeur ou d'un produit. Ainsi, le consommateur bénéficie d'un plus grand choix de produits. Côté vendeur l'interopérabilité permet une ouverture à la concurrence puisque ses produits sont normalisés. Mieux encore, la taille du marché est par conséquence beaucoup plus importante. En plus, le fait de produire un produit qui fonctionne partout sans adaptations ou modifications permet de diminuer les coûts.

« Du point de vue du développeur de contenu pédagogique, les normes permettent de produire le matériel pédagogique dans un seul format utilisable par plusieurs systèmes ou outils de formation en ligne.

Les normes facilitent aussi le travail du designer ou concepteur de matériel, en donnant accès à de larges banques d'objets ou ressources pédagogiques réutilisables, réduisant de ce fait le besoin de développer un produit en fonction de plusieurs systèmes. Les normes incitent également à créer des contenus modulaires plus faciles à maintenir et à mettre à jour. » [Ely, 2005]

« L'émergence de normes d'interopérabilité pour la formation en ligne se situe au croisement des nouvelles possibilités éducatives qu'offrent l'Internet et les TIC et des pressions financières que vivent les institutions éducatives. Cette poussée pédagogique et ces contraintes économiques engendrent une volonté de bonifier l'acte pédagogique, d'améliorer l'accessibilité au « meilleur » du monde de l'éducation, de gagner du temps, de réduire les coûts en partageant et réutilisant le matériel disponible et offert en ligne grâce à l'adoption de normes d'interopérabilité. » [Chouinard, 2002]

Dans la suite nous allons mettre l'accent sur SCORM comme étant l'un des formats les plus utilisés pour l'échange d'objets d'apprentissage.

4.2. SCORM

SCORM (Sharable Content Object Reference Model) est une spécification définie par l'initiative ADL (Advanced Distributed Learning) combinant des éléments produits par

l'ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe), l'IEEE (Institute of Electrical and Electronics Engineers), l'AICC (Aviation Industry CBT Committee) et l'IMS (Instructional Management System) [ADL, 2004a].

Cette spécification est décrite par quatre documents. Le premier [ADL, 2004a] donne une vue d'ensemble sur la spécification et ces différentes composantes. Il s'agit d'une introduction comportant des informations de haut niveau conceptuel comme le précise le document lui-même. Le deuxième document [ADL, 2004b] couvre les aspects liés à l'assemblage, l'étiquetage et le paquetage du contenu éducatif. Les aspects liés à l'environnement d'exécution, tels que la communication avec le système de gestion de contenu ou de la formation et la gestion des erreurs, sont traités dans le troisième document [ADL, 2004c]. Le quatrième document [ADL, 2004d] traite les aspects liés au séquençement et à la navigation dans le contenu éducatif.

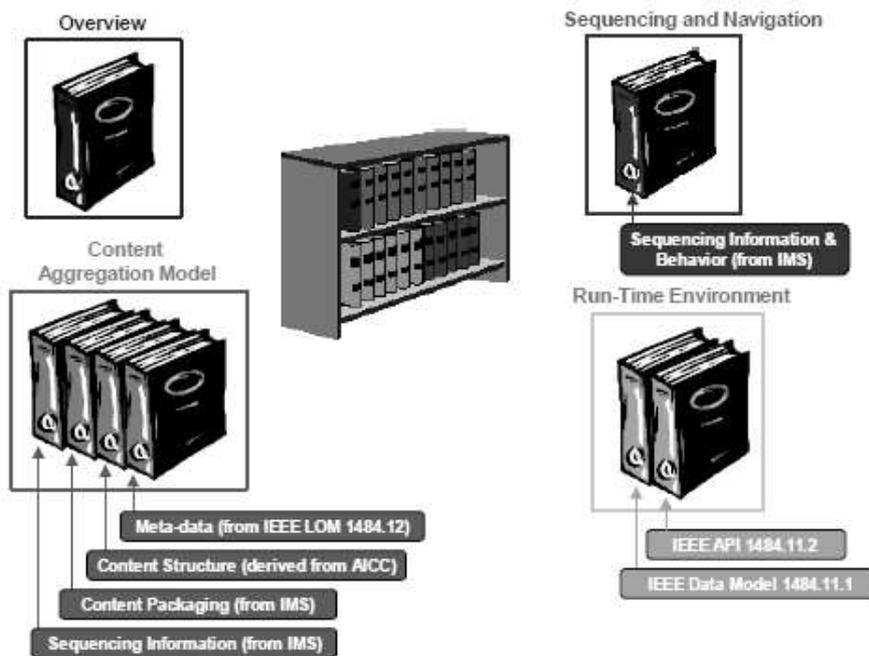


Figure 3 : Les quatre documents de référence de la spécification SCORM

5. Réutilisabilité

Un objet d'apprentissage doit être également réutilisable. Cette réutilisabilité peut être faite à deux niveaux. En effet, un objet peut être réutilisé dans plusieurs contextes d'apprentissage comme premier niveau de réutilisation. Le deuxième niveau de réutilisation qui nous intéresse est celui qui permet de réutiliser des objets d'apprentissage en les assemblant ensemble pour

former un nouvel objet d'apprentissage. C'est ce deuxième niveau de réutilisation qui nous intéresse.

Pour avoir cette propriété de réutilisabilité l'objet doit être déjà accessible et interopérable. Les aspects relatifs à l'accessibilité et l'interopérabilité étant déjà introduits, nous allons nous intéresser à deux autres aspects propres à la réutilisabilité : la granularité et l'assemblage.

5.1. Granularité

La granularité d'après Robert [Robert et al. 2003] est un niveau de découpage, restant cohérent et compatible, d'un contenu pédagogique en une série d'items élémentaires, appelés grains, que l'on peut re-combiner dans le déroulement de parcours pour répondre aux besoins individuels de formation. En fait, le niveau de granularité d'un objet d'apprentissage fait le plus souvent référence à sa taille. Plusieurs travaux ont fait le lien entre la capacité de réutilisation d'un objet et son niveau de granularité. David A. Wiley [Wiley, 2002] affirme que cette capacité de réutilisation est forcément inversement proportionnelle à la taille de l'objet. En d'autres termes plus l'objet est d'une granularité fine plus il a le potentiel d'être réutilisé pour créer de nouveaux objets d'apprentissage par assemblage. Toutefois, une question se pose concernant la granularité la plus appropriée d'un objet d'apprentissage afin de rendre la propriété de réutilisabilité effective.

Le standard LOM de l'IEEE [IEEE, 2002] considère quatre niveaux de granularité au niveau de la catégorie « General » l'élément « Aggregation Level ». Comme l'indique le standard cette information concerne la granularité fonctionnelle de l'objet d'apprentissage et peut prendre une valeur de un à quatre. Le niveau un est le niveau d'agrégation le plus petit et il correspond à des objets élémentaires tels qu'une image, ou un texte. Le niveau deux consiste en des objets formés d'objets de niveau un. C'est le cas des leçons composés de textes et d'images. Le niveau trois correspond aux cours composés de leçons. Le niveau quatre correspond au niveau de granularité le plus élevé. C'est le cas par exemple d'un ensemble de cours qui permettent d'avoir une certaine certification. Les objets de niveau quatre peuvent être composés d'objet de niveau de granularité trois et quatre.

Dans le cas de SCORM [ADL, 2004a] trois niveaux de granularité ont été défini. Les objets de la plus fine granularité sont appelés « assets ». Dans cette catégorie on peut classer par exemple les pages Web, les images et les textes. Les objets faisant partie du deuxième niveau de granularité sont appelés « SCOs » (Sharable Content Objects). Un objet de type « SCO » est formé d'une collection d'« assets » et il peut être traité par un système de gestion de

l'apprentissage (Learning Management System). Un « SCO » doit avoir en plus la capacité de permettre le suivi des interactions entre les apprenants et le contenu. Finalement, le plus haut niveau de granularité concerne le « content aggregation ». Il s'agit de grouper des objets de type « asset » et « SCO » suivant une certaine structuration conformément au Modèle d'agrégation de contenu (Content Aggregation Model) [ADL, 2004b]. Ce niveau correspond à un contenu éducatif qui consiste par exemple en un chapitre, un cours ou un module.

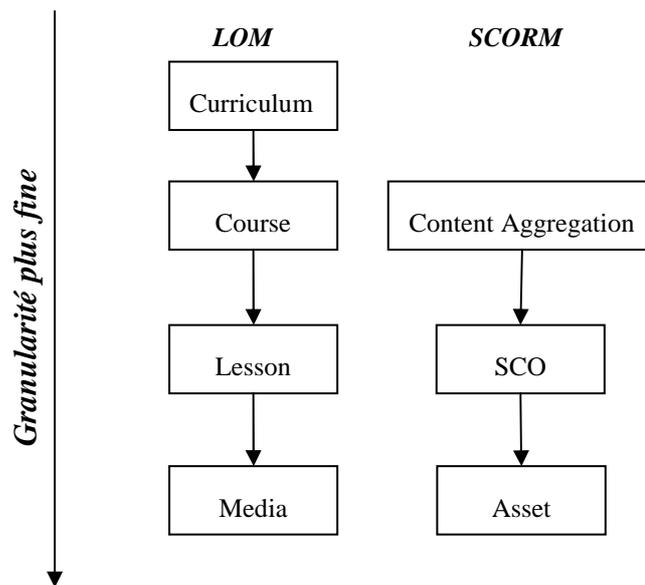


Figure 4 : Niveaux de granularité selon LOM et SCORM

5.2. Assemblage

Différents termes sont utilisés pour décrire l'activité qui consiste à combiner des objets d'apprentissage d'une certaine façon pour produire un nouvel objet. Parmi les termes utilisés il y en a « assemblage » et « composition ». Wiley définit l'assemblage ou la composition comme étant le fait de prendre des objets d'apprentissage et de les combiner ensemble d'une certaine façon qui permet de leur donner un sens d'un point de vue éducatif [Wiley, 2002].

L'assemblage des objets d'apprentissage doit suivre un certain nombre de règles et de principes dictés par les règles d'interopérabilité (i.e. les spécifications, les standards et les normes) et par les règles pédagogiques (i.e. les théories et les approches pédagogiques). Comme le précise Wiley [Wiley, 2002] c'est la métaphore des molécules de chimie qui

illustre le plus fidèlement le principe de réutilisation des objets d'apprentissage pour en former des nouveaux.

Au niveau des spécifications l'IMS a proposé, entre autres, l'« IMS Simple Sequencing » [IMS, 2003a] qui a été adopté au niveau de SCORM. Cette spécification offre les moyens nécessaires pour séquencer et organiser les activités d'apprentissage de diverses façons. Cette organisation prend la forme d'une arborescence hiérarchique. Cette spécification offre au concepteur ou développeur d'un contenu éducatif les moyens pour déclarer l'ordre de présentation des éléments de contenu à l'apprenant et les conditions selon lesquelles un élément est sélectionné, délivré ou ignoré durant la présentation du contenu à l'apprenant.

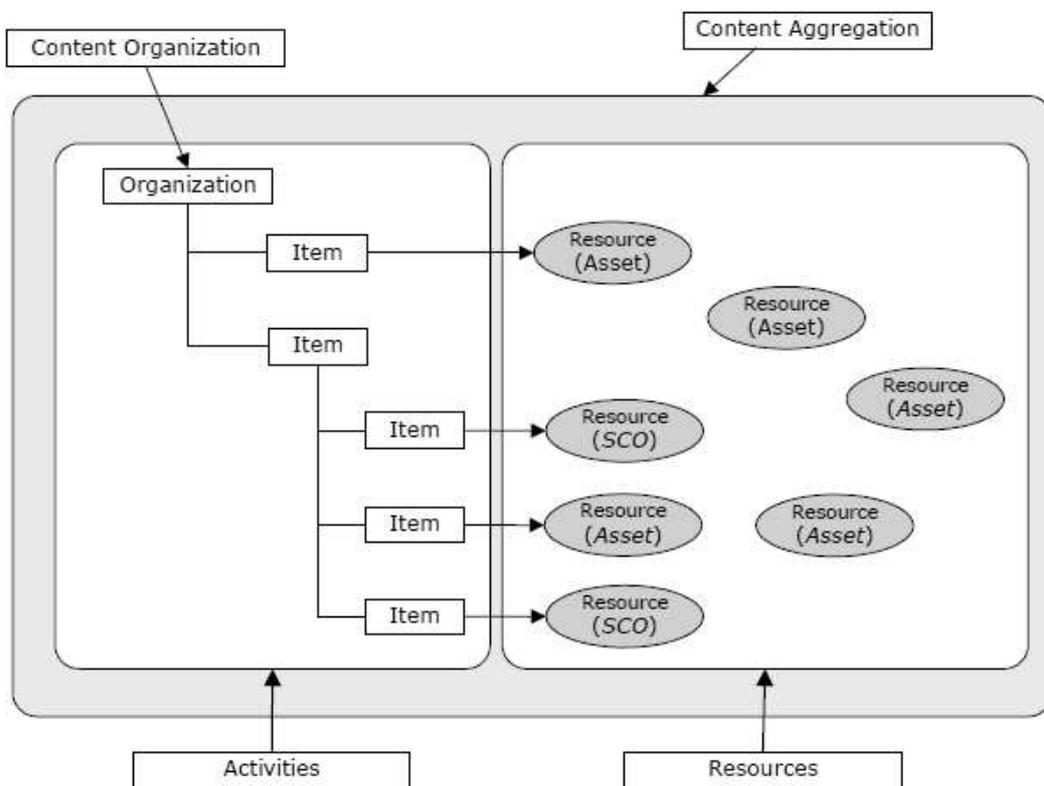


Figure 5 : Modèle d'agrégation de contenu [ADL, 2004b]

6. Cycle de vie d'objets d'apprentissage

6.1. Modèles de cycle de vie

L'objectif d'un modèle de cycle de vie est de présenter les différentes étapes par lesquelles un objet d'apprentissage passe durant son existence. Plusieurs modèles ont été proposés dans la littérature tels que le modèle de Rensing [Rensing et al., 2005] et le modèle de Strijker [Strijker, 2004]. Ces modèles peuvent être utilisés de plusieurs façons. Par exemple, ils peuvent être utilisés pour analyser les propriétés qui peuvent avoir une influence sur le succès ou non d'un objet d'apprentissage. Egalement, ils permettent d'identifier les acteurs intervenant sur un objet d'apprentissage le long de son existence. En plus, ils présentent un moyen permettant d'énumérer les outils nécessaires pour la gestion des objets d'apprentissage.

6.2. Modèle de Rensing

Dans [Rensing et al., 2005] les auteurs de ce modèle proposent un cycle de vie en quatre étapes :

- Conception (authoring) : il s'agit d'utiliser des logiciels auteurs afin de mettre en place un nouvel objet d'apprentissage.
- Re-conception (re-authoring) : il s'agit de reprendre un objet d'apprentissage existant et de le modifier. La modification peut être une mise-à-jour, une amélioration, une rectification ou une adaptation à un nouveau contexte pédagogique.
- Publication (provision) : il s'agit de publier l'objet d'apprentissage, le plus souvent en le déposant dans une banque d'objets d'apprentissage, pour le partager et le rendre accessible pour toute sorte d'utilisation.
- Apprentissage (learning) : il s'agit d'exploiter l'objet d'apprentissage à des fins d'apprentissage le plus souvent via une plate forme d'apprentissage en ligne.

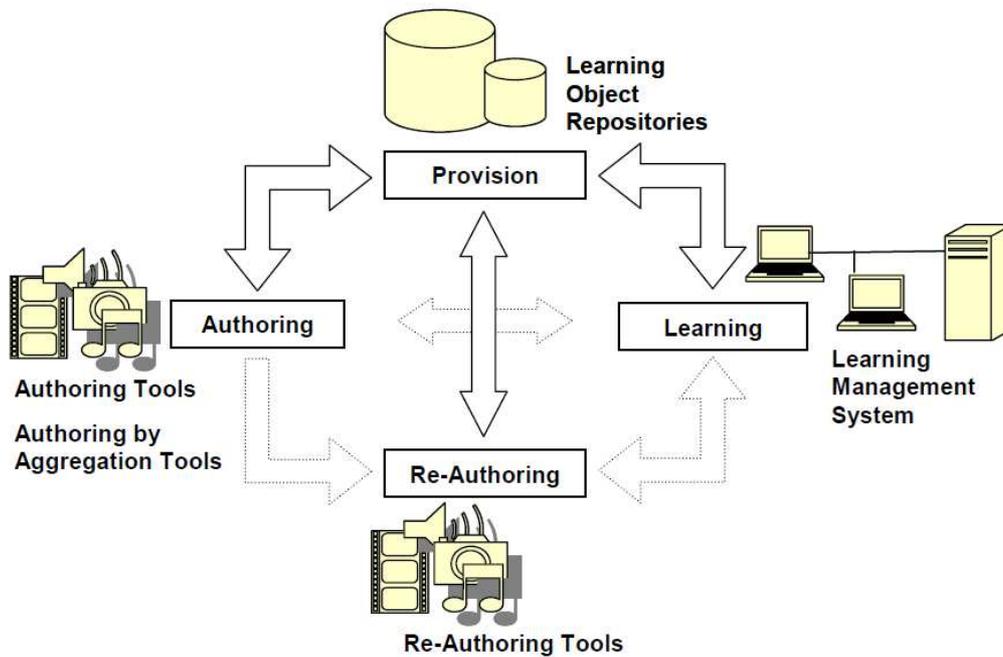


Figure 6 : Modèle de cycle de vie d'un objet d'apprentissage [Rensing et al., 2005]

6.3. Modèle de Strijker

Dans [Strijker, 2004] nous avons un modèle de cycle de vie d'un objet d'apprentissage défini en six étapes :

- Création (Obtaining) : il s'agit de concevoir des objets d'apprentissage en utilisant les moyens techniques et humains nécessaires.
- Annotation (Labelling) : il s'agit de renseigner les métadonnées éducatives des objets d'apprentissage conçus.
- Publication (Offering) : il s'agit de rendre les objets d'apprentissage disponibles en les publiant dans des banques d'objets d'apprentissage.
- Utilisation (Using) : il s'agit d'utiliser les objets d'apprentissage soit pour l'apprentissage soit pour créer d'autres objets par assemblage. Les objets peuvent subir des modifications afin de les adapter au nouveau contexte.
- Rétention (Retaining) : il s'agit d'étudier si un objet doit être conservé, révisé ou retiré de la banque d'objets d'apprentissage.

6.4. Synthèse

Le cycle de vie d'un objet d'apprentissage comporte nécessairement trois phases : Une phase de création, une phase de publication et une phase d'exploitation. Chaque méthode de cycle

de vie reprend ces différentes phases en détaillant ou non certains aspects suivant l'objectif visé. Par exemple, dans [Catteau et al. 2006] l'objectif était d'étudier l'impact du cycle de vie sur les métadonnées éducatives.

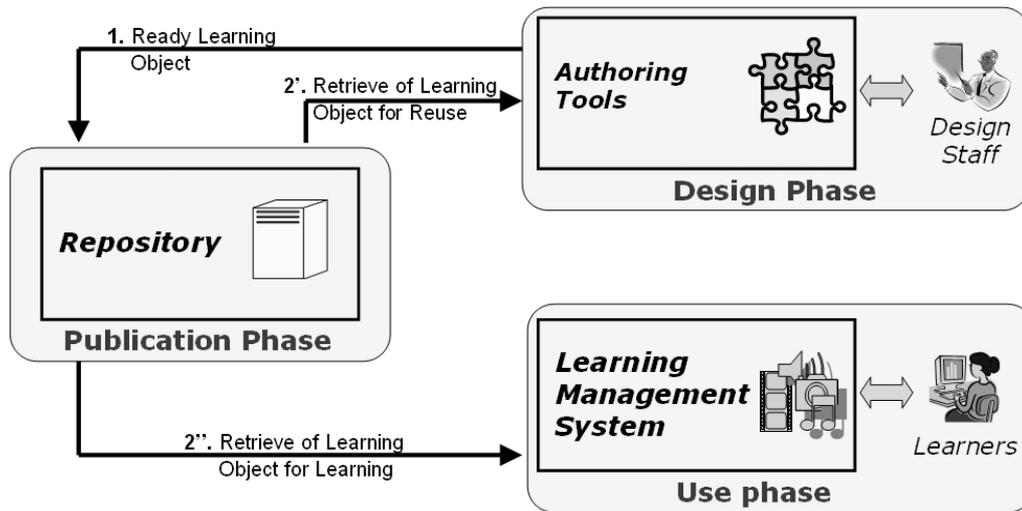


Figure 7 : Cycle de vie d'un objet d'apprentissage [Farhat et al., 2007b]

Il faut noter que chaque objet d'apprentissage acquiert ses caractéristiques au moment de sa conception. En effet, il s'agit d'un moment crucial influençant directement sur son efficacité lors de la phase d'exploitation. Par exemple, si au moment de la conception l'aspect annotation a été négligé, après l'objet sera difficilement accessible lorsqu'un utilisateur en aura besoin. Dans ce cas ce sont les aspects utilisation et réutilisation qui seront affectés. C'est la raison pour laquelle il faut apporter le plus d'attention à la phase de conception d'objets d'apprentissage.

Ainsi, il est nécessaire d'anticiper au niveau de la phase de conception pour répondre aux exigences liées aux phases publication et exploitation. C'est à travers l'étude de ces trois phases du cycle de vie d'un objet d'apprentissage qu'il sera possible d'identifier les principales classes de propriétés auxquels il faut attribuer le plus d'attention lors de la phase de conception.

6.4.1. Phase d'utilisation

L'utilisation d'un objet d'apprentissage se fait via un LMS (Learning Management System). Celui-ci permet son déploiement au profit de l'apprenant. Le succès du déploiement est conditionné par le degré de conformité aux spécifications, aux normes et aux standards supportés par le LMS. Ce déploiement est tout un processus faisant intervenir de nombreux paramètres et il est généralement centré sur l'apprenant. En effet, celui-ci commence par

lancer le processus en exprimant son besoin de récupérer un contenu permettant d'accomplir un objectif pédagogique particulier. Ainsi, le LMS en se basant sur des informations complémentaires, généralement issues du profil de l'apprenant, reformule la requête et l'envoie à un ou à plusieurs entrepôts. A la réception d'une requête l'entrepôt fournit un résultat contenant les identifiants des objets d'apprentissage qui lui sont conformes. Le succès de cette opération est fortement lié à la richesse des métadonnées éducatives des objets d'apprentissage. Finalement, l'objet d'apprentissage retenu est récupéré et soumis à un processus de personnalisation de contenu et de forme (selon l'approche retenue) afin de répondre aux indications mentionnées dans le profil de l'apprenant. Ceci suppose que l'objet est conçu de façon qu'il soit personnalisable tel que c'est le cas dans l'approche SIMBAD [Duitama, 2005] via l'utilisation de la notion d'opérateurs.

Il est clair que l'accessibilité, l'interopérabilité et la réutilisabilité influencent directement le succès ou l'échec de cette phase du cycle de vie de l'objet d'apprentissage. L'auteur d'un objet d'apprentissage doit donc soigner ces trois propriétés lors de la phase de conception.

6.4.2. Phase de publication

La publication des objets d'apprentissage se fait au niveau des entrepôts appelés aussi banques d'objets d'apprentissage. Ceux-ci fournissent des services de stockage, d'indexation, de recherche et de livraison des objets d'apprentissage. La livraison peut être faite en réponse à une demande d'un apprenant lors d'une session d'apprentissage en ligne. Mais, également cela peut être fait en réponse à une demande d'un auteur afin qu'il soit réutilisé dans la conception d'un nouvel objet d'apprentissage. Les opérations d'indexation, de recherche et de localisation des objets d'apprentissage dans un entrepôt se basent sur les informations contenues dans les métadonnées.

L'auteur doit donc soigner le renseignement des métadonnées autour desquelles se basent les fonctionnalités de tout entrepôt d'objets d'apprentissage.

6.4.3. Phase de conception

La phase de conception est généralement faite de façon incrémentale et itérative. A chaque itération le concepteur doit vérifier un certain nombre de propriétés, en particulier les aspects pédagogiques. En effet, un objet d'apprentissage est avant tout un didacticiel qui doit avoir une certaine qualité pédagogique afin d'accomplir correctement son rôle. Mais, dans certains contextes la neutralité pédagogique peut être requise pour faciliter la réutilisation par d'autres

auteurs dans d'autres contextes. Un autre élément à considérer est l'adhésion à une spécification ou un standard de livraison afin qu'il puisse être traité par d'autres environnements auteur. Les métadonnées jouent également ici un rôle important pour favoriser la découverte et la réutilisation de l'objet par d'autres auteurs.

7. Défis de la composition de nouveaux objets d'apprentissage par réutilisation

7.1. Défis de la composition par réutilisation

Malgré le fait que l'on dispose aujourd'hui d'un nombre important d'objets d'apprentissage disponibles dans des banques telles que : ARIADNE¹, Educanext², RPL³ ou autres, et bien que l'on dispose également de plusieurs modèles de composition par réutilisation d'objets d'apprentissage tels que IMS CP [IMS, 2004], SCORM [ADL 2004b] et SIMBAD [Duitama, 2005], les auteurs des objets d'apprentissage n'ont pas adopté à grand échelle ce mode de création d'objets d'apprentissage.

En fait, pour pouvoir créer des objets composés par réutilisation il faut être capable de surmonter un certain nombre de défis. Le premier défi est la *recherche efficace* d'objets d'apprentissage à réutiliser conformément aux besoins, aux exigences et aux préférences de l'auteur. En effet, l'auteur risque à ce niveau soit de faire face au problème de bruit qui se manifeste par un nombre très élevé de réponses à sa requête, soit au problème de silence qui se manifeste par un nombre très faible de réponses. Dans le premier cas, la tâche de trouver les objets les plus pertinents parmi les résultats risque d'être fastidieuse et consommatrice de temps, alors que dans le deuxième cas le choix risque d'être très restreint pour l'auteur.

Le deuxième défi à surmonter est la *flexibilité du modèle de composition*. Celui-ci doit être en mesure de supporter les choix des auteurs en termes de composition. En effet, les auteurs peuvent avoir des convictions très différentes en termes d'approches pédagogiques. Le modèle de composition doit ainsi supporter cette diversité et cette richesse pour qu'il soit adopté par les auteurs.

¹ <http://www.ariadne-eu.org/>

² <http://www.educanext.org>

³ <http://cours.uvt.rnu.tn/rpl/>

Le troisième défi porte sur la vérification de la *cohérence de la composition*. La vérification doit couvrir les aspects relatifs à la conformité au modèle de composition mais surtout la cohérence sémantique de l'objet composé. Dès que les objets réutilisés dans la composition ne sont pas forcément ceux de l'auteur, des incohérences sémantiques peuvent apparaître soit suite à un mauvais choix d'un ou de plusieurs objets parmi ceux réutilisés, soit à cause de la façon avec laquelle la composition du nouvel objet est faite.

Le quatrième défi est celui de la *compréhension* par l'auteur des propriétés et des caractéristiques du nouvel objet composé. En fait, l'auteur entame la phase de composition avec l'intention de créer un nouvel objet d'apprentissage ayant certaines caractéristiques. Toutefois, le fait qu'il est en train de réutiliser des objets qui ne sont pas forcément les siens, qui peuvent être nombreux et composés de manière spécifique peut entraîner deux difficultés. La première est de vérifier que l'objet tel qu'il était conçu est conforme aux intentions et aux objectifs fixés. Alors que la deuxième est de vérifier si l'objet comporte des propriétés et des caractéristiques particulières qui ne sont pas parmi celles voulu intentionnellement. Surmonter ce défi permettra aux auteurs soit de valider soit d'améliorer leurs objets d'apprentissage composés par réutilisation.

Le cinquième défi consiste à vérifier automatiquement la *conformité* à un patron de conception. En fait, un patron de conception est un modèle qui doit être respecté par les objets d'apprentissage composés. Ces patrons de conception peuvent par exemple permettre de suivre une approche pédagogique donnée ou d'homogénéiser les objets d'apprentissage dans un contexte d'usage particulier. Pouvoir vérifier automatiquement la conformité d'un objet composé à un patron de conception est un défi puisque la vérification manuelle est une tâche fastidieuse et consommatrice du temps.

Le sixième défi est de favoriser la composition d'objets d'apprentissage *réutilisables* à leur tour. En fait, lors de la composition d'un nouvel objet composé la propriété de réutilisation peut se trouver négligée puisque les considérations pédagogiques et didactiques sont le plus souvent celles qui sont mises en avant. Cependant, l'une des propriétés fondamentales d'un objet d'apprentissage, qu'il soit composé ou non, est sa capacité à être réutilisé.

7.2. Etat de l'art de la composition par réutilisation et problématiques ouvertes

Actuellement, certains outils auteur supportent la composition par réutilisation tels que : RELOAD⁴ et EXE⁵. La plupart d'entre eux supportent les modèles de composition IMS CP [IMS, 2004] et SCORM [ADL, 2004b]. On peut constater, suite à l'étude de ces outils, qu'ils sont loin de répondre aux défis énoncés de la composition par réutilisation. En effet, ils n'intègrent pas des outils de recherche d'objets à réutiliser efficaces, le modèle de composition supporté n'est pas assez flexible d'où la proposition de spécifications complémentaires tels que l'IMS Simple Sequencing [IMS, 2003a] et l'IMS Learning Design [IMS, 2003c]. En plus, ils ne permettent pas ni de vérifier la cohérence sémantique ni de permettre à l'auteur de comprendre le résultat de la composition d'une façon explicite. Finalement, ils n'intègrent ni des fonctionnalités permettant de vérifier la conformité à un patron de conception ni des moyens pour favoriser la production d'objets d'apprentissage à leur tour réutilisables.

Au niveau des travaux de recherche certains défis ont été étudiés. En particulier, la recherche d'objets d'apprentissage pertinents a été traitée par des travaux tels que [Ochoa and Duval, 2007]. Egalement, plusieurs modèles de composition flexibles ont été proposés tels que [Duitama, 2005]. La conformité à un patron de composition donné a fait également l'objet de plusieurs travaux tels que [Derntl et Motschnig-pitrik, 2003] et [Trigano et Giacomini-Pacurar, 2004]. Toutefois, ces travaux ne donnent que quelques éléments de réponses. De plus, les défis relatifs à la cohérence de la composition (en particulier d'un point de vue sémantique) et l'aide à la compréhension de l'objet composé n'ont pas été à notre connaissance traités pas des travaux de recherche d'une façon directe et satisfaisante.

8. Conclusion

Tout au long de ce chapitre nous avons passé en revue l'état de l'art concernant le concept d'objet d'apprentissage. Nous avons en particulier mis l'accent sur les divers essais de définition ainsi que sur l'étude des propriétés caractéristiques. Trois propriétés sont unanimement acceptées comme propriétés caractéristiques d'un objet d'apprentissage, à savoir l'accessibilité, l'interopérabilité et la réutilisabilité. Nous avons souligné l'importance de ces propriétés au niveau de chaque phase du cycle de vie d'un objet d'apprentissage.

⁴ <http://www.reload.ac.uk/>

⁵ <http://exelearning.org/>

Ensuite, nous avons détaillé les différents défis à relever afin de rendre la composition par réutilisation d'objets d'apprentissage effective et efficace. Nous avons, également réalisé l'état de l'art vis-à-vis de ces défis.

Dans le chapitre suivant nous allons étudier le processus de composition et de personnalisation des documents numériques de façon générale. Puis, nous allons nous focaliser sur l'approche SIMBAD en tant qu'exemple d'approche qui s'intéresse particulièrement aux objets d'apprentissage. Finalement nous allons énoncer les problématiques auxquelles nous nous intéressons dans cette thèse par rapport aux approches de composition par réutilisation d'objets d'apprentissage.

Chapitre II :

COMPOSITION ET PERSONNALISATION

1. Introduction

Durant ces dernières années plusieurs travaux ont traité les aspects assemblage et adaptation/personnalisation des contenus numériques. Les objets d'apprentissage en tant que documents numériques ont emprunté certaines techniques de personnalisation/adaptation de ces travaux.

Dans ce chapitre nous allons présenter le concept de documents virtuels adaptatifs et le concept d'hypermédia adaptatif. Nous allons ensuite introduire l'approche SIMBAD [Duitama, 2005] qui s'est inspiré de ces concepts pour proposer un modèle pour les objets d'apprentissage adaptatif/personnalisable ainsi que les éléments nécessaires pour leur gestion.

2. Composition et personnalisation de documents numériques

Un objet d'apprentissage est un document numérique et donc bénéficie des avantages apportés par ce type de documents mais également de certains défauts. En effet, plusieurs problématiques relatives aux documents numériques ont été identifiées [Ranwez, 2000]. Parmi elles, certaines prennent plus d'importance dans le contexte de la formation en ligne, telles que :

- Problème de perte de la structure du document : Les documents numériques sont souvent des documents hypertexte avec une structure non linéaire. Dans un contexte d'apprentissage la liberté de navigation et de parcours du document peut avoir un effet négatif. L'apprenant peut se perdre, ce qui peut se manifester sous plusieurs formes. En effet, certains apprenants choisissent certains chemins de navigation qui ne comportent pas les éléments nécessaires par rapport au concept étudié via le document électronique. D'autres apprenants se retrouvent avec la sensation d'être abandonné et perdu dans le document.

- Problème de la surcharge cognitive : Si un apprenant se trouve en face d'un nombre important de connaissances à acquérir, au bout d'un certain temps il peut être atteint du phénomène de surcharge cognitive. Ceci est expliqué par le fait que la mémoire de travail a une

capacité limité. D'autre part, l'auteur du document numérique souhaite généralement toucher la plupart des apprenants. Ainsi, il met le maximum de connaissances dans le document.

- Problème de recherche des informations pertinentes : Pour un apprenant qui n'est pas le plus souvent connaisseur du domaine d'études il est difficile de pouvoir repérer les informations pertinentes dans un document numérique. La tâche devienne de plus en plus difficile lorsque le document offre beaucoup d'alternatives de navigation.

Pour remédier à ces problèmes relatifs en particulier aux documents numériques plusieurs techniques ont été inventées et utilisées dans le cadre des documents virtuels adaptatifs et des hypermédias adaptatifs.

2.1. Documents virtuels adaptatifs

2.1.1. Définitions et principes

Un document virtuel adaptatif est un document numérique généré par assemblage de fragments de contenus via un processus automatique guidé par l'utilisateur. Ce type de document numérique a deux caractéristiques principales [Tazi et Altawki, 1999] : il répond à un besoin d'interactivité et il est généralement éphémère.

En effet, il répond à un besoin d'interactivité car il est assemblé en réponse à une requête d'un utilisateur donné à un instant précis concernant un besoin d'information spécifique. Donc le processus de composition de document est initialisé suite à une interaction entre l'utilisateur et le système. Le résultat doit dépendre fortement de l'utilisateur initiateur de la requête et ceci sur plusieurs plans : besoins d'information, connaissances préalables et préférences en terme de médias. C'est cet aspect qui est exprimé par le terme « adaptatif ».

Un document virtuel adaptatif est généralement éphémère car il est le résultat instantané d'un assemblage de fragments existants. Cependant, ces fragments évoluent dans le temps puisque ils peuvent apparaître, disparaître ou changer entre deux instants différents. Ainsi, une même requête lancée par un même utilisateur à deux instants différents risque de retourner deux résultats différents. C'est cet aspect qui est exprimé par le terme « virtuel ».

2.1.2. Composition et personnalisation

Plusieurs approches de composition et de personnalisation de documents virtuels ont été proposées. Nous nous limitons à deux d'entre elles qui illustrent les mécanismes de composition et de personnalisation des documents virtuels adaptatifs.

Dans [Ranwez et Ranwez, 2001] une approche de composition en quatre étapes a été proposée. La première étape c'est la recherche d'informations. Elle permet de récupérer les briques d'informations candidates à faire partie du document final. Cette recherche se base sur les objectifs de l'utilisateur et sur la description sémantique des briques d'informations.

Au niveau de la deuxième étape les briques d'informations candidates sont filtrées cette fois à base du profil de l'apprenant et du temps pendant lequel il est disponible pour consulter le document.

La troisième étape c'est l'organisation des briques d'informations au sein du document. A ce niveau l'organisation se base sur une trame narrative jugée pertinente et sur les rôles joués par chaque brique.

La dernière étape est celle de l'assemblage du document. Il s'agit d'appliquer certaines techniques permettant d'insérer des transitions entre les briques afin d'éviter l'effet mosaïque. Autrement dit, ça permet de donner une allure hétérogène au niveau de la présentation du document.

Dans [Iksal, 2002] l'approche de composition de documents virtuels est basée sur l'usage de quatre ontologies : une ontologie de domaine, une ontologie de métadonnées, une ontologie d'utilisateurs et une ontologie de modèles. Ces ontologies sont utilisées pour composer et adapter le document par rapport à un utilisateur particulier.

La composition est basée sur l'utilisation d'un modèle prédéfini qui comporte un espace d'informations (un ensemble de fragments de documents) et au moins une structure narrative sous forme d'un graphe orienté acyclique. Les nœuds du graphe sont soit atomiques soit abstraits et doivent répondre aux besoins d'informations exprimés par l'utilisateur en se basant sur les concepts présents dans l'ontologie de domaine.

Une fois que le document virtuel est généré, une phase d'adaptation par rapport au profil de l'utilisateur est appliquée. Cette phase permet d'appliquer deux classes d'adaptation : l'adaptation de la présentation et l'adaptation de la navigation.

2.2. Hypermédias adaptatifs

Souvent les documents virtuels adaptatifs sont présentés sous forme d'un hypermédia aux lecteurs lorsqu'ils sont diffusés via le Web [Iksal, 2002]. Ceci sous-entend que plusieurs techniques ont été empruntées aux travaux sur les systèmes d'hypermédias adaptatifs. Dans la

suite nous allons présenter quelques définitions et principes relatifs à ces systèmes ainsi que quelques techniques de composition et d'adaptation proposées dans ce cadre.

2.2.1. Définitions et principes

La notion d'hypermédia est basée sur le fait que les documents à structure linéaire présentent plusieurs inconvénients, notamment le manque d'interactivité avec les utilisateurs. Ainsi, Ted Nelson [Nelson, 1965] (cité dans [Chorfi, 2007]) affirme que la principale caractéristique de l'hypertexte est de ne pas être séquentiel/linéaire. Il faut noter que cette affirmation reste valable pour les hypermédias qui, par rapport aux hypertextes, font référence à des éléments de contenu multimédia et non pas uniquement textuel.

Les systèmes hypermédia offrent un contenu constitué de fragments multimédia avec des liens entre des fragments au sein d'un même document ou des fragments qui font partie d'autres documents. Ces liens permettent à l'utilisateur d'interagir avec le système afin de construire son propre chemin de navigation au sein de l'hyperespace.

Cette structure a l'avantage d'offrir aux utilisateurs une liberté de navigation. Ainsi, l'utilisateur peut décider des fragments qui méritent d'être visités et dans quel ordre. Toutefois ceci peut provoquer certains problèmes cités plus haut, tels que la surcharge cognitive.

Les hypermédias adaptatifs ont été ainsi proposés pour garder d'une part les avantages offerts aux utilisateurs par les hypermédias et d'autre part pour remédier aux problèmes auxquels l'utilisateur peut faire face.

Brusilovsky définit un hypermédia adaptatif comme étant un ensemble d'items multimédias interconnectés dont certains aspects, qui sont visibles aux utilisateurs, sont adaptés automatiquement en se basant sur leurs profils [Brusilovsky, 1996].

2.2.2. Composition et adaptation

Afin d'introduire les principes de composition et d'adaptation dans le cadre des hypermédias adaptatifs nous allons prendre comme exemple illustratif le cas de l'approche InterBook [Brusilovsky et al., 1998].

Afin de mettre en place un système hypermédia adaptatif il faut disposer de deux types de connaissances : les concepts du domaine sujet d'étude et les spécificités des apprenants. La première connaissance est exprimée dans un modèle de domaine (domain model) et la deuxième dans un modèle de l'apprenant (student model).

Le modèle de domaine doit au mieux représenter les concepts du domaine ainsi que leurs relations. Ces relations entre concepts doivent refléter la structure pédagogique du domaine de connaissance couvert.

Le modèle de l'apprenant, quant à lui, doit donner une estimation du niveau de connaissance de l'apprenant par rapport à chaque concept du modèle. Ce modèle dit de recouvrement (overlay model) permet une mis-à-jour facile à base des traces de navigation de l'apprenant dans l'hypermédia. Le modèle de l'apprenant est complété par un ensemble d'objectifs d'apprentissage (learning goals). Il s'agit d'un ensemble de concepts qui doivent être acquis par l'apprenant.

L'hypermédia adaptatif doit être organisé en unités d'une façon hiérarchique. Le niveau terminal d'une hiérarchie correspond à une unité qui peut être une présentation, exemple, problème ou test. Chaque unité doit être décrite par rapport au modèle de domaine. Ainsi, à chaque unité correspond une liste de concepts. A chaque concept est attaché l'un des rôles suivants : pré-requis ou résultat. Si le concept a le rôle « résultat » alors il présente un concept traité par l'unité. Si le concept est un pré-requis alors il présente un concept qui doit être connu par l'apprenant pour comprendre le contenu de l'unité.

En se basant sur cette approche plusieurs techniques d'adaptation peuvent être utilisées. Une première technique qui peut être utilisée c'est la navigation avancée (advanced navigation). Il s'agit d'offrir un hyperespace bien structuré avec diverses méthodes de navigation. Toutefois, les liens affichés sont générés automatiquement par un module qui prend en considération l'état courant des connaissances de l'apprenant.

Une deuxième technique proposée est appelée guidage direct et support de navigation adaptative. Cette technique est basée sur la recommandation d'unité à base du profil de l'apprenant. Par rapport à un apprenant une unité peut être prête pour apprentissage, non prête pour apprentissage ou déjà apprise. D'autres indications visuelles offrent des indices à l'apprenant pour qu'il puisse trouver un chemin de navigation approprié.

Ces deux techniques d'adaptation permettent un guidage par objectif d'apprentissage (goal-driven guidance). Une autre technique d'adaptation guidée par le problème (problem-driven guidance) est également proposée. Dans ce cas le processus d'adaptation est déclenché dès que l'apprenant éprouve une difficulté à assimiler le contenu d'une unité. A ce moment l'apprenant peut demander de l'aide au système. Celui-ci propose une liste de liens qui peuvent aider l'apprenant à surmonter la difficulté éprouvée en se basant sur deux

informations. La première c'est les pré-requis des concepts traités par l'unité d'apprentissage en cours. En fait, cette information est disponible au niveau du modèle de domaine. La deuxième information concerne le niveau de l'apprenant par rapport aux pré-requis identifiés. Quant à elle, cette information est disponible au niveau du profil de l'apprenant. Les liens proposés pour aider l'apprenant à surmonter la difficulté éprouvée sont donc générés. En plus, ils sont classifiés en commençant par ceux jugés les plus prioritaires en allant vers ceux jugés les moins prioritaires

Au-delà de l'approche InterBook plusieurs méthodes d'adaptation ont été proposées. Le schéma suivant, proposé dans [Garlatti et Prié, 2004], résume les principales méthodes d'adaptation utilisées par les hypermédias adaptatifs.

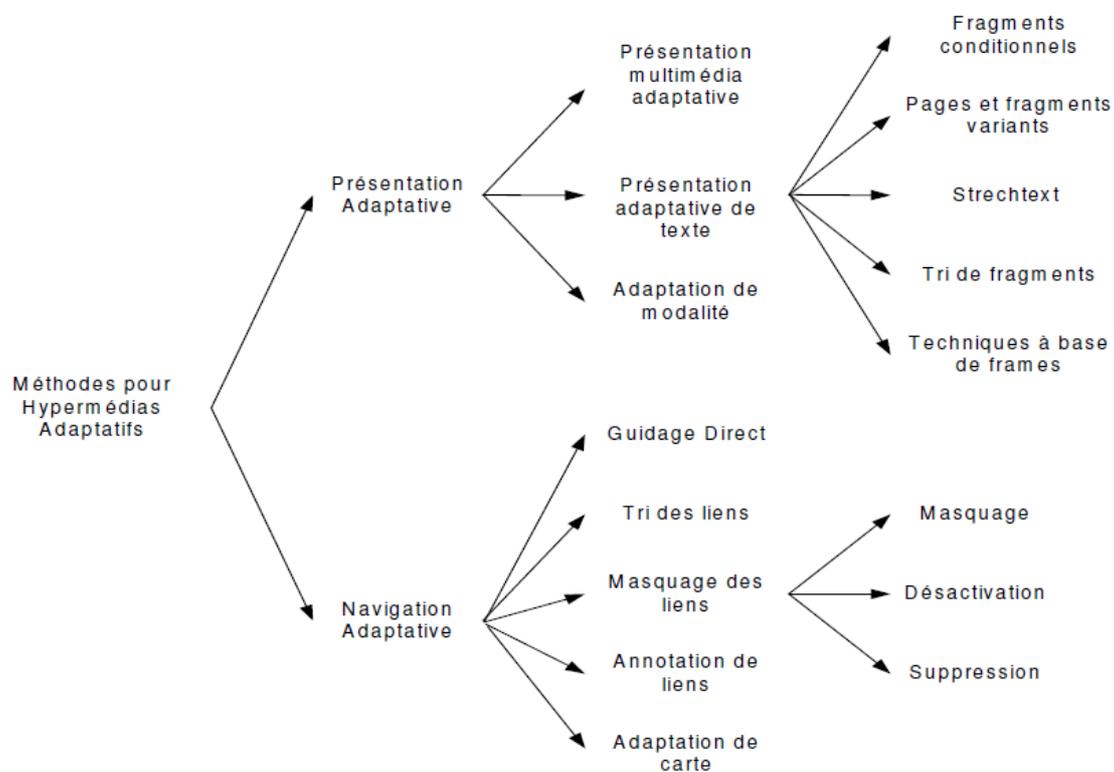


Figure 8 : Méthodes d'adaptation

2.3. Synthèse

L'utilisation des documents c'est confrontée à un certain nombre de problèmes dont la perte de la structure du document, la surcharge cognitive et la recherche des informations pertinentes, pour ne citer que ceux qui prennent particulièrement de l'importance dans le contexte de la formation en ligne.

Les documents virtuels adaptatifs et les hypermédias adaptatifs sont deux familles de documents numériques qui ont été proposées pour surmonter ces problèmes. Ils ont particulièrement en commun la capacité de s'adapter aux besoins, aux préférences et au niveau de connaissance de l'utilisateur. Plusieurs méthodes d'adaptations ont été utilisées soit individuellement ou combinées. Elles sont classifiées en deux familles : l'adaptation de la présentation et l'adaptation de la navigation.

Si l'aspect adaptatif est la caractéristique commune la plus significative aux documents virtuels adaptatifs et aux hypermédias adaptatifs, deux autres caractéristiques permettent de les discriminer. La première caractéristique est la durabilité. En effet, si les hypermédias adaptatifs sont persistants, les documents virtuels adaptatifs sont éphémères. Cette première caractéristique nous mène à la deuxième caractéristique qui est la nature de la construction du document. En effet, dans le cas des hypermédias adaptatifs le document est construit au préalable par un ou plusieurs auteurs, alors que pour les documents virtuels adaptatifs le document est auto-construit par assemblage de fragments au moment de la soumission d'une requête de la part de l'utilisateur.

La persistance et la construction au préalable de contenu s'apprêtent mieux en tant que caractéristiques au cas des objets d'apprentissage. C'est la raison pour laquelle les hypermédias adaptatifs sont souvent utilisés dans la formation en ligne. L'approche SIMBAD [Duitama, 2005] est l'une des approches qui s'est fortement inspirée des travaux sur les hypermédias adaptatifs et qui a proposé en particulier des extensions sémantiques pour une meilleure adaptation du contenu éducatif à l'apprenant.

3. Approche SIMBAD

3.1. Vue d'ensemble

L'approche SIMBAD se base sur la description sémantique comme un moyen pour promouvoir la réutilisation et la personnalisation des objets d'apprentissage. En fait, l'approche s'adresse à la fois aux auteurs d'objets d'apprentissage et aux apprenants. Pour les auteurs l'approche supporte l'identification des objets d'apprentissage et leur composition par réutilisation. Pour les apprenants l'approche supporte l'adaptation du contenu éducatif délivré. Il faut noter que plusieurs modes d'accès au contenu sont pris en compte : mode cours, mode concept et mode requête. En mode cours l'apprenant choisit le cours à suivre parmi une liste de cours disponibles. En mode concept l'apprenant choisit un ou plusieurs concepts et le

système doit trouver les cours qui couvrent ces concepts. Le mode requête permet à l'apprenant d'exprimer son objectif d'apprentissage sous forme d'une requête sans faire référence aux concepts. L'approche prévoit une phase d'adaptation de contenu éducatif par rapport aux préférences et au niveau cognitif de l'apprenant et ceci quelque soit le mode d'accès utilisé. Pour atteindre ces objectifs l'approche SIMBAD s'appuie sur trois modèles : le modèle de domaine, le modèle de l'apprenant et le modèle de description des objets d'apprentissage.

Trois niveaux de modélisation sont associés à l'approche et illustrés par la figure ci-dessous. Le premier niveau concerne la modélisation du domaine de connaissance à travers les concepts sous-jacents ainsi que la sémantique qui les relie les uns aux autres. Le deuxième niveau concerne la modélisation du contenu à travers la description de sa sémantique et notamment de sa structure en se basant sur le modèle de domaine. Le troisième niveau de modélisation concerne les apprenants. Il s'agit de décrire entre autres leur niveau de connaissances par rapport aux concepts définis au niveau du modèle de domaine afin de pouvoir supporter une adaptation du contenu.

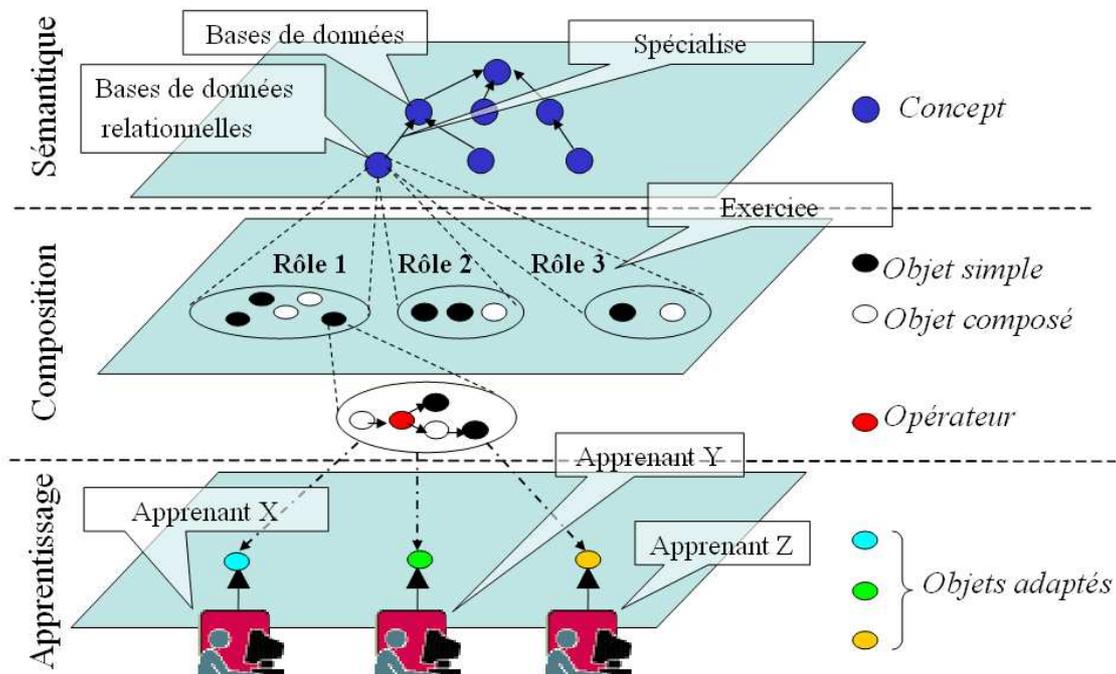


Figure 9 : Niveaux de modélisation de l'approche SIMBAD

3.2. Modèle de domaine

Le modèle de domaine consiste en une description de l'ensemble des concepts couverts par un domaine de connaissance ainsi que les relations entre eux. Les relations retenues sont les relations hiérarchiques « est plus générale » et « est plus spécifique » ainsi que les relations rhétoriques telles que « est antithèse de », « contraste avec », etc. Ces relations sont extraites de [Mann 1987]. La définition du modèle de domaine doit être faite par des experts.

L'objectif du modèle de domaine est d'offrir une vue commune et normalisée au niveau du système par rapport au domaine de connaissance relatif aux formations dispensées dans un cadre d'apprentissage donné. Le niveau de précision du modèle de domaine va influencer l'efficacité du système. En effet, si la communauté d'experts définit un modèle de domaine très détaillé alors les métadonnées sémantiques et les profils d'apprenants seront plus précis.

Au niveau conceptuel c'est une ontologie qui est utilisée pour définir le modèle de domaine. Rappelons qu'une ontologie est une spécification explicite d'une conceptualisation, c'est-à-dire une représentation, formelle ou non, de ce qui existe dans un domaine [Gruber 1993].

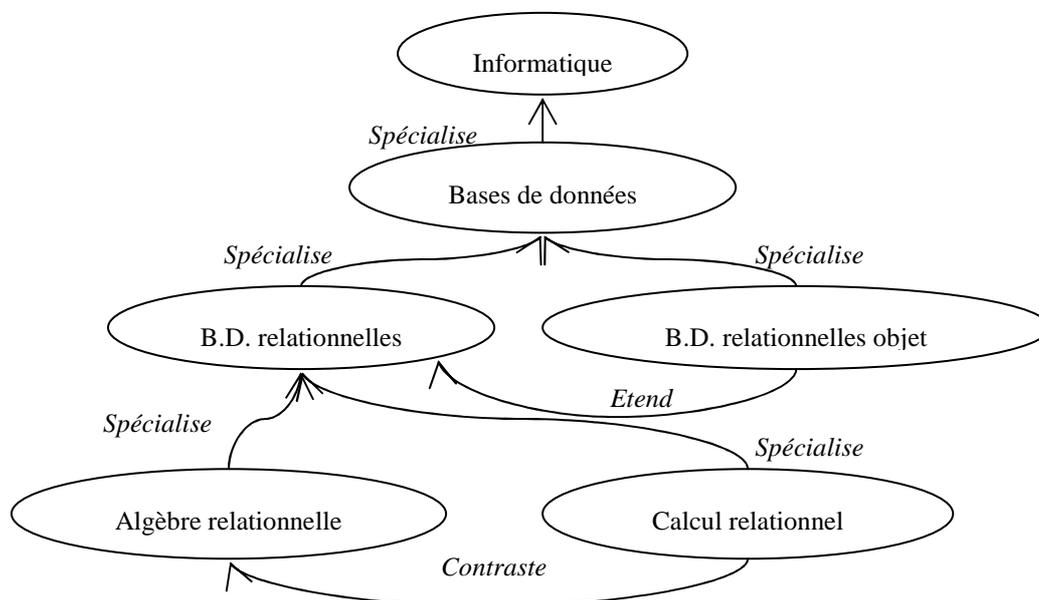


Figure 10 : Extrait d'un modèle de domaine

La figure ci-dessus représente un exemple d'une ontologie de domaine. Il faut noter que le recours aux ontologies est très fréquent dans les travaux sur la formation en ligne tels que [Stojanovic 2001], [Henze 2004], [Chorfi 2007] et [Cheniti 2008] pour en citer quelques uns.

3.3. Modèle de composition

Le modèle de composition définit la façon avec laquelle s'effectue l'assemblage des objets d'apprentissage existants afin d'en créer un nouveau. L'objet créé par assemblage est appelé « objet composé ». La structure d'un tel objet est décrite par un graphe orienté acyclique. Les sommets du graphe se déclinent en deux types, les sommets de type « objet d'apprentissage » et les sommets de type « opérateur ». Un sommet de type objet d'apprentissage peut être soit un objet d'apprentissage atomique, soit un objet d'apprentissage composé. Il peut être l'extrémité de plusieurs arcs, mais ne peut être que l'origine d'un seul. Le deuxième type de sommets concerne les opérateurs dont nous allons expliquer leurs rôles par la suite. Finalement, le graphe doit avoir un sommet racine unique et des feuilles, tous de type objets d'apprentissage.

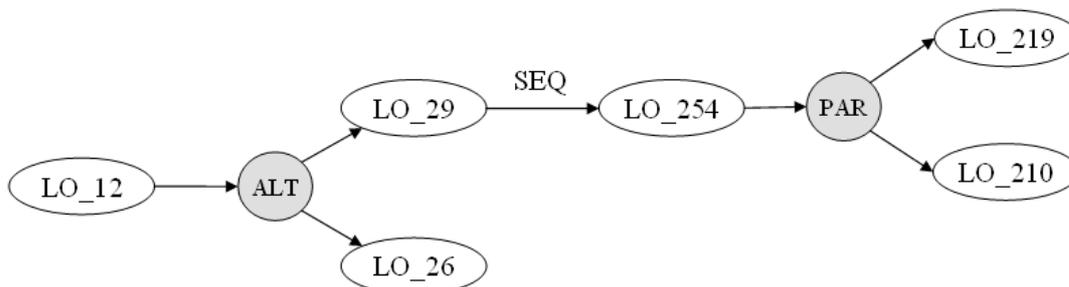


Figure 11 : Graphe de composition d'un objet d'apprentissage composé

Trois types d'opérateurs sont introduits par ce modèle de composition. Le premier opérateur est l'opérateur séquentiel « SEQ ». Il peut être l'extrémité de plusieurs arcs d'un graphe (au moins un) mais ne peut être à l'origine que d'un seul arc. Il permet d'exprimer un ordre d'affichage des objets d'apprentissage. Cet opérateur est considéré comme implicite puisque les arcs du graphe sont capables de véhiculer cette information. Par exemple, dans le cas du graphe ci-dessus les objets d'apprentissage « LO_29 » et « LO_254 » sont connectés par le biais d'un opérateur « SEQ », ceci veut dire que le premier objet, en l'occurrence « LO_29 », doit être affiché à l'apprenant avant l'objet « LO_254 ».

Le deuxième opérateur est l'opérateur parallèle « PAR ». Un sommet de type « PAR » doit être l'extrémité d'au moins un arc et l'origine d'au moins deux arcs. Il permet d'exprimer le

fait que les chemins, dont il est l'origine et dont la fin est le point de convergence ou les feuilles du graphe, peuvent être affichés à l'apprenant dans n'importe quel ordre. La décision sur l'ordre d'affichage est prise par l'apprenant via le système de gestion d'apprentissage utilisé. Dans le graphe donné comme exemple, nous avons un opérateur « PAR » suivi par deux chemins, chacun comporte un seul nœud de type objet d'apprentissage qui sont des feuilles. Ceci signifie qu'une fois que l'apprenant a accédé à l'objet « LO_254 » il a le choix soit d'accéder à l'objet « LO_219 » puis l'objet « LO_210 », ou l'inverse.

Le troisième opérateur est l'opérateur alternative « ALT ». Un sommet de type « ALT » doit être l'extrémité d'au moins un arc et l'origine d'au moins deux arcs. Il permet d'exprimer le fait que les chemins, dont il est l'origine et dont la fin est soit le sommet de convergence soit les feuilles du graphe, sont des alternatifs possibles pour la navigation dans l'objet d'apprentissage. Le choix d'un chemin plutôt qu'un autre est pris par le système de gestion de l'apprentissage à base du profil de l'apprenant (ses préférences et ses connaissances). Le graphe donné comme exemple contient un opérateur « ALT » qui est l'origine de deux chemins délimités par des sommets qui sont des feuilles. Une fois que l'apprenant accède à l'objet « LO_12 » le système lui affiche après soit le chemin qui contient l'objet « LO_26 » soit le chemin qui commence par « LO_29 ». Concrètement, « LO_26 » peut être une animation flash, alors que l'autre chemin alternatif contient du texte illustré par des images. Si le profil de l'apprenant contient une indication au niveau des préférences définissant que l'apprenant préfère les animations flash plutôt que le texte, le système de gestion de l'apprentissage va afficher à l'apprenant l'alternatif contenant l'objet « LO_26 ».

3.4. Modèle de l'apprenant

Le modèle de l'apprenant joue un rôle fondamental dans tout processus de personnalisation. Dans le cas du projet SIMBAD l'apprenant est décrit sous deux facettes : ses préférences et ses connaissances. Concernant les préférences se sont des informations factuelles modélisées sous forme de couples *<attribut, valeur>*. Parmi les attributs on peut citer le nom, le courriel, la langue préférée, etc. Quant aux connaissances elles sont décrites par rapport au modèle de domaine sous forme d'un triplet *<concept, rôle, niveau>*. Le concept fait référence à l'un des concepts du modèle de domaine. Le rôle est l'un des rôles éducatifs tels que : application, définition, introduction, etc. Finalement, le niveau est une valeur discrète qui peut prendre l'une des valeurs suivantes : très faible, faible, moyen, élevé et très élevé.

En fait, un apprenant peut avoir des niveaux différents par rapport à un seul concept selon le rôle. Par exemple, il peut avoir le niveau « moyen » par rapport au rôle « présentation » du concept « bases de données relationnelles » s’il a suivi avec succès un objet d’apprentissage dont la fonction d’acquisition comporte le triplet $\langle \text{bases de données relationnelles, présentation, moyen} \rangle$. Cependant, il peut avoir le niveau « faible » par rapport au rôle « exercice » concernant le même concept s’il a suivi un objet avec la fonction d’acquisition comportant le triplet $\langle \text{bases de données relationnelles, exercice, faible} \rangle$.

Ces informations seront utilisés par le système pour adapter et personnaliser le contenu par rapport aux préférences et au niveau de connaissance de l’apprenant.

4. Objets d’apprentissage dans SIMBAD

4.1. Modèle d’objet d’apprentissage

Dans le cadre de l’approche SIMBAD les objets d’apprentissage sont considérés comme des composants avec une interface décrivant leurs caractéristiques.

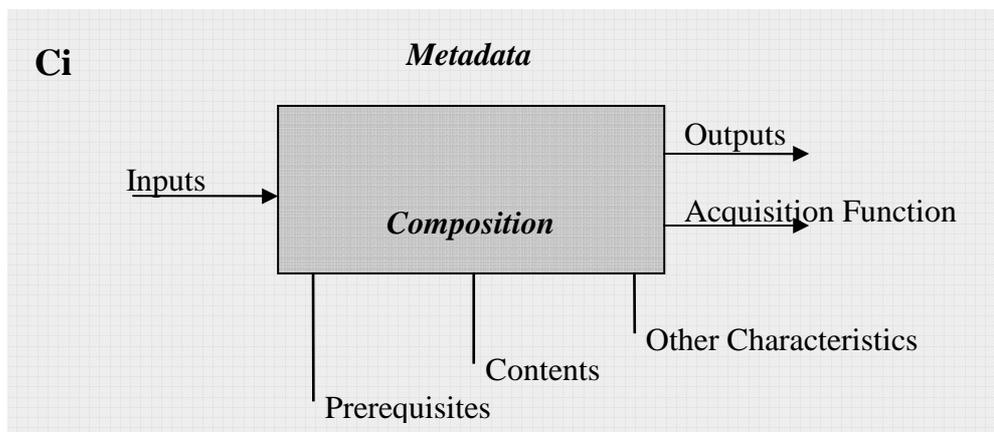


Figure 12 : Vue globale d'un objet d'apprentissage

La Figure 12 montre la façon avec laquelle un objet d’apprentissage est défini dans le projet SIMBAD. En fait, un objet d’apprentissage a un identifiant, des métadonnées et une éventuelle composition. Ainsi, il peut être décrit par le tuple suivant :

$\langle C_i, \text{metadata}, \text{composition} \rangle$
--

C_i est un identifiant unique de l’objet d’apprentissage. Il peut être un identifiant interne au système ou un URI.

La partie métadonnée comporte plusieurs catégories. La première est le contenu qui est décrit par rapport à un référentiel unique qui est le modèle de domaine. Le contenu est décrit par un ensemble non vide de tuples sous la forme suivante :

<domain-concept, role>

Concernant le « domain-concept », il s'agit d'un concept parmi ceux présents dans le modèle de domaine. Le rôle doit refléter ce qu'est l'objet d'apprentissage par rapport au concept.

La deuxième catégorie de métadonnées concerne les pré-requis. Ils sont décrits sous forme d'un ensemble de tuples de la forme suivante :

<domain-concept, role, knowledge-level>

Ici le « knowledge-level » représente le niveau requis par rapport au rôle du concept donné. Il s'agit de comparer ce tuple par rapport aux tuples présents dans le modèle de l'apprenant. Cette comparaison doit prendre en compte le fait que l'apprenant peut avoir un niveau supérieur au niveau de connaissances demandé.

La troisième catégorie concerne la fonction d'acquisition. Cette fonction permet de retourner, en se basant sur le comportement de l'apprenant, s'il y a « échec » ou « succès ». Dans le deuxième cas, une information supplémentaire est fournie pour indiquer le niveau de connaissances associé. Ainsi, le modèle de l'apprenant sera mis à jour, en cas de succès, de façon à ce que chaque tuple du contenu soit ajouté, s'il n'existe pas ou si le niveau de connaissance est amélioré.

La quatrième catégorie concerne les entrées (inputs). Il s'agit d'un ensemble de tuples permettant de paramétrer le déploiement de l'objet d'apprentissage s'il est paramétrable. Chaque tuple prend la forme suivante :

<operation, {message}, documentation>

La quatrième catégorie concerne les sorties (outputs). Il s'agit d'un ensemble de tuples qui prennent la même forme que celles des inputs et qui permettent à l'objet d'apprentissage de retourner des informations (par exemple, l'objet va retourner une évaluation des performances de l'apprenant comme un entier entre 0 et 10).

La cinquième catégorie concerne les autres caractéristiques. Il s'agit de métadonnées éducatives classiques. Le standard LOM [IEE 02] ou un profil d'application LOM peuvent être retenus.

4.2. Graphes de composition

Durant la phase de conception d'un objet d'apprentissage, une fois que sa composition est établie, l'auteur a une vision de l'objet que nous appelons le *graphe abstrait* [Farhat 2004]. Celui-ci peut contenir des objets d'apprentissage complexes. Ces derniers, peuvent à leur tour récursivement contenir des objets d'apprentissage complexes.

Pour que le système puisse déterminer le contenu qui doit être généré une fois que l'objet d'apprentissage est demandé, il doit instancier chaque objet d'apprentissage complexe jusqu'à produire un graphe qui n'est composé que d'objets d'apprentissage simples et d'opérateurs. Il s'agit du *graphe concret*.

Le graphe abstrait, le graphe concret et les graphes intermédiaires reflètent tous les différentes vues d'un même graphe de composition d'un objet d'apprentissage donné.

Par exemple, soit un objet d'apprentissage dont l'identifiant est LO_400 et qui est décrit par le graphe abstrait suivant :

$$LO_{400} = LO_{12} \text{ ALT } (LO_{350}, LO_{26})$$

Supposons que LO_350 est un objet d'apprentissage complexe ayant le graphe abstrait suivant :

$$LO_{350} = LO_{29} \text{ SEQ } LO_{271}$$

LO_271 est un LO complexe décrit par le graphe abstrait suivant :

$$LO_{271} = LO_{254} \text{ PAR } (LO_{219}, LO_{210})$$

Alors LO_400 aura le graphe concret suivant :

$$LO_{400} = LO_{12} \text{ ALT } (LO_{29} \text{ SEQ } LO_{254} \text{ PAR } (LO_{219}, LO_{210}), LO_{26})$$

La figure ci-dessous montre une illustration graphique pour cet exemple.

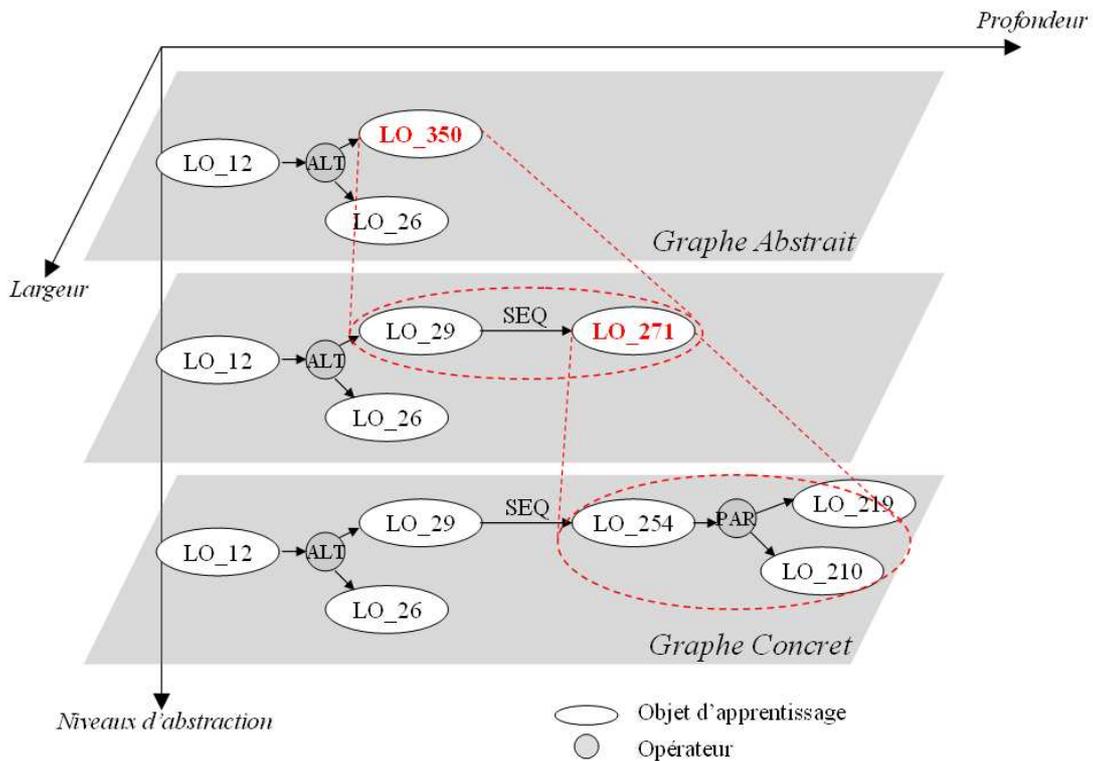


Figure 13 : Différents points de vue d'un graphe de composition

Un graphe concret peut être parcouru de plusieurs façons, chacune est appelée *graphe affichable*. Ils représentent les différentes possibilités de parcours du graphe concret et les différentes façons avec lesquelles le contenu peut être affiché aux apprenants. Le choix d'un graphe affichable parmi d'autres est fait par le système de gestion de l'apprentissage et dépend du profil de l'apprenant. Dans le cas de l'exemple précédant voici les graphes affichables possibles :

```

CC (LO_400) = {
    LO_12, LO_26;
    LO_12, LO_29, LO_254, (LO_219; LO_210);
}

```

Pour générer le graphe de composition concret nous avons défini l'algorithme suivant :

Soient :

C est l'ensemble d'objets d'apprentissage composés.

$root : C \rightarrow O$, est la fonction qui associe à tout objet d'apprentissage composé, l'objet d'apprentissage qui est la racine de son graphe de composition.

E_i est l'ensemble des objets d'apprentissage qui représentent les sommets du graphe de composition de l'objet d'apprentissage composé lo_i .

V_i est l'ensemble des sommets appartenant au graphe de composition de lo_i tel que : $lo_i \in C$.

T_i est l'ensemble des transitions du graphe de composition de l'objet d'apprentissage lo_i tel que : $lo_i \in C$.

t_{kj} tel que $t_{kj} \in T_i$ est la transition dont la source est le sommet lo_k et l'extrémité est le sommet lo_j tel que : $\{lo_k, lo_j\} \subset V_i$

Soit $lo_i \in C$. Le graphe abstrait de lo_i est défini par le triplet $\langle T_i, V_i, r \rangle$ tel que r est le sommet racine du graphe de composition de l'objet composé lo_i . Le graphe concret, défini par le triplet $\langle T'_i, V'_i, r' \rangle$ tel que r' est le sommet racine du graphe de composition concret, est calculé de la façon suivante :

Algorithme 1 : Algorithme de génération du graphe concret

Debut

// Déterminer le sommet racine r' du graphe de composition concret

Si ($r \in C$) Alors

$r' \leftarrow root(r)$

Sinon

$r' \leftarrow r$

Fin si

$T'_i \leftarrow T_i$

$V'_i \leftarrow V_i$

// Substitution des objets d'apprentissage composés par leurs compositions

Tant que ($\exists j \in C \cap V'_i$) Faire

// Etape 1: Ajouter les nouvelles transitions et sommets

$T'_i \leftarrow T'_i \cup T_j$

$V'_i \leftarrow V'_i \cup V_j$

// Etape 2: Connexion du graphe de l'objet j au graphe de composition

$r'' \leftarrow \text{root}(j)$

$T'_i \leftarrow T'_i \cup (\bigcup \{t_{z''}\} / z \in V'_i, \exists t_{z'} \in T'_i)$ // Connexion du début du graphe

$T'_i \leftarrow T'_i \cup (\bigcup \{t_{ey}\} / e \in E_j, y \in V'_i, \exists t_{jy} \in T'_i)$ // Connexion de la fin

// Etape 3: Déconnexion de l'objet j

$T'_i \leftarrow T'_i \setminus (\bigcup \{t_{zj}\} / z \in V'_i, \exists t_{zj} \in T'_i)$ // transitions dont j est l'extrémité

Si ($\exists t_{jy} \in T'_i$) Alors

$T'_i \leftarrow T'_i \setminus \{t_{jy}\}$ // transition dont j est la source

Fin si

// Suppression du sommet j

$V'_i \leftarrow V'_i \setminus \{j\}$

Fin tant que

Fin

5. Problématiques

5.1. Conception d'objets d'apprentissage composés

Certes l'approche SIMBAD offre les moyens nécessaires pour rendre effectif d'une part la réutilisation des objets existants et d'autre part la personnalisation de l'objet composé suivant les profils des apprenants. En effet, l'approche offre en plus des métadonnées conventionnelles des métadonnées sémantiques. De plus, grâce aux opérateurs les possibilités de connexion et de structuration de l'objet sont multiples et variées.

Cependant, cette approche ainsi que les approches similaires sont souvent difficiles à faire adopter par les auteurs et les concepteurs d'objets d'apprentissage. Les raisons derrière cette

résistance peuvent se résumer en deux facteurs. Le premier c'est le manque de motivations pour convaincre les auteurs d'adopter ces approches en adaptant leurs méthodes de travail. Le deuxième facteur réside dans le fait que dès que la taille de l'objet composé dépasse une certaine limite l'auteur a de la difficulté à garantir certaines propriétés, voire même à bien appréhender toutes les facettes de cet objet (structurelles et sémantiques).

5.2. Assistance et guidage

Adopter la réutilisation par composition comme moyen pour concevoir et développer des nouveaux objets d'apprentissage est un défi malgré les avantages qui en découle au niveau institutionnel et démontrés par plusieurs travaux tels que [Downes 2001]. Toutefois, ceci oblige les auteurs à adapter leurs méthodes de travail et à faire l'effort de comprendre d'une part les possibilités offertes et d'autre part les contraintes à respecter.

Pour surmonter cette difficulté il est important à notre avis de proposer une approche d'assistance. Celle-ci doit offrir d'une part une démarche logique et claire pour la conception d'objets d'apprentissage par composition. Il est également important de décharger dans la mesure du possible l'auteur des tâches qui peuvent être automatisées.

5.3. Vérification de la cohérence

Un autre facteur pouvant compromettre l'adoption de la conception par composition d'objets d'apprentissage est le risque d'incohérence. En effet, il est possible que l'assemblage donne naissance à un objet comportant des anomalies au niveau sémantique ou au niveau structurel. Un exemple d'une incohérence sémantique c'est le cas où un objet B doit être délivré à l'apprenant après un objet A alors qu'il traite un concept qui figure parmi les pré-requis nécessaires pour la compréhension des concepts traités par l'objet A. Un autre exemple d'incohérence structurelle est celui où un graphe contient une feuille de type opérateur.

La vérification de la cohérence de l'objet composé par l'auteur peut s'avérer une tâche difficile en particulier dans les deux cas suivants. Le premier cas c'est lorsque le graphe de composition atteint une certaine taille, à ce moment la vérification manuelle de la cohérence devient inévitablement difficile et pénible sans la garantie que la vérification soit complète et valide. Le deuxième cas c'est lorsque l'incohérence n'est visible qu'au niveau du graphe de composition concret ou d'un graphe intermédiaire.

Pour recourir à cette problématique il faut que l'approche d'assistance soit capable d'analyser et de vérifier automatiquement la cohérence sémantique et structurelle d'un objet composé. Il

est également souhaitable à ce niveau de calculer et de fournir des informations supplémentaires permettant à l'auteur d'avoir une vue d'ensemble sur l'objet qu'il vient de composer. En effet, des informations implicites déduites automatiquement peuvent aider l'auteur à découvrir des facettes cachées de l'objet en particulier la vue système (graphe de composition concret) et la vue apprenant (graphes affichables). Ceci doit offrir aux auteurs et concepteurs des objets d'apprentissage composés des éléments nécessaires pour améliorer et pour raffiner leurs objets suivant un processus itératif et incrémental.

5.4. Métadonnées

Renseigner les métadonnées pour un objet d'apprentissage est certainement une tâche pénible. En effet, dans le cas du standard LOM de l'IEEE [IEEE 2002] on se trouve avec soixante seize éléments classés en neuf catégories. Ceci doit démotiver les auteurs pour remplir ce nombre élevé d'informations.

Ainsi, il est préférable de pouvoir contribuer à cette tâche en exploitant les informations disponibles pour contribuer d'une façon automatique à la génération des métadonnées éducatives de l'objet composé. Pour y parvenir il est possible d'utiliser d'une part les métadonnées éducatives des objets réutilisés et d'autre part les informations explicites et implicites propres à l'objet composé.

5.5. Etude de conformité

Il est d'usage qu'une organisation ou une institution éducative définissent des règles ou des directives ou encore des lignes directrices qui portent sur la conception des objets d'apprentissage. Certains des aspects évoqués dans ce contexte sont mesurables et calculables automatiquement. Par exemple il est possible de vérifier automatiquement si une instance de métadonnées éducatives est conforme ou pas à un profil d'application LOM donné. D'autres aspects relatifs à la sémantique, à la structure ou mixtes doivent pouvoir être vérifiés d'une façon automatique.

6. Conclusion

Après avoir passé en revue des approches de composition et de personnalisation de documents numériques, nous avons présenté l'approche SIMBAD. L'accent a été mis sur le modèle d'objets d'apprentissage. Nous nous sommes intéressés en particulier au cas des objets composés par réutilisation d'autres objets. La dernière partie a été réservée à l'énoncé de la

problématique. Celle-ci a été détaillée tout en précisant les différents aspects qui méritent d'être pris en compte afin de faciliter la conception et le développement des objets d'apprentissage composés par réutilisation d'autres objets.

Dans le chapitre suivant nous allons introduire notre approche d'assistance pour les auteurs d'objets d'apprentissage composés. Cette approche doit tenir compte des différents aspects évoqués lors de l'analyse de la problématique. En effet, elle doit offrir les moyens nécessaires pour aider les auteurs à chercher, composer, analyser, annoter et vérifier la conformité lors de la phase de conception et de développement d'un objet d'apprentissage composé.

Chapitre III :

VUE D'ENSEMBLE DE NOTRE APPROCHE D'ASSISTANCE

1. Introduction

Dans ce chapitre nous allons présenter notre approche d'assistance aux auteurs des objets d'apprentissage composés de type SCORM ou SIMBAD. Cette approche comporte cinq phases d'assistance et d'aide aux auteurs, à savoir : l'assistance à la recherche des objets d'apprentissage, l'assistance à la composition, l'assistance à la compréhension, le calcul automatique des métadonnées éducatives et la vérification de la conformité à des patrons de conception. Après avoir donné la vue d'ensemble de notre approche, nous allons détailler les phases de recherche des objets à réutiliser et de calcul automatique des métadonnées. Nous terminons le chapitre par la présentation d'un algorithme de transformation d'un objet SCORM en un équivalent SIMBAD ce qui nous permet d'appliquer notre approche à un plus grand nombre d'objets et d'étendre la portée pratique de nos contributions.

2. Vue d'ensemble

2.1. Présentation

Nous proposons d'adopter une approche assistée de conception et de développement d'objets d'apprentissage composés [Farhat et al., 2009]. Cette approche comporte cinq phases. Lors de la première phase l'approche propose des mécanismes de recherche et de découverte d'objets d'apprentissage candidats à la réutilisation. L'auteur doit choisir un ensemble d'objets qui doivent couvrir le ou les concepts à traiter (ex. algèbre relationnelle) et les rôles éducatifs à couvrir (ex. exercice, exemple, description, introduction). Il peut ensuite passer à la deuxième phase dans laquelle il va concevoir le graphe de composition abstrait en utilisant les objets retenus lors de la première phase et les opérateurs d'assemblage. Suite à cette phase l'auteur peut revenir à la phase deux s'il souhaite faire des modifications sur l'ensemble des objets à réutiliser. Sinon, il passe à la phase trois dans laquelle une analyse automatique est effectuée sur l'objet créé afin de permettre à l'auteur de mieux comprendre le résultat de la composition.

L'analyse effectuée au niveau de la troisième phase a pour objectif de générer une cartographie détaillée sur l'objet en question. Il s'agit d'offrir à l'auteur des indicateurs divers et variés qui vont lui permettre d'avoir une meilleure vue sur les différentes facettes de l'objet d'apprentissage en cours de conception. En particulier, il aura une analyse de la vue système (graphe de composition concret) et de la vue apprenant (graphe affichable).

Une fois que les analyses sont satisfaisantes l'auteur passe à la quatrième phase. Celle-ci permet d'enrichir les métadonnées déjà introduites par l'auteur par des métadonnées complémentaires calculées automatiquement en se basant sur les métadonnées éducatives des objets utilisés dans la composition de l'objet.

La cinquième phase consiste à confronter l'objet d'apprentissage composé à des patrons de conception. Ceux-ci permettent de décrire certains critères structurels et sémantiques recherchés. La vérification de conformité doit permettre de détecter la nature et les causes de la non-conformité si c'est le cas. L'auteur sera donc en mesure de revenir à l'une des phases précédentes pour apporter les modifications nécessaires permettant à l'objet de s'aligner avec les directives. Tout ceci se déroule d'une façon itérative et incrémentale vers la mise au point de l'objet d'apprentissage composé.

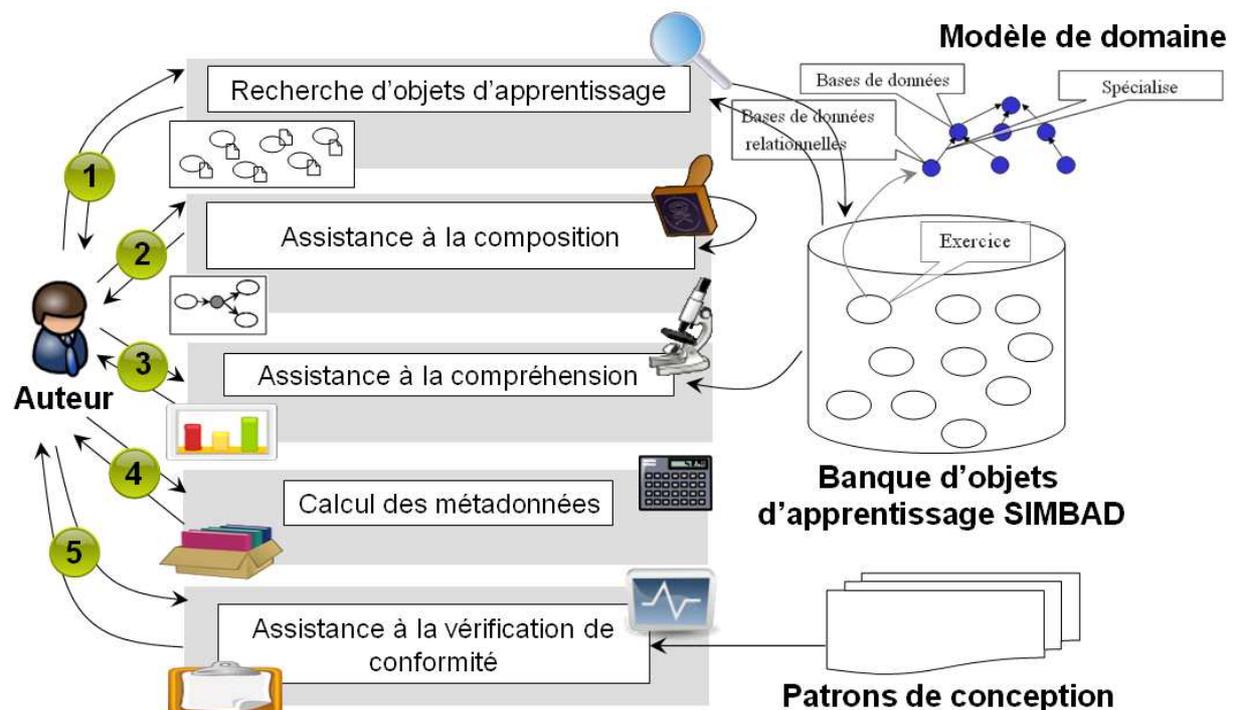


Figure 14 : Vue d'ensemble de notre approche d'assistance

La figure ci-dessus résume les différentes phases de notre approche d'assistance des auteurs lors de la conception d'objets d'apprentissage composés.

2.2. Déroulement de l'assistance

Le déroulement de processus d'assistance peut prendre plusieurs formes et variantes (prise en compte de toutes les phases ou non). Le diagramme d'activité suivant présente un scénario d'application de notre approche d'assistance :

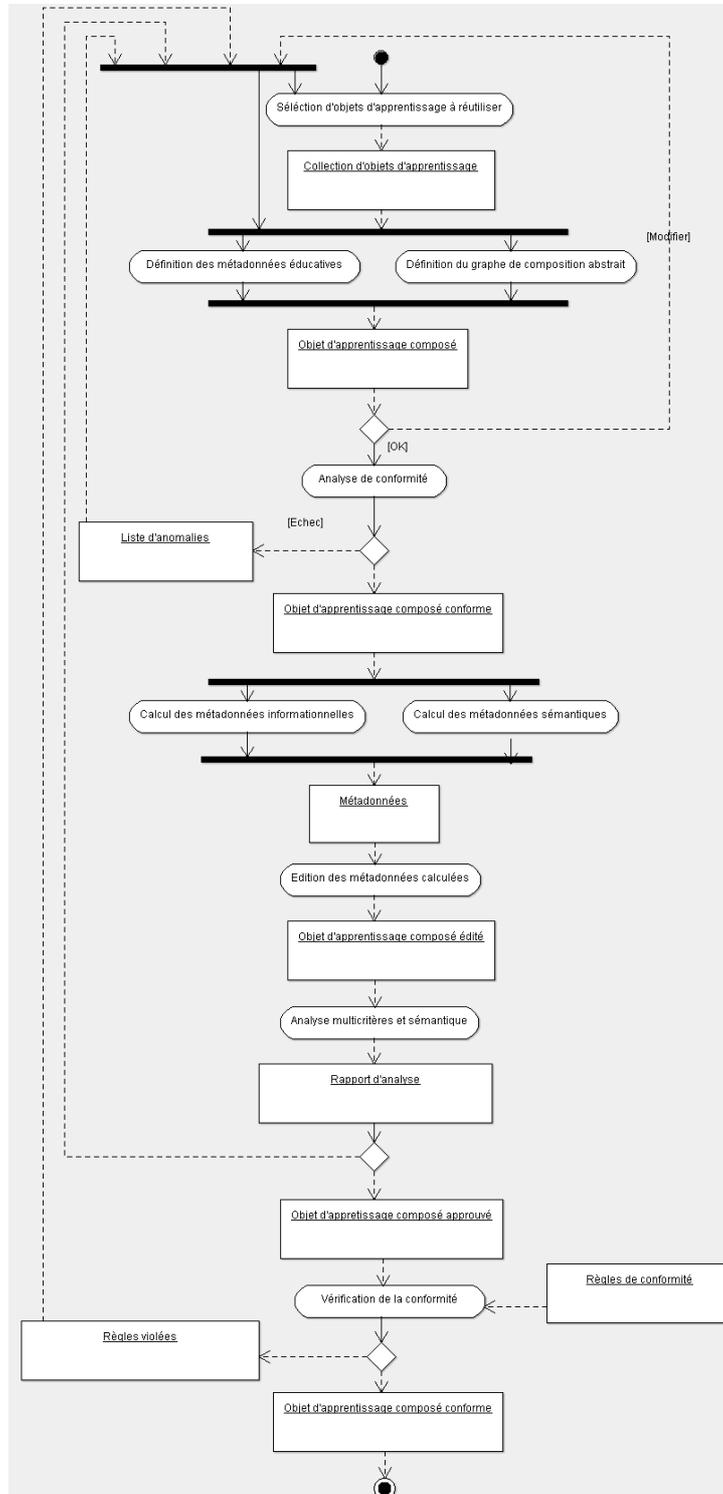


Figure 15 : Diagramme d'activité relatif à l'approche d'assistance

3. Présentation des phases recherche et calcul de métadonnées

Dans cette thèse nous allons investiguer principalement les phases vérification de la cohérence, de l'analyse et de la vérification de la conformité. Toutefois, nous allons présenter certains aspects relatifs aux phases de recherche et de calcul automatique de métadonnées.

3.1. Assistance à la recherche

Lors de la conception d'un objet d'apprentissage composé il s'agit de réutiliser des objets d'apprentissage existant et de les combiner ensemble suivant une certaine logique. Ces objets sont traditionnellement stockés dans des banques d'objets d'apprentissage. Ces banques offrent le plus souvent un moteur de recherche permettant à l'utilisateur de trouver les objets d'apprentissage qui répondent à certains critères.

En effet, lors de l'ajout d'un objet d'apprentissage au niveau d'une banque d'objets d'apprentissage les métadonnées sont extraites et sauvegardées. Ainsi, la résolution de toute requête de recherche se base sur les informations disponibles au niveau des métadonnées éducatives des objets d'apprentissage.

Dans le cas de l'approche SIMBAD les métadonnées éducatives comportent en particulier des indications sur la sémantique et la structure de l'objet en plus des métadonnées du standard LOM. Comme il a été avancé dans la partie introductive de l'approche SIMBAD, les métadonnées sémantiques concernent en particulier le contenu, les pré-requis nécessaires et la fonction d'acquisition permettant de mettre à jour le profil de l'apprenant qui a validé le contenu de l'objet.

Nous définissons deux modes de recherche : la recherche intentionnelle et la recherche extensionnelle. La recherche intentionnelle est celle qui est utilisée principalement lors de la première itération lors de la conception d'un objet d'apprentissage composé. L'objectif est de repérer les objets d'apprentissage à réutiliser.

La requête est formulée via la description de l'intention de l'auteur en termes de contenu de pré-requis et de fonction d'acquisition.

Ainsi, l'auteur peut formuler ses requêtes pour récupérer les objets à réutiliser. Principalement, cette recherche est guidée par la sémantique. En effet, l'auteur fixera en premier lieu le concept auquel il s'intéresse. Ensuite, il précisera à chaque fois le rôle de l'objet recherché et son niveau de complexité. Il peut également faire entrer en jeu d'autres paramètres tels que le temps d'apprentissage, le média, la langue, la taille, etc.

Toutefois, nous prenons comme hypothèse que l'auteur a accès à un large corpus d'objets d'apprentissage disposant des métadonnées éducatives nécessaires pour assurer leur accessibilité. Cette hypothèse est à notre avis tout à fait réaliste en partant du principe qu'un grand nombre de ressources éducatives est déjà accessible sur Internet. D'autre part, nous soulignons que de plus en plus d'initiatives d'interconnexion et de partage entre banques d'objets d'apprentissage sont en train de voir le jour, tels que eduSource⁶ ou VLORN⁷. Cependant, au cas où l'auteur se trouve avec des objets qui manquent il peut les ajouter lui-même. Ceci va permettre d'une part à l'auteur de ne s'investir que dans un contenu qui n'existe pas, ce qui présente un gain considérable en termes de temps. Et d'autre part, ceci doit encourager les auteurs à contribuer, et d'une façon distincte, à deux niveaux de granularité. La contribution à granularité fine consiste principalement en un effort de production de contenu sous forme d'exercice, de schéma illustratif, de démonstration, etc. La contribution à granularité large consiste principalement en un effort d'ingénierie pédagogique en organisant des éléments de contenu d'une façon particulière afin de permettre à l'apprenant d'atteindre de la façon la plus efficace possible un objectif pédagogique.

3.2. Calcul de métadonnées

Malgré l'importance des métadonnées éducatives, les renseigner est considéré par les auteurs comme l'une des tâches les plus fastidieuses à accomplir. Ceci peut être expliqué par le nombre important de catégories et d'entrées dans les profils d'application LOM. Dans le cas de SIMBAD la tâche est encore plus difficile dans la mesure où il faut également renseigner les métadonnées sémantiques.

Dans le cadre de notre approche nous proposons d'assister l'auteur grâce au calcul automatique des métadonnées éducatives. En fait, dans le cas des objets d'apprentissage composés nous disposons déjà de certaines informations concernant les métadonnées grâce à la collecte et l'utilisation des métadonnées des objets qui le forment.

Il faut noter que certains champs de métadonnées peuvent facilement être calculés. C'est le cas par exemple des mots clés qui peuvent être générés en appliquant la formule suivante : les mots clés de l'objet d'apprentissage composé sont l'union des mots clés donnés par l'auteur, et les mots clés des objets d'apprentissage réutilisés.

⁶ <http://edusource.netera.ca>

⁷ <http://pre2005.flexiblelearning.net.au/projects/resources/2005/vlorn/>

Toutefois le calcul de certaines métadonnées est plus compliqué. C'est le cas du calcul automatique des pré-requis. Rappelant que les pré-requis sont décrits sous la forme d'un triplet $\langle \text{Contenu}, \text{Rôle}, \text{Niveau} \rangle$. Pour parvenir à produire automatiquement les pré-requis nous avons opté pour une approche ascendante par rapport au graphe de composition abstrait. Nous allons utiliser la notation suivante :

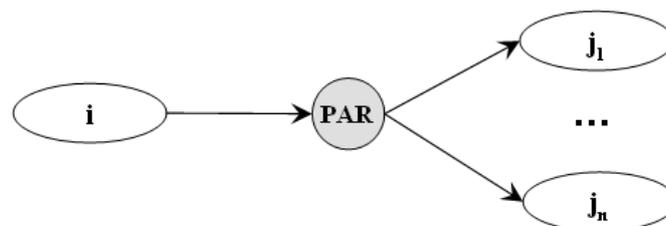
- P_k est l'ensemble de triplets $[c, r, l]$ (c: un concept, r: un rôle, l: un niveau) qui représente les pré-requis d'un objet d'apprentissage k.
- A_k est l'ensemble de triplets $[c, r, l]$ (c: un concept, r: un rôle, l: un niveau) qui représente la fonction d'acquisition de l'objet d'apprentissage k.

Pour calculer l'ensemble P_k il faut commencer à partir des feuilles du graphe de composition concret pour remonter vers la racine. Pour chaque itération de l'algorithme de parcours on peut se trouver dans l'un des trois cas :

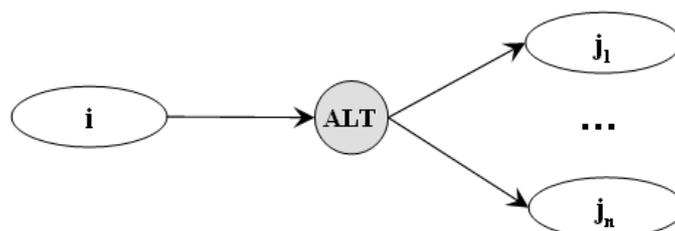
- Le premier cas correspond à la situation où deux objets i et j soient reliés dans le graphe de composition par un opérateur SEQ :



- Le deuxième cas correspond à la situation où l'objet i précède plusieurs objets auxquels il est attaché via l'opérateur PAR :



- Le troisième cas correspond à la situation où l'objet i précède plusieurs objets auxquels il est attaché via l'opérateur ALT :



L'objectif est de calculer à chaque étape l'ensemble P'_i tel que i est le sommet de type objet d'apprentissage en cours d'analyse. Le dernier sommet i est la racine du graphe de composition de l'objet k .

L'ensemble des pré-requis de l'objet d'apprentissage composé k (noté P'_k) est l'union de l'ensemble des pré-requis renseignés par l'auteur (P_k) et de l'ensemble des pré-requis calculés automatiquement (P'_i) :

$$P'_k = P_k \cup P'_i$$

Le calcul démarre au niveau des feuilles du graphe en remontant vers la racine. Pour les sommets feuilles il faut noter que $P'_i = P_i$, alors pour le reste des sommets il faut appliquer l'une des formules suivantes selon le cas.

Pour le premier cas de figure (l'opérateur est SEQ) l'ensemble des pré-requis est formé par l'ensemble des pré-requis de l'objet i auxquels il faut ajouter les pré-requis de l'objet j qui ne sont pas déjà satisfaits par la fonction d'acquisition de l'objet i :

$$P'_i = P_i \cup (P'_j - A_i)$$

Pour le deuxième cas de figure (l'opérateur est PAR) et pour le troisième cas de figure (l'opérateur est ALT) l'ensemble des pré-requis est formé par l'ensemble des pré-requis de l'objet i auxquels il faut ajouter les pré-requis de tous les objets suivants et qui ne sont pas déjà satisfaits par la fonction d'acquisition de l'objet i :

$$P'_i = P_i \cup \left(\bigcup_{x \in [1..n]} P'_x - A_i \right)$$

Le calcul des pré-requis s'arrête suite au calcul de l'ensemble P'_i lorsque i est le nœud racine.

Il faut noter également que nous proposons que le calcul automatique des métadonnées soit effectué suite à l'analyse sémantique et structurelle de l'objet composé. Rappelons que l'analyse sémantique est basée sur une utilisation des métadonnées sémantiques. Nous préférons que ces métadonnées soient celles renseignées explicitement par l'auteur. Ainsi, toute incohérence sémantique viendra soit de la façon avec laquelle l'objet a été assemblé (l'ordre suivant lequel les objets réutilisés ont été interconnectés), soit à cause d'un renseignement imprécis, erroné ou incomplet de la part de l'auteur. Il nous semble que calculer automatiquement les métadonnées avant cette phase peut masquer certains défauts alors que l'objectif est de justement les mettre en évidence. Une fois que l'objet est cohérent et qu'il respecte les règles qui seront énoncées dans le chapitre suivant, le processus de calcul

automatique de métadonnées permettra d'enrichir davantage la description de l'objet en profitant des métadonnées des objets réutilisés. Ces métadonnées seront présentées à l'auteur pour vérification, éventuellement modification et validation. Ceci présente un gain de temps et une simplification de la tâche par rapport à un renseignement explicite par l'auteur de toutes les métadonnées.

4. Transformation d'objets SCORM en objets SIMBAD

4.1. Objectif de la transformation

Actuellement la spécification SCORM⁸ est considérée comme un standard de facto [Ostyn, 2003]. Il nous semble important de pouvoir transformer des objets SCORM en des objets SIMBAD. Ainsi, tout ce que nous allons définir pour SIMBAD sera donc applicable aux objets SCORM. En particulier, il sera possible d'appliquer notre approche d'assistance afin de composer un nouvel objet par réutilisation d'objets SCORM transformés en objets SIMBAD.

Afin de pouvoir étudier la faisabilité de cette transformation il est nécessaire d'étudier les spécificités liées principalement à la navigation dans un contenu SCORM. Ceci doit nous permettre de définir un algorithme pour la transformation. En fait, cet algorithme doit pouvoir permettre de générer le graphe de composition abstrait de l'objet SCORM décrit conformément à l'approche SIMBAD. Concernant les métadonnées il n'y a pas de transformations particulières à faire. En fait, SCORM ne prévoit pas l'utilisation de métadonnées sémantiques et donc nous allons considérer qu'elles ne sont pas renseignées. Quant aux métadonnées éducatives informationnelles elles ne nécessitent pas de transformations particulières puisque SIMBAD utilise un profil d'application LOM [IEEE, 2002].

Il faut noter également que la version SCORM 2004 permet l'utilisation de la spécification IMS Simple Sequencing [IMS, 2003b] [ADL, 2004d]. Celle-ci est sensée permettre d'offrir à l'auteur les moyens nécessaires pour mettre en place certaines stratégies d'apprentissage. Le terme « Simple » fait référence au fait que le modèle ne prend en compte qu'un nombre limité de ces stratégies. Celles-ci doivent influencer la façon avec laquelle l'objet est parcouru. Toutefois, nous allons nous contenter des informations relatives au modèle d'agrégation de contenu IMS Content Packaging [IMS, 2004] [ADL, 2004b]. En fait, nous avons constaté que

⁸ <http://www.adlnet.gov/Technologies/scorm/default.aspx>

l'IMS Simple Sequencing connaît des difficultés à se faire adopter vu la complexité de sa mise en œuvre et le nombre important de paramètres à fixer qu'elle comporte, et ceci sans toutefois qu'elle soit capable de supporter une large variété de stratégies d'apprentissage. Bien que certains systèmes annoncent leur conformité à SCORM 2004, nous n'avons pas trouvé (en Juin 2009) un système de gestion d'apprentissage qui supporte le « simple sequencing ». Ceci s'explique par le fait qu'il existe trois niveaux de conformité pour ces systèmes [ADL 2009]. La majorité des systèmes sont conformes au premier niveau (LMS RTE 1.0), alors qu'il faut être conforme au troisième niveau (LMS SN 1.0) pour pouvoir supporter le « simple sequencing ».

Ainsi, nous allons tout d'abord étudier les subtilités liées au modèle d'agrégation de contenu du standard SCORM. Puis nous allons proposer notre algorithme de transformation d'un objet SCORM en un objet SIMBAD équivalent.

4.2. Modèle d'agrégation de contenu

La spécification SCORM offre à l'auteur la possibilité de décrire plusieurs parcours d'un objet d'apprentissage en se basant sur la spécification de paquetage de contenu d'IMS [IMS 2004]. Cette spécification offre à l'auteur la possibilité de définir plusieurs parcours, au moins un, appelés organisations. Chaque organisation comporte une suite d'éléments qui se déclinent en deux catégories. La première catégorie concerne les éléments de type contenu alors que la deuxième concerne les conteneurs.

Les éléments contenus se déclinent à leur tour en deux sous catégories. D'une part on a les « assets » et d'autre part on a les SCO (Sharable Content Object). Les « assets » sont des média élémentaires qui peuvent être visualisés par un navigateur Web. Par exemple les fichiers HTML, les documents PDF, les animations flash et les applets Java sont considérés comme étant des « assets ». La deuxième sous catégorie de contenu est celle des SCO. Il s'agit d'un ou de plusieurs « assets » capables de communiquer avec un système de gestion de l'apprentissage conforme à SCORM et qui ne contient pas de liens vers un autre objet d'apprentissage. Concrètement, il faut rattacher à la ressource un code JavaScript permettant la communication avec l'API SCORM [ADL 2004c]. En fait, les SCO à leur tour se déclinent en deux sous-catégories. La première c'est les SCO minimaux. Il s'agit d'un contenu capable d'être lancé via un navigateur Web et utilisant l'API (Application Program Interface) SCORM pour une communication minimale avec le système de gestion de l'apprentissage. Ce dernier est ainsi capable de surveiller le temps entre le lancement du contenu et sa terminaison

normale. La deuxième sous-catégorie de SCO est les « data-enabled SCO ». Elle est différente de la première sous-catégorie dans la mesure où l'API SCORM est utilisée pour échanger des données avec le système de gestion de l'apprentissage. Ces données peuvent concerner l'apprenant, le contenu, etc.

Concernant les éléments conteneurs au sein d'une organisation de contenu c'est soit une référence à un « sub-manifest », soit un nœud qui à des nœuds fils. Dans le cas d'un « sub-manifest » il s'agit d'un ensemble d'organisations (au moins une) avec les mêmes principes d'organisation que l'organisation qui lui fait appel. Ces organisations, contiennent à leur tour des éléments de type contenu ou conteneurs. Toutefois, les feuilles de l'arbre qui représente l'agrégation du contenu dans un « sub-manifest » sont obligatoirement des nœuds de type contenu, donc des SCO ou des « assets ». Pour ce qui est un conteneur qui est un nœud avec des fils, son seul rôle est de permettre une meilleure structuration de contenu sans que ça influence réellement la façon avec laquelle un auteur désire que le contenu soit parcouru. Toutefois dans la version SCORM 1.2 ce nœud peut faire référence à une ressource qui joue généralement le rôle d'une page de garde. Dans la version SCORM 2004 seules les feuilles peuvent faire référence à une ressource.

SCORM spécifie comment concevoir un paquetage mais sans préciser comment un système de gestion de l'apprentissage doit utiliser les différentes organisations de contenu disponibles. Toutefois, la spécification précise qu'une fois qu'une organisation est choisie, par l'apprenant ou par le système de gestion de l'apprentissage, l'apprenant doit pouvoir accéder à tout objet (SCO) défini dans l'organisation. Cependant chaque organisation de contenu reflète la vision auteur concernant la façon avec laquelle l'auteur désire que l'apprenant navigue dans le contenu.

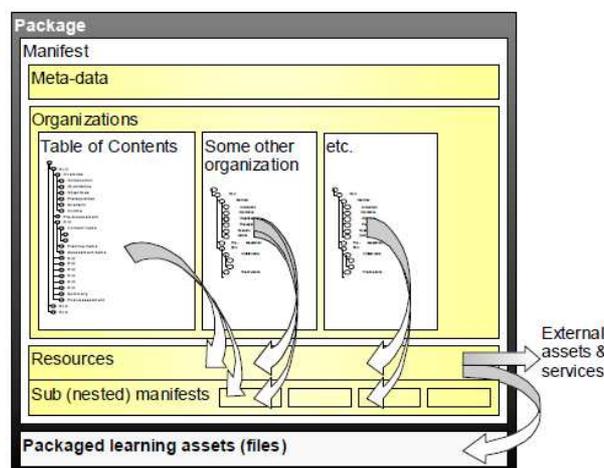


Figure 16 : Vue d'ensemble d'un paquetage SCORM

4.3. Algorithme de transformation

Dans le cas de l'approche SIMBAD c'est le graphe de composition abstrait qui permet à un auteur d'objet d'apprentissage composé de décrire les parcours possibles d'un objet d'apprentissage. Pour transformer les organisations proposées à un auteur d'objets d'apprentissage SCORM en un objet composé SIMBAD nous proposons un algorithme qui opère en deux étapes.

Lors de la première étape il transforme toute organisation dans le paquetage en un chemin élémentaire dont le nœud racine est le premier élément de l'organisation et le dernier nœud est le dernier élément de l'organisation. Ainsi tous les nœuds sont reliés les uns aux autres par l'opérateur SEQ fidèlement au parcours proposé initialement par l'auteur. Les éléments de type SCO ou Asset sont considérés comme des objets d'apprentissage atomiques, alors que les éléments qui font référence à un « sub-manifest » sont considérés comme des objets d'apprentissage composés. Pour ceux-ci leur graphe de composition abstrait est déduit par le même algorithme. Pour y parvenir nous utilisons un algorithme de parcours d'arbre en profondeur à gauche pour chaque organisation.

Lors de la deuxième étape il vérifie s'il y a plusieurs organisations dans le paquetage. Dans ce cas, il définit un nœud supplémentaire de type objet d'apprentissage atomique. Celui-ci est un fichier (par exemple HTML) qui contient quelques éléments extraits des métadonnées (par exemple le titre, la description et les noms des auteurs). Cet objet sera le nœud racine du graphe de composition. Il sera relié à chacun des nœuds racine des organisations via un opérateur ALT. Ceci traduit la sémantique décrite dans la spécification SCORM concernant l'interprétation de l'existence de plusieurs organisations. Ci-après l'algorithme définit d'une façon plus formelle.

Soient :

- NO le nombre d'organisations dans un paquetage SCORM.
- NE la liste de nombres d'éléments par organisation, avec NE [j] est le nombre d'éléments dans la $j^{\text{ième}}$ organisation.
- EP les éléments du paquetage avec EP [j] [k] est le $k^{\text{ième}}$ élément de la $j^{\text{ième}}$ organisation.
- T l'ensemble des transitions du graphe de composition abstrait, avec T [1] [j] est le nœud origine de la $j^{\text{ième}}$ transition et T [2] [j] est le nœud extrémité de cette transition.

Début

j :=1

// Commencer par traduire les organisations en des transitions

Pour i de 1 à O Faire

k :=1

Tant que k < NE [i] Faire

T [1] [j] := EP [i] [k]

T [2] [j] := EP [i] [k+1]

k++

j++

Fin tant que**Fin pour**

// Traiter le cas où on a plusieurs organisations

Si O > 1 Alors

// Créer un opérateur, ici c'est un ALT

op := créerOpérateur(ALT)

// Créer un nœud de type objet d'apprentissage qui contient quelques

//métadonnées, tel que le titre et les auteurs

root := créerRoot()

T [1] [j] := root

T [2] [j] := op

j++

Pour i de 1 à O Faire

T [1] [j] := op

T [2] [j] := EP [i] [1]

j++

Fin pour**Fin si****Fin**

4.4. Exemples de transformation

Afin d'illustrer l'algorithme nous proposons deux exemples. Le premier concerne un objet SCORM contenant une seule organisation qui ne comporte pas d'éléments de type « sub-

manifest ». L'application de l'algorithme à cet objet permet de produire un graphe de composition abstrait conforme au modèle SIMBAD. La seule organisation proposée par ce contenu est traduite par un graphe comportant trois sommets de type objet d'apprentissage et deux opérateurs SEQ indiquant l'ordre de parcours désiré par l'auteur de l'objet.

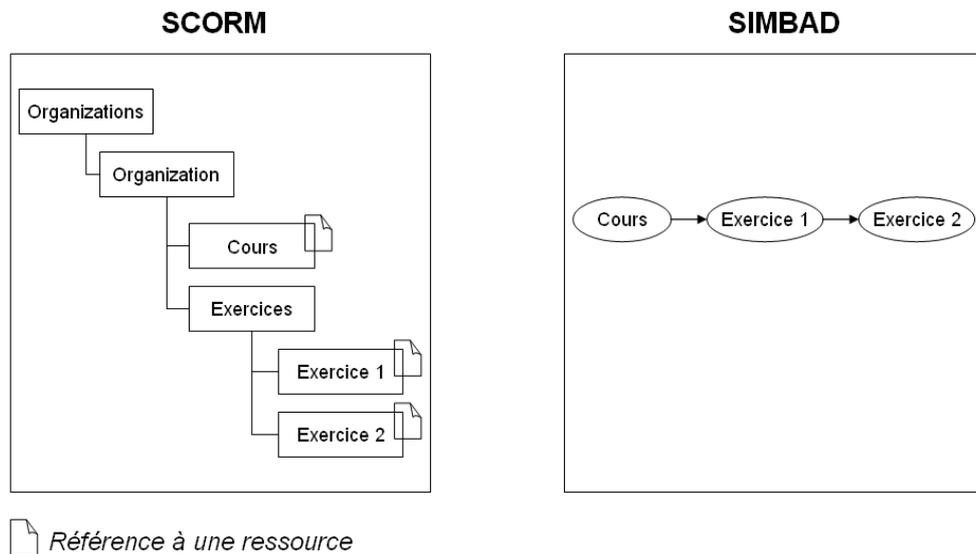


Figure 17 : Exemple de transformation d'un contenu SCORM simple en SIMBAD

Le deuxième exemple illustre la transformation d'un objet SCORM contenant deux organisations dont l'une contient un « sub-manifest ». Chaque organisation est traduite tout d'abord en un chemin. Un sommet racine est créé. Il contient une présentation sommaire de l'objet dont le contenu est issu des métadonnées LOM. Ce sommet racine est à l'origine d'une transition dont l'extrémité est un opérateur ALT qui donne accès aux deux chemins qui représentent les deux organisations proposées par l'auteur.

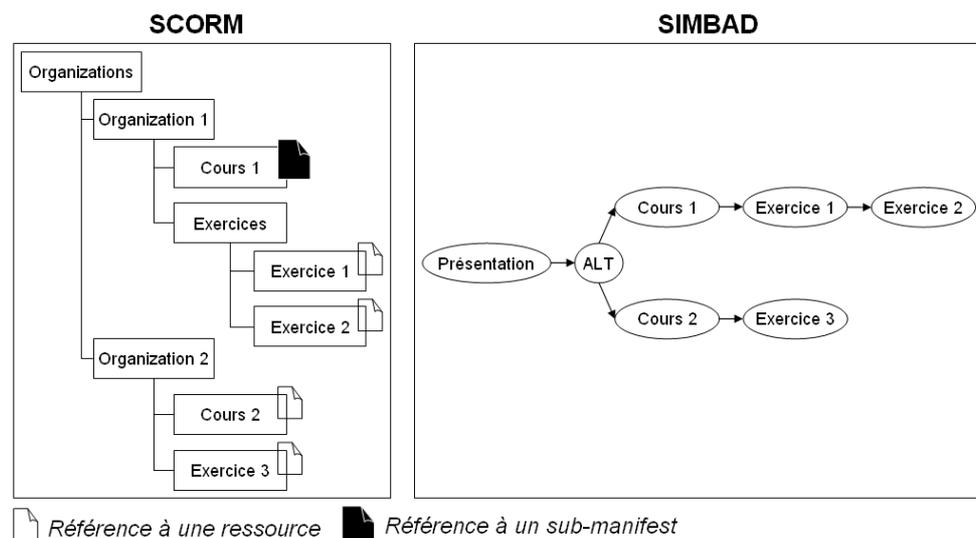


Figure 18 : Exemple de transformation d'un contenu SCORM complexe en SIMBAD

La figure suivante illustre le déroulement de l’algorithme :

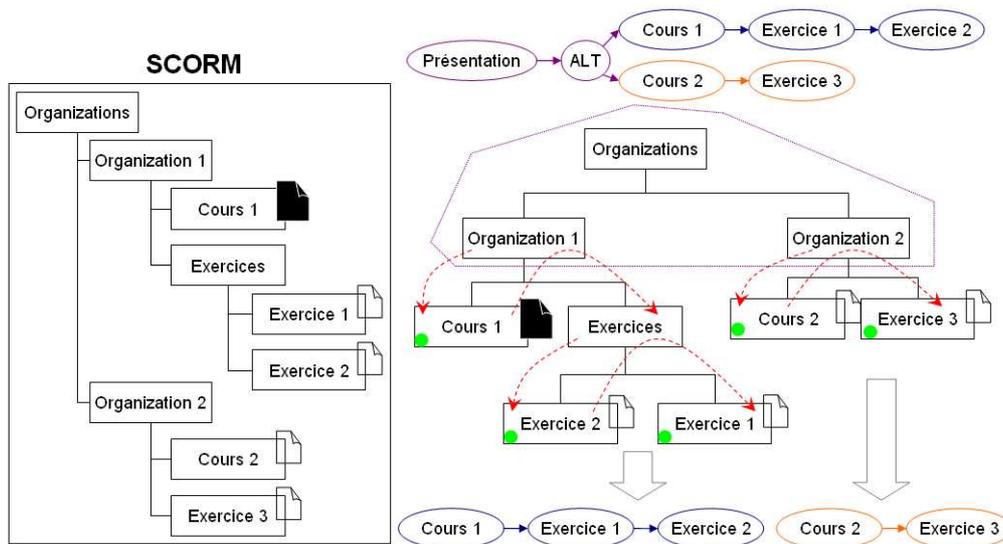


Figure 19 : Transformation d’un contenu SCORM en SIMBAD

En conclusion, nous avons montré dans cette section que tout objet SCORM peut se transformer en un objet SIMBAD équivalent. Le contraire n’est cependant pas vrai puisque SIMBAD permet de définir des graphes de composition plus complexes, non supportés par SCORM (par l’utilisation de l’opérateur PAR ou le fait que les opérateurs peuvent être combinés récursivement).

5. Conclusion

Dans ce chapitre nous avons commencé par donner une vue d’ensemble de notre approche d’assistance aux auteurs d’objets d’apprentissage composés. Cette description a mis en évidence les différentes phases et la logique d’assistance proposée.

Nous avons détaillé deux phases de l’approche d’assistance : la recherche et le calcul automatique des métadonnées. Ces deux phases ont été présentées à ce niveau afin de détailler ensuite les phases qui comportent plus d’éléments à analyser.

Nous avons, ensuite proposé un algorithme de transformation d’un objet SCORM en un objet SIMBAD. Cette transformation permettra de profiter de l’approche d’assistance pour la mise au point d’un objet SCORM sans investissement au niveau de l’approche même. De plus, ceci permettra aussi de récupérer des objets SCORM pour leur ajouter la couche sémantique et construire ainsi une banque d’objet SIMBAD.

Le chapitre suivant sera ainsi consacré à la phase de vérification automatique de la cohérence structurelle et sémantique d’un objet d’apprentissage composé.

Chapitre IV :

VERIFICATION DE LA COHERENCE

1. Introduction

L'étape de composition d'un objet d'apprentissage peut se traduire par la définition de sa structure via un graphe orienté connexe et acyclique et de sa sémantique via la définition de ses métadonnées et en particulier les métadonnées sémantiques.

Nous proposons d'assister l'auteur durant cette phase pour vérifier les aspects structurels et sémantiques de l'objet composé. Cette vérification nécessite d'une part l'identification des contrôles à effectuer et d'autre part les moyens nécessaires pour y parvenir.

2. Vérification de la cohérence structurelle

A partir de l'ensemble d'objets d'apprentissage retenus lors de la première phase, l'auteur définit un nouvel objet en les combinant ensemble. Dans le cadre de l'approche SIMBAD cette combinaison d'objets d'apprentissage se manifeste sous forme d'un graphe nommé « graphe de composition abstrait ». La connexion des objets d'apprentissage réutilisés se fait par le biais des opérateurs : ALT, PAR et SEQ.

Cette étape de la phase de conception nécessite un certain nombre de vérifications que ce soit au niveau structurel ou sémantique [Bouzeghoub, 2009].

2.1. Règles de cohérence structurelle

Concernant les aspects structurels, dans le cadre de l'approche SIMBAD six règles de cohérence structurelles doivent être vérifiées au niveau du graphe de composition abstrait.

1^{ère} règle : Le graphe doit avoir un seul point d'entrée de type objet d'apprentissage.

En fait, l'idée est d'avoir un point d'entrée unique. A partir de celui-ci, et selon le profil de l'apprenant, un chemin particulier de navigation va être retenu. Cette idée est empruntée au cas des sites Web qui offrent un point d'entrée unique qui est la page d'accueil à partir de laquelle l'internaute commence la navigation dans le site Web.

Le point d'entrée unique doit être un objet d'apprentissage et il est caractérisé par le fait qu'il n'est pas à l'extrémité d'aucun arc.

Voici un exemple d'un graphe de composition abstrait non conforme puisqu'il comporte plusieurs points d'entrée de type objet d'apprentissage :

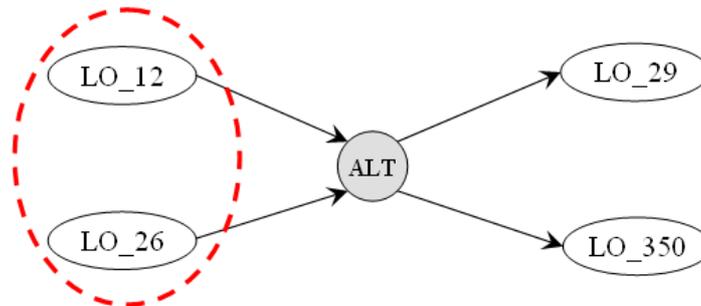


Figure 20 : Exemple de violation de la première règle de cohérence SIMBAD

2^{ème} règle : Chaque opérateur ALT ou PAR doit être à l'origine d'au moins deux arcs.

En fait les opérateurs ALT et PAR expriment respectivement la présence d'alternatives et de choix. L'alternative et le choix doivent se faire par rapport au moins deux chemins. Au niveau structurel ceci se manifeste par le fait que dès qu'il y a un opérateur PAR ou ALT, il y a présence d'au moins deux arcs dont il est la source

L'exemple suivant est un graphe de composition abstrait où cette règle est violée.

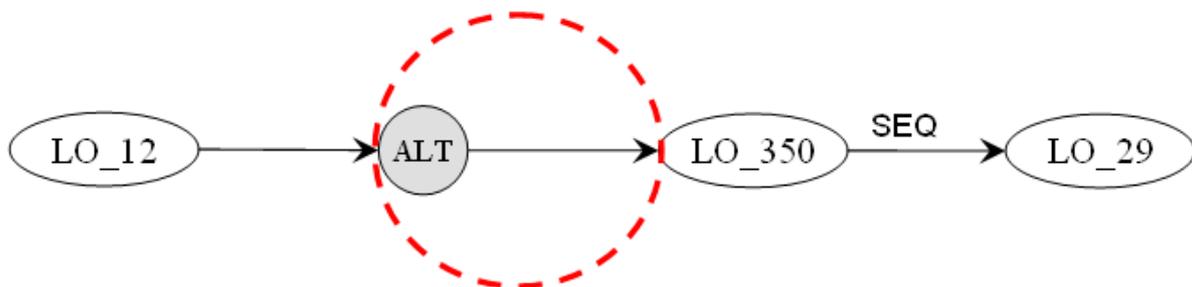


Figure 21 : Exemple de violation de la deuxième règle de cohérence SIMBAD

3^{ème} règle : Le graphe doit se terminer par un ou plusieurs sommets de type objet d'apprentissage.

Un nœud feuille par définition n'est pas à l'origine d'aucun arc. Et donc c'est absurde d'avoir un opérateur en tant que feuille dans un graphe de composition d'un objet d'apprentissage composé puisque l'opérateur n'a de sens que s'il est la source d'au moins deux arcs.

L'exemple suivant montre un graphe de composition qui viole cette règle à cause d'un nœud terminal de type opérateur.

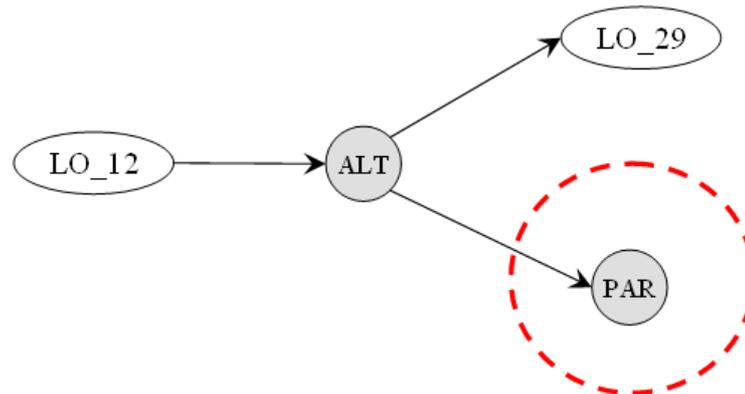


Figure 22 : Exemple de violation de la troisième règle de cohérence SIMBAD

4^{ème} règle : Chaque sommet de type objet d'apprentissage doit être à l'origine d'au plus un arc.

Puisqu'il n'y a pas d'interprétation possible du cas où un objet est à l'origine de plusieurs arcs, il est interdit d'avoir une telle structure. Si l'auteur veut exprimer une alternative il doit faire usage de l'opérateur ALT et s'il veut exprimer que le choix de l'ordre entre plusieurs chemin est libre il doit faire l'usage de l'opérateur PAR. Voici un exemple qui viole cette règle :

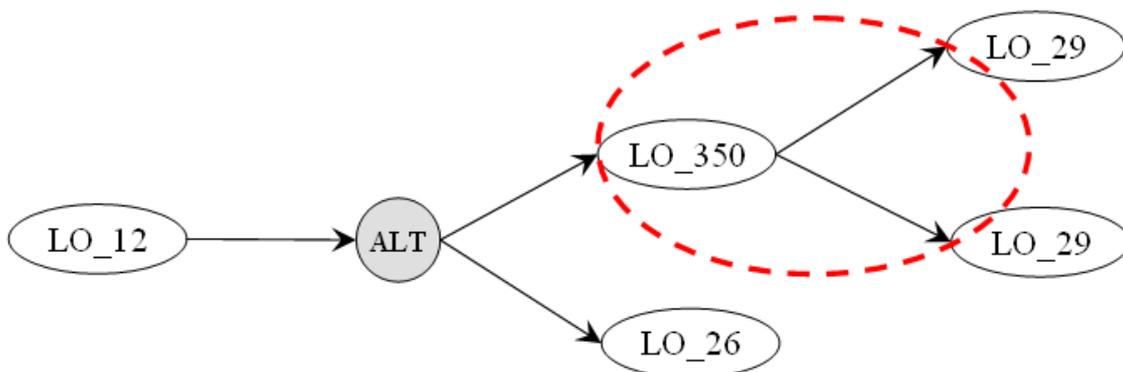


Figure 23 : Exemple de violation de la quatrième règle de cohérence SIMBAD

5^{ème} règle : Le graphe doit être acyclique.

Le graphe de composition traduit les différentes possibilités offertes par l'auteur pour avancer dans le contenu de l'objet d'apprentissage composé. Ainsi, un cycle dans le graphe ne peut pas refléter la notion d'avancement au sein d'un contenu.

L'exemple suivant présente un graphe de composition abstrait qui viole cette règle puisqu'il comporte un cycle.

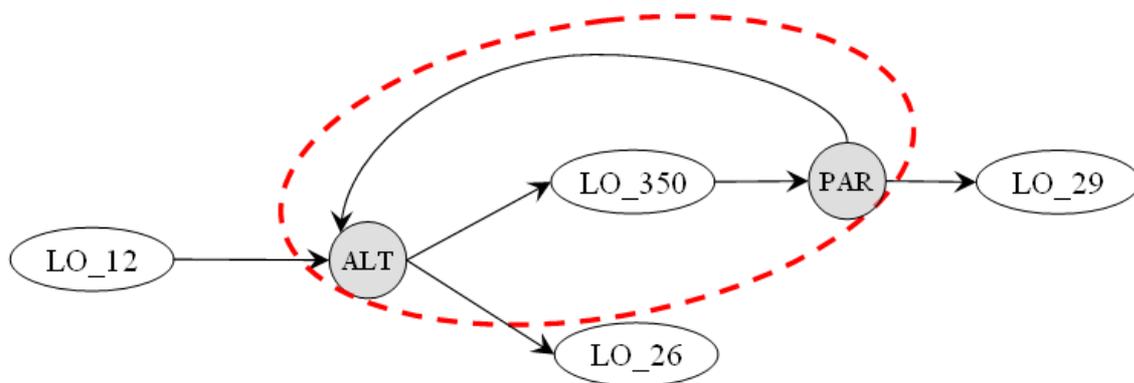


Figure 24 : Exemple de violation de la cinquième règle de cohérence SIMBAD

6^{ème} règle : Le graphe doit comporter une seule composante connexe.

En fait, le graphe de composition abstrait doit avoir une seule composante connexe. En effet, il est structuré de façon qu'à partir d'un seul point d'entrée (le sommet racine) on doit pouvoir accéder à tous les autres sommets du graphe. Cette propriété revient à dire que l'objet doit avoir une seule composante connexe.

L'exemple suivant présente un graphe de composition abstrait qui viole cette règle puisqu'il comporte trois composantes connexes. Dans le cas de la composante qui comporte le sommet LO_351 on se trouve déjà avec un graphe de composition abstrait avec deux points d'entrés possibles. Ceci revient à dire que le graphe de composition abstrait viole la première règle.

Pour la composante connexe qui comporte le sommet LO_352 on a un seul point d'entrée au niveau du graphe de composition abstrait mais avec un cycle. Ceci revient à dire que le graphe de composition abstrait viole la cinquième règle.

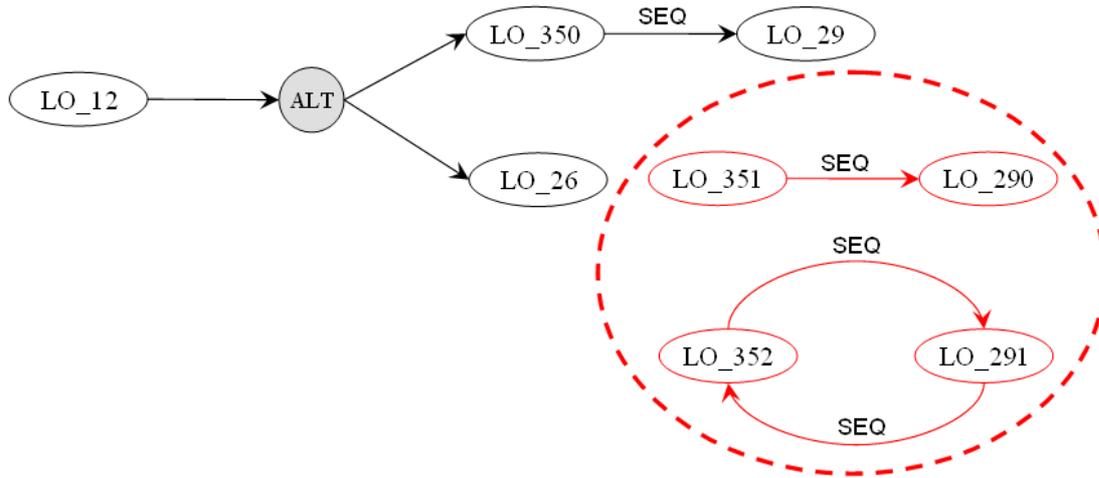


Figure 25 : Exemple de violation de la sixième règle de cohérence SIMBAD

2.2. Algorithme de vérification de la première règle

Nous allons considérer les quatre ensembles suivants :

O_i l'ensemble des objets d'apprentissage du graphe de composition de l'objet i .

V_i est l'ensemble des sommets du graphe de composition abstrait de l'objet i .

T_i est l'ensemble des arcs du graphe de composition abstrait de l'objet i .

t_{ij} est l'arc qui a comme source le sommet i et comme extrémité le sommet j .

Rappelons que la première règle indique que le graphe de composition d'un objet d'apprentissage composé doit avoir un seul point d'entrée qui est un sommet de type objet d'apprentissage.

Algorithme 3 : Vérification de l'existence d'un seul point d'entrée de type objet d'apprentissage

Booléen VerifierR1 (G : Graphe)

Début

Si $\exists i \in O / (\exists t_{ji} \in T, \forall j \in V)$ **Alors**

// Pas de sommet racine de type objet d'apprentissage

Retourner (Faux)

Sinon Si $\exists \{i, i'\} \subset O / (\exists t_{ji} \in T \text{ et } \exists t_{j'i'} \in T, \{j, j'\} \in V)$ **Alors**

// Il existe au moins deux sommets racine

Retourner (Faux)

Fin Si

Retourner (Vrai)

Fin

2.3. Algorithme de vérification de la deuxième et troisième règle

La deuxième règle à suivre, pour être structurellement cohérent avec le modèle SIMBAD, concerne les feuilles du graphe de composition abstrait. Ceux-ci doivent être de type objet d'apprentissage.

La troisième règle consiste à indiquer que tout opérateur PAR ou ALT est à l'origine d'au moins deux arcs. Il faut noter que cette troisième règle permet de garantir les propriétés demandées au niveau de la deuxième règle.

Algorithme 4 : Vérification des opérateurs

Booléen VerifierR2R3 (G : Graphe)

Début

Si $(\exists i \in V / (\exists \{t_{ij}, t_{ik}\} \subset T, \{j, k\} \subset V))$ ET $(i \notin O)$ **Alors**

Retourner (Faux)

Fin Si

Retourner (Vrai)

Fin

2.4. Algorithme de vérification de la quatrième règle

La quatrième règle précise que chaque sommet de type objet d'apprentissage doit être à l'origine d'au plus un arc.

Algorithme 5 : Vérification des objets d'apprentissage

Booléen VerifierR4 (G : Graphe)

Début

Si $(\exists i \in O)$ ET $(\exists \{t_{ij}, t_{ik}\} \subset T / \{j, k\} \subset V)$ **Alors**

Retourner (Faux)

Fin Si

Retourner (Vrai)

Fin

2.5. Algorithme de vérification de la cinquième règle

La cinquième règle impose le fait que le graphe de composition abstrait doit être acyclique. Pour y parvenir il faut utiliser un algorithme de parcours de graphe orienté avec deux marqueurs, un marqueur pour dire qu'on a visité déjà le nœud et l'autre est un jeton pour

marquer le chemin en cours de traitement. Si tous les sommets sont marqués et que l'algorithme trouve dans un chemin deux fois le même sommet alors le graphe est certainement connexe (tous les sommets sont marqués et donc visité depuis le point d'entrée) et sans cycles.

L'algorithme est récursif. Le premier appel se fait avec l'identifiant du sommet qui représente le point d'entrée du graphe et l'identifiant du graphe de composition abstrait. Dès que l'algorithme détecte un cycle il retourne « faux ».

Algorithme 6 : Détection de cycle dans un graphe

Booléen verifierR5R6(i : Sommet, G : Graph)

Début

Si (existeJeton(i) = Vrai) **Alors**

// S'il existe un jeton alors il y a un cycle car le sommet est visité deux

// fois lors du parcours du même chemin

Retourner (Faux)

Sinon

mettreMarqueur (i) // Pour dire que i est déjà visité

mettreJeton(i) // Pour dire que i appartient au chemin en cours de traitement

S = suivants(i) // S est la liste des sommets qui peuvent être visités depuis i

Pour j de 0 à taille-1 Faire

Si (verifierR5R6(S[j], G) == Faux) **Alors**

Retourner (Faux)

Sinon

j = j+1

Fin Si

Fin Pour

supprimerJeton(i)

// Le chemin auquel appartient le sommet i est visité sans trouver un cycle

Fin Si

Retourner (Vrai)

Fin

Il suffit après de vérifier que tous les sommets du graphe G sont marqués pour dire que le graphe est connexe.

3. Vérification de la cohérence sémantique

3.1. Présentation

La sémantique propre à l'objet d'apprentissage en cours de développement est définie à deux niveaux. Le premier niveau correspond à la description donnée par l'auteur de la sémantique de l'objet. Le deuxième niveau correspond aux descriptions de la sémantique des objets d'apprentissage réutilisés pour sa mise en place. Rappelons que cette description couvre trois aspects : les pré-requis, le contenu et la fonction d'acquisition. Cette description sémantique est renseignée au niveau des métadonnées de l'objet d'apprentissage.

Cette sémantique doit être cohérente. Pour pouvoir assister l'auteur de l'objet à vérifier la cohérence sémantique, nous avons défini des règles de cohérence sémantique qui touchent les différentes facettes de la description sémantique.

Par la suite nous allons détailler chacune des règles, les anomalies qu'elles permettent de découvrir et les algorithmes nécessaires pour les implémenter.

3.2. Satisfaction des pré-requis

3.2.1. Principe de satisfaction des pré-requis

Un objet d'apprentissage composé comporte un graphe de composition abstrait qui décrit une stratégie de navigation entre les objets réutilisés. Pour qu'un apprenant puisse accéder au contenu d'un objet il faut vérifier qu'il a le niveau nécessaire pour l'aborder. Si cette condition est satisfaite, il faut qu'il soit en mesure de parcourir la totalité du contenu de l'objet. C'est-à-dire, il ne doit pas se trouver bloqué à cause d'un objet d'apprentissage dont les pré-requis ne sont pas satisfaits. Ce cas de figure est le signe d'une mauvaise conception que la phase de vérification doit être en mesure de découvrir.

En fait, il faut que les pré-requis de chaque objet du graphe de composition soient satisfaits.

Un pré-requis est satisfait dans l'un des deux cas suivants :

- Il est satisfait s'il est couvert par les pré-requis de l'objet composé. C'est-à-dire si le profil de l'apprenant lui a permis d'accéder à l'objet d'apprentissage composé alors il va lui permettre également d'accéder au contenu de cet objet d'apprentissage réutilisé.
- Il est satisfait s'il est couvert par une fonction d'acquisition d'un des objets qui le précèdent dans le graphe de composition abstrait. C'est-à-dire si l'objet a un pré-requis qui ne figure pas dans les pré-requis de l'objet composé, il est certainement satisfait par le

profil de l'apprenant si un objet qui le précède (donc déjà validé par l'apprenant) contient une fonction d'acquisition qui couvre ce pré-requis.

Si ni l'un ni l'autre des cas de figure ci-dessus est vérifié ceci suppose que l'objet d'apprentissage composé tel qu'il est conçu présente une anomalie d'ordre sémantique.

3.2.2. Exemple

Afin d'illustrer le principe de satisfaction de pré-requis nous allons considérer l'objet d'apprentissage composé lo_903, ayant le graphe de composition suivant :

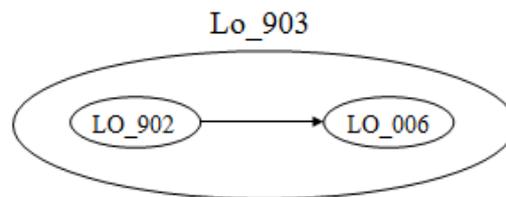


Figure 26 : Graphe de composition abstrait de l'objet d'apprentissage composé lo_903

Supposons que l'objet lo_903 est décrit par les métadonnées sémantiques suivantes :

Pré-requis : {
 <"Théorie relationnelle", "principes", "intermédiaire">,
 <"Bases de données relationnelles", "Introduction", "Débutant">}

Contenu : {
 <"Bases de données relationnelles", "principes">,
 <"Bases de données relationnelles", "exemple">,
 <"Bases de données relationnelles", "exercice">}

Fonction d'acquisition: {
 <"Bases de données relationnelles", "principes", "intermédiaire">,
 <"Bases de données relationnelles", "exemple", "intermédiaire">,
 <"Bases de données relationnelles", "exercice", "intermédiaire">}

Nous allons supposer que l'objet lo_902 est décrit quant-à-lui par les métadonnées sémantiques suivantes :

Pré-requis : { <"Théorie relationnelle", "principes", "intermédiaire">}

Contenu: {
 <"Bases de données relationnelles", "principes">,
 <"Bases de données relationnelles", "exemple">
 }

Fonction d'acquisition: {

<"Bases de données relationnelles", "principes", "intermédiaire">,

<"Bases de données relationnelles", "principes", "intermédiaire">}

Nous allons supposer que l'objet lo_006 est décrit quant-à-lui par les métadonnées sémantiques suivantes :

Pré-requis : {<"Bases de données relationnelles", "principes", "intermédiaire">}

Contenu: {<"Bases de données relationnelles", "exercice">}

Fonction d'acquisition : {<"Bases de données relationnelles", "exercice", "débutant">}

Dans cet exemple nous trouvons les deux cas de figures autorisés. En effet, pour le premier objet d'apprentissage lo_902 dans le graphe de composition nous avons un pré-requis qui est satisfait puisque l'apprenant ne peut accéder à cet objet que si son profil satisfait les pré-requis de l'objet composé lo_903. Or les pré-requis de lo_903 incluent le seul pré-requis de lo_902. Donc on est sûr que l'apprenant à un niveau intermédiaire par rapport aux principes de la théorie relationnelle lui permettant d'accéder à l'objet lo_902.

L'objet d'apprentissage lo_006 a un seul pré-requis. Celui-ci ne fait pas parti des pré-requis de l'objet composé lo_903. Donc, à priori l'apprenant qui accède au contenu de l'objet lo_903 n'est pas censé avoir un niveau intermédiaire par rapport aux principes relatifs aux bases de données relationnelles. Toutefois, l'apprenant doit accéder, d'après le graphe de composition abstrait, à l'objet lo_006 qu'une fois qu'il a acquis le contenu de l'objet lo_902. Or ce dernier, d'après sa fonction d'acquisition, permet à l'apprenant justement d'avoir un niveau intermédiaire par rapport aux principes relatifs aux bases de données relationnels et donc de satisfaire les pré-requis de l'objet lo_006.

L'intérêt de cet exemple est double. D'une part il permet de présenter les deux cas qui permettent de garantir la satisfaction des pré-requis. D'autre part il met en évidence la difficulté de la vérification manuelle même dans le cas d'un objet ayant le graphe de composition le plus simple qu'un objet d'apprentissage composé peut avoir. Ainsi, nous proposons un algorithme permettant la vérification automatique de cette règle.

3.2.3. Algorithme de vérification de la satisfaction des pré-requis

Afin d'introduire notre algorithme nous introduisant la relation d'ordre (\prec) entre les triplets de la forme $\langle c, r, l \rangle$ où : c est un concept parmi les concepts du modèle du domaine, r est un rôle éducatif et l : un niveau de maîtrise d'un concept.

Nous supposons que $\langle c_i, r_i, l_i \rangle \prec \langle c_j, r_j, l_j \rangle$ si $c_i \approx c_j$ et $r_i = r_j$ et $l_i \leq l_j$.

Le signe ‘ \approx ’ exprime une proximité sémantique entre deux concepts. Nous considérons que : $c \approx c'$ si les deux concepts c et c' sont en relation via une relation de type spécifique/générique ou une relation rhétorique (*contraste*, *similaire* ou équivalent) dans le modèle du domaine.

L’algorithme suivant décrit le processus de vérification automatique de la satisfaction de pré-requis. Nous utilisons la notation suivante :

G_i est le graphe de composition abstrait de l’objet d’apprentissage composé i . P_i est l’ensemble des pré-requis de l’objet i . A_i est l’ensemble des fonctions d’acquisition de l’objet i . C_i est l’ensemble des contenus de l’objet i . $prev_i$ est l’ensemble des objets d’apprentissages qui doivent être acquis par l’apprenant avant l’objet i d’après le graphe de composition abstrait défini par l’auteur. Nous considérons que clo est l’identifiant de l’objet d’apprentissage composé en cours de vérification.

Algorithme 7 : Vérification de la satisfaction des pré-requis

Booléen `verifierSatisfactionPrerequis (clo : ObjetComposé)`

Début

Pout tout $lo / lo \in G_{clo}$ **Faire**

Pour tout $p \in P_{lo}$ **Faire**

Si $(\exists x / p \prec x, x \in \{P_{clo} \cup A_i\} / i \in prev_{lo})$ **Alors**

Retourner (Faux)

Fin

Fin

Fin

Retourner (Vrai)

Fin

3.3. Cohérence sémantique du contenu

3.3.1. Principe de la cohérence sémantique du contenu

Lors de la composition et du renseignement des métadonnées sémantiques d’un nouvel objet d’apprentissage composé certaines anomalies peuvent apparaître à l’insu de la volonté de l’auteur. Ces anomalies peuvent être répertoriées en deux catégories :

- Métadonnées sémantiques non fidèles au contenu : c'est le cas où les métadonnées de l'objet composé ne décrivent pas fidèlement le contenu tel qu'il est décrit par les métadonnées sémantiques des objets réutilisés.
- Contenu non fidèle aux métadonnées sémantiques : c'est le cas de certains objets réutilisés dont le contenu a une description sémantique non cohérente à celle de l'objet composé.

Afin de pouvoir découvrir ces éventuelles anomalies nous devons utiliser les informations sémantiques disponibles à propos de l'objet composé et des objets réutilisés au niveau de leurs métadonnées. La vérification doit donc porter sur l'adéquation entre les métadonnées de chaque objet réutilisé et ceux de l'objet composé.

3.3.2. Exemple

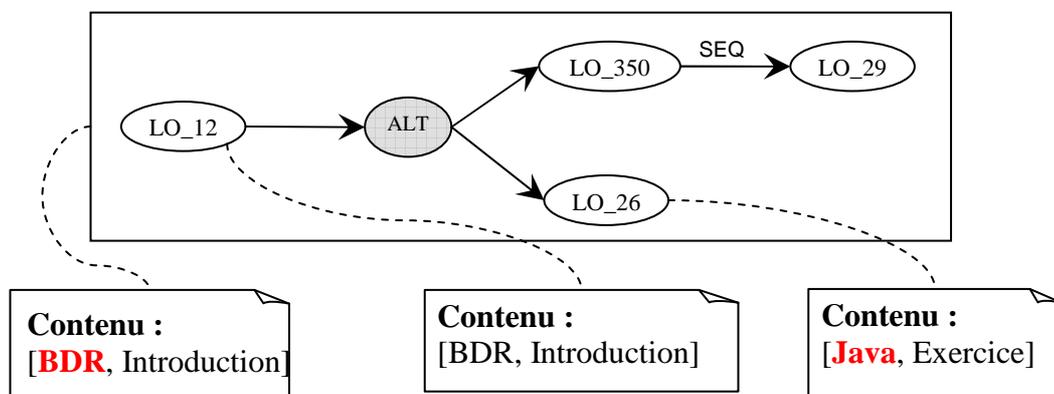


Figure 27 : Graphe de composition abstrait de l'objet d'apprentissage composé lo_904

Cet exemple illustratif présente un objet d'apprentissage composé lo_904. Le contenu de l'objet est censé être une introduction aux bases de données relationnelles. Si on examine les objets réutilisés dans le graphe de composition abstrait, on peut tomber sur l'un des deux cas suivants. Le premier cas est celui de l'objet lo_12 qui concerne le concept « bases de données relationnelles ». Ceci doit être considéré comme acceptable puisque ce concept est en proximité sémantique avec le concept traité par l'objet composé.

Par contre, le contenu de l'objet lo_26 concerne le concept « Java ». Supposons que le modèle de domaine indique que ce concept n'est pas en proximité sémantique avec le concept « bases de données relationnelles ». A ce moment nous devons être capable d'informer l'auteur sur ce risque d'incohérence dû à la présence d'un objet dont le contenu concerne un concept qui n'est pas en proximité sémantique avec aucun concept parmi ceux qui décrivent le contenu de l'objet d'apprentissage composé.

3.3.3. Algorithme de vérification de la cohérence sémantique du contenu

Afin d'introduire notre algorithme nous utilisons l'opérateur ' \approx ' qui exprime une proximité sémantique entre deux concepts. Nous considérons que $c \approx c'$ si les deux concepts c et c' sont en relation via une relation de type spécifique/générique ou une relation rhétorique (*contraste*, *similaire* ou équivalent) dans le modèle du domaine.

L'algorithme suivant décrit le processus de vérification automatique de cohérence sémantique du contenu. Nous utilisons la notation suivante :

G_i est le graphe de composition abstrait de l'objet d'apprentissage composé i . C_i est l'ensemble qui décrit le contenu de l'objet i . Nous considérons que clo est l'identifiant de l'objet d'apprentissage composé en cours de vérification.

Algorithme 8 : Vérification de la cohérence sémantique du contenu

Booléen `verifierCohérenceContenu` (clo : `ObjetComposé`)

Début

Pout tout $lo / lo \in G_{clo}$ **Faire**

Pour tout $\langle c, r \rangle \in C_{lo}$ **Faire**

Pour tout $\langle c', r' \rangle \in C_{clo}$ **Faire**

Si $c' \neq c$ **Alors**

Retourner (Faux)

Fin

Fin

Fin

Fin

Retourner (Vrai)

Fin

4. Conclusion

Dans ce chapitre nous avons présenté en détails la phase de vérification de la cohérence d'un objet d'apprentissage composé. Les vérifications touchent à la fois aux aspects structurels et aux aspects sémantiques.

Ces vérifications doivent permettre de dévoiler à l'auteur les différentes anomalies ou incohérences possibles afin de pouvoir y remédier assez tôt. Nous allons détailler dans le chapitre suivant la phase d'analyse.

Chapitre V :

ANALYSE DE L'OBJET D'APPRENTISSAGE

1. Introduction

Notre approche d'assistance doit offrir aux auteurs les éléments nécessaires pour pouvoir prendre du recul vis-à-vis de l'objet conçu. A ce niveau de notre approche, l'auteur dispose déjà d'un objet d'apprentissage composé cohérent structurellement et sémantiquement. Au-delà de la cohérence, l'auteur doit vérifier que l'objet répond aux objectifs qu'il a fixé. De plus, il doit pouvoir vérifier que l'impact de la réutilisation d'autres objets composés n'a pas altéré la nature de l'objet par rapport ce qu'il a planifié au départ. L'analyse peut mener l'auteur même à découvrir certains aspects auxquels il n'a pas songé au départ. Ceci prend plus de sens dans le cas des auteurs non expérimentés et dans le cas des objets d'apprentissage complexes.

L'objectif, au niveau de cette phase, est donc de pouvoir fournir plusieurs mesures permettant à l'auteur de *comprendre* la vraie nature de l'objet conçu. Toutefois pour pouvoir effectuer des mesures il faut déjà définir des métriques. C'est dans cette optique que nous avons défini un certain nombre de métriques élémentaires [Defude, 2005] [Farhat, 2009a]. Ces métriques doivent être combinées et agrégées par la suite sous forme d'indicateurs de plus haut niveau et plus significatifs pour les auteurs des objets d'apprentissage. Dans la deuxième partie de ce chapitre nous allons présenter les notions d'espace sémantique et de composante sémantique. Nous allons examiner les différentes métriques sémantiques que nous pouvons en extraire. Nous allons également proposer diverses interprétations et utilisation de ces métriques. Les algorithmes utilisés pour l'extraction et pour le calcul de ces métriques sont également présentés.

2. Métriques

Les métriques élémentaires que nous avons définies peuvent être classées dans des ensembles. Le premier ensemble contient des métriques qui portent sur les propriétés structurelles.

2.1. Métriques Structurelles

Au niveau structurel nous avons des graphes orientés qui décrivent la logique de parcours de l'objet d'apprentissage composé. L'idée ici est de fournir à l'auteur des indications sur la complexité de la structure du graphe de composition. Un graphe est décrit par l'ensemble de ses sommets et par les arcs les reliant. Ceci nous permet de dégager deux métriques. La première métrique (M1) permettra de mesurer le nombre de sommets du graphe. La deuxième métrique (M2) permettra de mesurer le nombre des arcs dans le graphe.

Les sommets du graphe de composition peuvent être soit des objets d'apprentissage, soit des opérateurs ALT ou encore des opérateurs PAR. Ceci permet de dégager trois métriques supplémentaires. Une métrique (M3) permettra de mesurer le nombre de sommets de type objet d'apprentissage. Une métrique (M4) permettra de mesurer le nombre de sommets qui sont des opérateurs ALT. Une autre métrique (M5) permettra de mesurer le nombre de sommets qui sont des opérateurs PAR.

Voici la liste des métriques structurelles :

M1 : nombre de sommets

M2 : nombre d'arcs

M3 : nombre de sommets de type objet d'apprentissage

M4 : nombre des opérateurs de type ALT

M5 : nombre des opérateurs de type PAR

Ces métriques sont mesurables au niveau du graphe de composition abstrait et au niveau du graphe de composition concret.

2.2. Métriques Sémantiques

Plusieurs aspects sémantiques peuvent être mesurés via des métriques spécifiques. Une première métrique (M6) permet d'évaluer le nombre de concepts traités au niveau des contenus. Il s'agit d'un indicateur sur la densité sémantique de l'objet d'apprentissage.

Une deuxième métrique (M7) permet de mesurer le nombre de concepts requis au niveau des pré-requis de l'objet d'apprentissage composé. Il s'agit d'un indicateur sur la difficulté du contenu.

Nous avons également défini une métrique (M8) qui permet de mesurer le nombre de rôles éducatifs par concept. Il s'agit d'un indicateur d'ordre pédagogique.

La dernière métrique proposée (M9) est relative au nombre moyens de sommets par rôle éducatif. Il s'agit également d'un indicateur de la densité sémantique par rapport à chaque rôle.

Voici la liste des métriques relatives aux aspects sémantiques :

M6: *nombre de concepts au niveau du contenu*

M7: *nombre de concepts au niveau des pré-requis*

M8: *nombre moyen de rôles par concept*

M9: *nombre moyen de sommets par rôle*

Contrairement aux deux premières métriques (M6 et M7) les métriques M8 et M9 peuvent être utilisés pour faire des mesures d'une part sur le graphe de composition abstrait et d'autre part sur le graphe de composition concret.

2.3. Métadonnées

Les métriques proposées concernant les métadonnées ont pour objectif de donner des indicateurs à propos de la qualité de renseignement des métadonnées, ce qui a un impact direct sur l'accessibilité de l'objet d'apprentissage.

La première métrique (M10) permet de mesurer le nombre de catégories de métadonnées renseignées. Une catégorie est considérée comme renseignée si au moins une entrée (un champ) de cette catégorie est renseignée.

La deuxième métrique (M11) permet de mesurer le nombre total d'entrées de métadonnées renseignées. Il s'agit d'un indicateur supplémentaire sur l'accessibilité de l'objet d'apprentissage.

La troisième et la quatrième métrique (M12 et M13) permettent de mesurer le pourcentage respectivement de catégories et d'entrée de métadonnées renseignées. Il s'agit d'un indicateur sur la couverture des catégories et des métadonnées selon le profil d'application LOM retenu.

Voici un récapitulatif des métriques retenues :

M10: *nombre de catégories de métadonnées renseignées.*

M11: *nombre d'entrées de métadonnées renseignées.*

M12: pourcentage de catégories de métadonnées renseignées.

M13: pourcentage des entrées de métadonnées renseignées.

2.4. Niveaux d'abstraction

L'ensemble suivant de métriques porte sur des propriétés structurelles relatives aux différentes vues d'un graphe de composition d'un objet d'apprentissage. Nous les avons mis à part, par rapport aux métriques structurelles, vu que leur champ d'application est l'objet d'apprentissage et non pas chaque graphe de composition à part.

La première métrique (M14) correspond au niveau d'abstraction du graphe de composition abstrait. La mesure s'effectue en calculant le nombre de substitutions nécessaires pour convertir tous les objets composés en objets atomiques afin d'obtenir le graphe de composition concret.

La deuxième métrique (M15) permet de mesurer le nombre de graphes affichables. Il s'agit de calculer le nombre de parcours possibles au sein du graphe de composition concret.

La troisième métrique (M16) permet de mesurer le pourcentage de sommets présents dans tous les graphes affichables par rapport au total des sommets au sein du graphe concret.

Voici un récapitulatif des métriques d'abstraction :

M14: Nombre de niveaux d'abstraction.

M15: Nombre de graphes affichables.

M16: pourcentage de sommets du graphe concret présents dans tous les graphes affichables.

2.5. Synthèse

Rappelons que l'objectif de cette phase au sein de notre approche d'assistance est de fournir aux auteurs une cartographie détaillée des propriétés et caractéristiques de l'objet d'apprentissage conçu. Certes le concepteur de l'objet d'apprentissage est censé le mieux connaître les propriétés et les caractéristiques de son objet. Toutefois, certains aspects peuvent lui échapper surtout dès que l'objet atteint une certaine taille et/ou qu'il réutilise des objets créés par d'autres. Ainsi, réexaminer l'objet d'une façon assistée et automatisée ne peut être que bénéfique.

L'analyse doit être basée sur les valeurs qui correspondent aux différentes métriques qu'on a définies. Il faut noter que notre objectif ici est loin d'être l'exhaustivité ou la complétude,

mais de démontrer qu’il est possible de définir des métriques et d’effectuer des mesures sur les objets d’apprentissage. Ces mesures peuvent ensuite être combinées et agrégées pour produire des indicateurs de plus haut niveau comme nous allons le présenter dans la suite de ce chapitre. Le tableau suivant donne une synthèse des métriques et un aperçu sur la façon avec laquelle nous allons les utiliser pour générer suite à l’analyse des indicateurs de haut niveau aux auteurs des objets d’apprentissage composés :

	Métriques
Concordance structurelle	M1, M2, M3, M4, M5, M14
Capacité à être adapté	M4, M5, M15
Capacité à être réutilisé	M6, M7, M10, M11, M12, M13
Richesse didactique	M8, M9, M16

Il faut noter que d’autres métriques liées aux styles d’apprentissage d’un objet d’apprentissage ont été identifiés dans [Rojas 2008]. Dans ces travaux la classification de Honey et Mamford pour les styles d’apprentissage [Honey 1982], à savoir « *Activists* », « *Reflectors* », « *Theorists* » et « *Pragmatists* », a été retenue. Cette étude a donné lieu à quatre métriques relatives aux styles d’apprentissage. La première permet de mesurer le nombre de concepts théoriques. La deuxième permet de mesurer le nombre de vidéos, modèles visuels et images. La troisième permet de mesurer le nombre d’exercices pratiques. Alors que la dernière permet d’évaluer la possibilité de travailler en groupe.

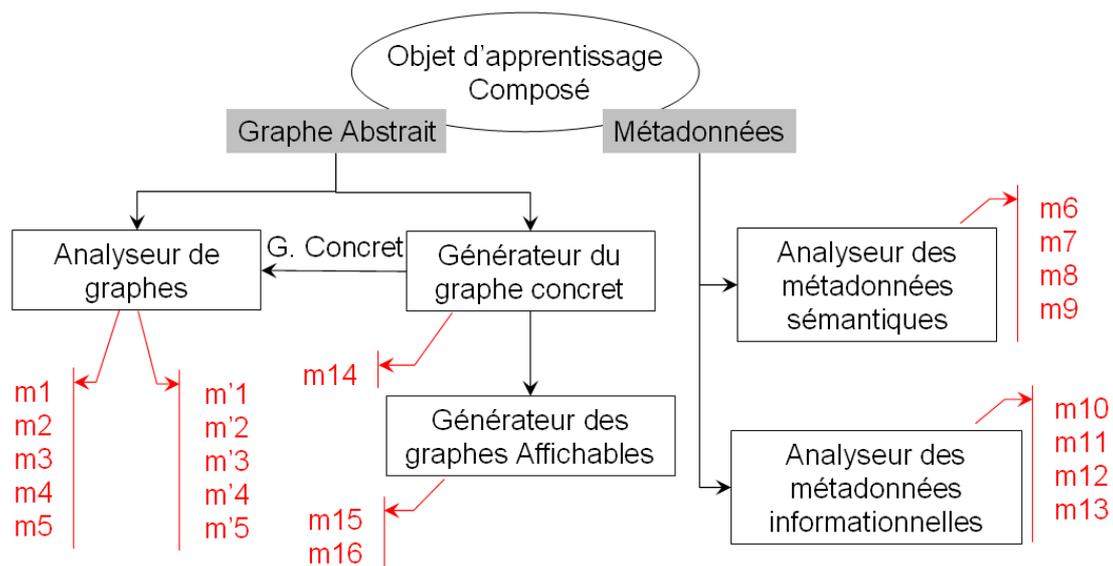
3. Indicateurs

Ces métriques de bas niveau seront difficilement utilisables par les auteurs. Pour leur donner un sens d’un point de vue métier il faut pouvoir les utiliser afin de produire des indicateurs de haut niveau compréhensible et significatif pour les auteurs d’objets d’apprentissage.

Les différentes métriques sont produites par différents modules. Trois des modules exploitent la description du graphe abstrait. En effet, le « générateur du graphe concret » permet en plus de sa fonction principale de générer la métrique relative aux niveaux d’abstraction (M14). Le

graphe de composition concret généré par ce module sera utilisé par le « générateur de graphes affichables » qui permettra de fournir les métriques relatives au nombre de graphes affichables (M15) et au pourcentage de sommets présents dans tous les graphes affichables (M16). Le graphe de composition concret et le graphe de composition abstrait seront analysés par « l'analyseur de graphes » afin de donner des mesures relatives aux métriques structurelles. Les métriques relatives au graphe de composition concret sont marquées avec une apostrophe.

Les métriques relatives à la sémantique et aux métadonnées sont générées par « l'analyseur de métadonnées sémantiques » et « l'analyseur des métadonnées informationnelles ».



Notation : m_i est la mesure associée à la métrique M_i

Figure 28 : Génération des métriques relatives aux objets d'apprentissage composés

Quatre indicateurs sont calculés à base des métriques relatives aux objets d'apprentissage composés : le degré de concordance structurelle, la capacité à être adapté, la capacité à être réutilisé et la consistance didactique. Dans un but d'uniformisation nous allons considérer une échelle de mesure qui varie de zéro à Max.

3.1. Concordance structurelle

Le premier indicateur est celui qui concerne la *concordance structurelle*. Pour cet indicateur une valeur proche de Max doit être attribuée s'il n'y a pas de différences entre la vue auteur (le graphe de composition abstrait) et la vue système (graphe de composition concret). Une valeur proche de zéro indiquera qu'une différence considérable est constatée entre la structure du graphe abstrait et celle du graphe concret.

$$V_1 = \left(c_1 \times \frac{m1}{m'1} + c_2 \times \frac{m2}{m'2} + c_3 \times \frac{m3}{m'3} + c_4 \times \frac{m4+1}{m'4+1} + c_5 \times \frac{m5+1}{m'5+1} + c_6 \times \frac{1}{m14+1} \right) \times \frac{Max}{\sum_{i=1}^6 c_i}$$

$V_1 \in [0, Max]$, en effet :

Pour les métriques m_i telles que $i \in [1,3]$, nous avons $\frac{m_i}{m'_i} \leq 1$ puisque la structure devient plus compliquée structurellement au niveau du graphe concret et donc toujours $m_i \leq m'_i$.

Pour les métrique m_4 et m_5 relatives au nombre des opérateurs ALT et PAR nous avons utilisé la formule $\frac{m_i + 1}{m'_i + 1} \leq 1$ pour éviter le cas de la division par zéro en cas d'absence d'opérateurs.

Pour la métrique m_{14} relative au niveau d'abstraction nous avons $\frac{1}{m_{14} + 1} \leq 1$ puisque d'une part m_{14} est inversement proportionnelle au degré de concordance structurelle et d'autre part cette valeur peut être nulle s'il n'y a pas d'objets composés dans le graphe de composition abstrait.

Chacun des termes de la formule est pondéré par un coefficient c_i qui permettra de paramétrer l'impact de chaque terme sur l'estimation du degré de concordance. La valeur finale est normalisée par la division de la somme des termes multipliés par leurs coefficients respectifs par la somme des coefficients. La valeur obtenue, qui est entre 0 et 1, est multipliée par la valeur maximum de l'échelle de mesure.

Rappelons que la valeur V_1 est égale à Max si le graphe concret et le graphe abstrait sont identiques (au cas où tous les sommets sont atomiques). Si la structure est plutôt la même au niveau du graphe abstrait et au niveau du graphe concret la valeur V_1 va avoir une valeur qui converge vers Max, ce qui indique une concordance structurelle élevée. Dans le cas contraire la valeur de V_1 va converger vers 0 ce qui reflète une faible concordance entre le graphe abstrait et le graphe concret.

C'est ce dernier cas qui devrait amener l'auteur à vouloir investiguer plus avant sur la cause de cette discordance structurelle et son impact sur l'objet d'apprentissage. Les valeurs des métriques utilisées peuvent donner des indications supplémentaires à l'auteur.

3.2. Capacité à être adapté

Le deuxième indicateur à déterminer concerne la capacité de l'objet d'apprentissage à être adapté aux profils des apprenants. Cette capacité dépend du nombre des opérateurs PAR et ALT, donc des métriques M4 et M5 du graphe concret. Ainsi lorsque leur nombre est élevé alors la capacité d'adaptation est importante. D'autre part cette capacité dépend du nombre de graphes affichables. Lorsque le nombre de graphes affichables est important alors la capacité de personnalisation est importante.

Ainsi, nous proposons la formule suivante :

$$V_2 = \left(c_1 \times \frac{m'_4 + m'_5}{m'_1} + c_2 \times \left(1 - \frac{1}{m_{15}} \right) \right) \times \frac{Max}{c_1 + c_2}$$

$V_2 \in [0, Max]$, en effet :

Le terme $\frac{m'_2 + m'_3}{m'_1}$ est inférieur à 1 puisque le nombre des opérateurs est forcément inférieur au nombre total des sommets du graphe de composition. La valeur du terme croît vers 1 lorsque le nombre des opérateurs PAR et ALT est important.

Le deuxième terme $1 - \frac{1}{m_{15}}$ varie de 0 vers 1 lorsque le nombre de graphes affichables est plus important.

Chacun des deux termes de la formule est pondéré par un coefficient c_i qui permettra de paramétrer l'impact de chaque terme sur l'estimation du degré de concordance. La valeur finale est normalisée par la division de la somme des termes multipliés par leurs coefficients respectifs par la somme des coefficients. La valeur obtenue, qui est entre 0 et 1, est multipliée par la valeur maximum de l'échelle de mesure.

3.3. Capacité à être réutilisé

Le deuxième indicateur concerne la capacité d'un objet d'apprentissage à être réutilisé. Cette capacité dépend des métadonnées et de la complexité sémantique de l'objet. Les mesures liées aux métadonnées sont utilisées dans des termes qui convergent vers 1 lorsqu'ils croissent et vers 0 si elles sont faibles. Quant aux termes qui utilisent les métriques liées au nombre des concepts dans le contenu et au nombre des pré-requis elles convergent vers 1 si les valeurs

sont faibles (pas beaucoup d'exigences au niveau des pré-requis et le contenu n'est pas trop chargé de concepts).

$$V_3 = \left(\frac{c_1}{m_6} + \frac{c_2}{1+m_7} + c_3 \times \left(1 - \frac{1}{m_{10}+1} \right) + c_4 \times \left(1 - \frac{1}{m_{11}+1} \right) + c_5 \times m_{12} + c_6 \times m_{13} \right) \times \frac{Max}{\sum_{i=1}^6 c_i}$$

$V_3 \in [0, Max]$, en effet :

Le terme $\frac{1}{m_6}$, pour la métrique M6 est inférieur à 1. Ce terme décroît dès que le nombre des concepts au niveau du contenu (respectivement au niveau des pré-requis) augmente, sachant que ce nombre ne peut pas être nul.

Le terme $\frac{1}{1+m_7}$, pour la métrique M7, est inférieur à 1. Ce terme décroît dès que le nombre des concepts au niveau des pré-requis augmente, sachant que ce nombre peut être nul et donc on l'additionne à 1.

Le terme $1 - \frac{1}{m_i + 1}$, pour les métriques M10 et M11, est inférieur à 1. En effet, au pire des cas, si la métrique a une valeur nulle le terme prend également la valeur nulle. Par contre, si les métadonnées sont totalement renseignées, la valeur du terme va tendre vers 1.

Les valeurs relatives aux métriques M12 et M13 sont des pourcentages. Ainsi, ils vont naturellement être compris entre 0 et 1.

Chacun des termes de la formule est pondéré par un coefficient c_i qui permettra de paramétrer l'impact de chaque terme sur l'estimation du degré de concordance. La valeur finale est normalisée par la division de la somme des termes multipliés par leurs coefficients respectifs par la somme des coefficients. La valeur obtenue, qui est entre 0 et 1, est multipliée par la valeur maximum de l'échelle de mesure.

3.4. Consistance didactique

Le quatrième indicateur porte sur la *consistance didactique* de l'objet d'apprentissage. Sa valeur croît lorsque le nombre moyen de rôles par concept, le nombre moyen de sommets par rôle et la similarité entre les graphes affichables sont importants.

$$V_4 = c_1 \times \left(\left(1 - \frac{1}{m_8} \right) + c_2 \times \left(1 - \frac{1}{m_9} \right) + c_3 \times m_{16} \right) \times \frac{Max}{\sum_{i=1}^3 c_i}$$

$V_4 \in [0, Max]$, en effet :

Le terme $1 - \frac{1}{m_i}$, pour les métriques M8 et M9, est inférieur à 1. En effet, les métriques relatives au nombre moyen de rôles par concept et au nombre moyen de sommets par rôle ne peuvent pas déjà être nulles. Si la valeur de la mesure augmente la valeur du terme va tendre vers 1.

La métrique M16 est un pourcentage. Ainsi, sa valeur est naturellement comprise en 0 et 1.

Chacun des termes de la formule est pondéré par un coefficient c_i qui permettra de paramétrer l'impact de chaque terme sur l'estimation du degré de concordance. La valeur finale est normalisée par la division de la somme des termes multipliés par leurs coefficients respectifs par la somme des coefficients. La valeur obtenue, qui est entre 0 et 1, est multipliée par la valeur maximum de l'échelle de mesure.

3.5. Exemple illustratif

Pour illustrer le processus d'analyse d'objet d'apprentissage composé nous allons considérer l'objet, intitulé lo_400, présenté par la figure suivante.

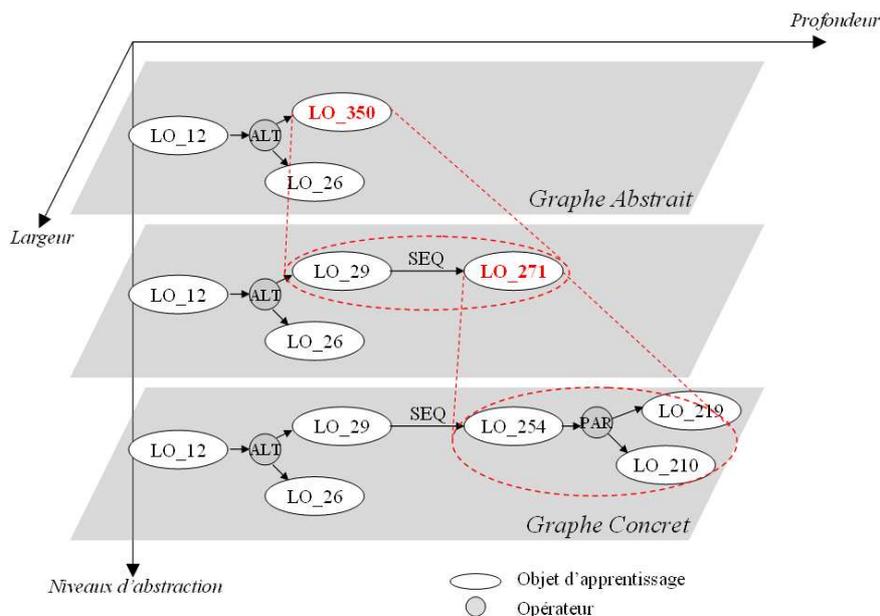


Figure 29 : Structure de l'objet d'apprentissage lo_400

Soient les métriques pour cet objet d'apprentissage composé :

Métriques liées à la structure

$m_1=4$; $m'_1=8$; $m_2=3$; $m'_2=7$; $m_3=3$; $m'_3=6$; $m_4=1$; $m'_4=1$; $m_5=0$; $m'_5=1$

Mesures relatives aux métriques liées à la sémantique

$m_6=2$; $m_7=1$; $m_8=3$; $m_9=2$

Métriques liées aux métadonnées

$m_{10}=3$; $m_{11}=10$; $m_{12}= 0.33$; $m_{13}=0.17$

Métriques liées aux niveaux d'abstraction

$m_{14}=2$; $m_{15}=2$; $m_{16}=0.17$

Nous allons considérer une échelle de 10 (Max = 10) et des coefficients de pondération tous égaux à 1 ($c_i = 1$). L'application des différentes formules présentées auparavant permet d'obtenir les valeurs suivantes pour les divers indicateurs :

Degré de concordance structurelle	Capacité d'être adapté	Capacité d'être réutilisé	Richesse didactique
V1 = 5.4	V2 = 3.75	V3 = 6.1	V4 = 4.46

Ces résultats peuvent être présentés de diverses façons notamment graphiques :

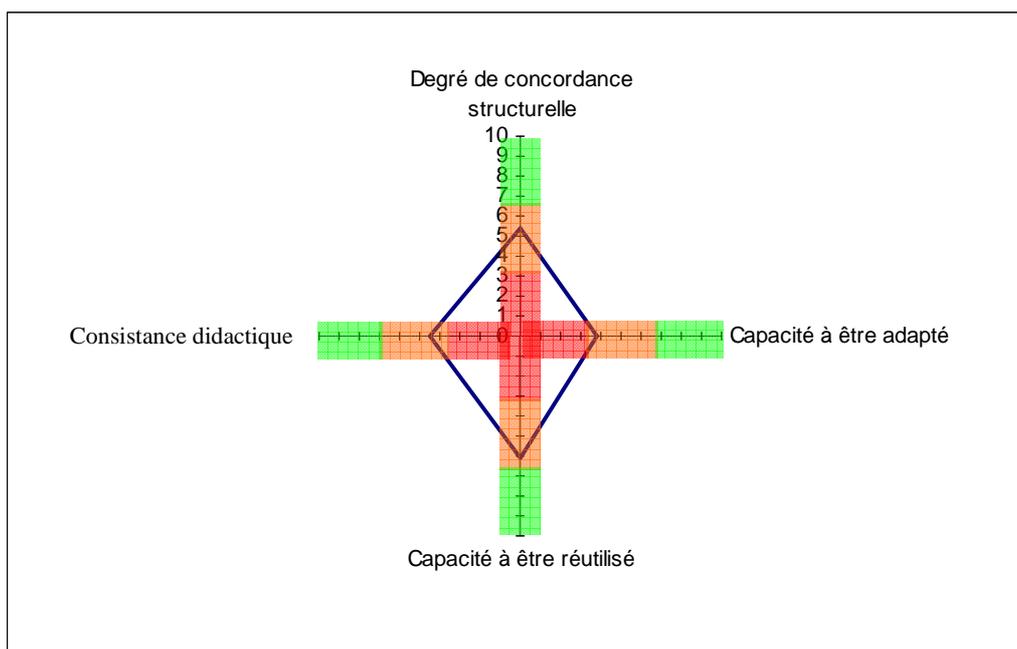


Figure 30 : Vue en mode radar des résultats d'analyse d'un objet d'apprentissage composé

Cette vue permettra à l'auteur d'avoir un résumé sur les caractéristiques de l'objet d'apprentissage qu'il a composé par assemblage d'objets existants. Dans le cas de notre exemple, l'auteur sera surtout alerté de la faible capacité de l'objet à être adapté. Ceci viens du fait qu'il y'en a que deux opérateurs chacun à l'origine de deux arcs.

4. Analyse de l'espace sémantique

Vu la nature didactique des objets d'apprentissage nous proposons un outil d'analyse supplémentaire dédié à la sémantique du contenu. L'objectif de cet outil est de permettre d'analyser le contenu de l'objet d'apprentissage en prenant comme référentiel le modèle de domaine. C'est dans ce contexte que nous introduisons la notion d'espace sémantique. En se basant sur cette notion, nous proposons des méthodes d'analyse permettant d'assister l'auteur afin de décortiquer les spécificités sémantiques de son objet d'apprentissage et de lui permettre d'améliorer son objet (en ajoutant ou en retirant des concepts du contenu).

Nous allons introduire au départ les notions d'espace sémantique et de composante sémantique telles que nous les avons définies. Ces notions seront utilisées afin de dégager les propriétés et caractéristiques relatives au contenu de l'objet d'apprentissage.

4.1. Notion d'espace sémantique

L'espace sémantique d'un objet d'apprentissage est défini comme étant l'ensemble des concepts traités au niveau du contenu d'un objet d'apprentissage, ainsi que les relations de type spécifique/générique qui les relient.

L'espace sémantique est déduit du modèle de domaine qui représente un référentiel commun dans les systèmes d'apprentissage en ligne sémantiques tel que SIMBAD. Pour y parvenir il suffit de marquer les concepts du modèle de domaine qui font partie du contenu et de retenir les relations spécifique/générique qui les relient. Ainsi, l'espace sémantique d'un objet d'apprentissage « clo » est défini de la façon suivante :

$$ES_{clo} = \{SC_{clo}, SR_{clo}\}$$

L'ensemble SC_{clo} est formé par les concepts du modèle de domaine qui décrivent le contenu de l'objet d'apprentissage « clo ». L'ensemble SR_{clo} est formé par les relations spécifique/générique qui existent entre les concepts qui font partie de SC_{clo} .

En fait, nous ne nous intéressons qu'aux relations de type spécifique/générique car nous ne voulons garder que la hiérarchie entre les concepts. Ce type de relations à notre avis est

suffisant pour pouvoir étudier les aspects cohérence, continuité et complexité de contenu d'un point de vue sémantique.

Soit un objet d'apprentissage, dont l'identifiant est LO_222, avec le contenu suivant :

```

{
  <Algorithme, Introduction>, <Algorithme, Définition>,
  <Algorithme, Exemple>, <Déclaration, Introduction>,
  <Type, Introduction>, <Type simple, Introduction>,
  <Variable, Définition>, <Variable, Exemples>,
  <Constante, Définition>, <Constante, Exemple>,
  <Corps, Introduction>, <Instruction, Définition>,
  <Ecriture, Définition>, <Ecriture, Exemple>,
  <Lecture, Définition>, <Lecture, Exemple>,
  <Affectation, Définition>, <Affectation, Exemple>,
  <Affectation, Exercice>, <Conditionnel, Définition>,
  <Conditionnel, Exemple>, <Conditionnel, Exercice>,
  <Itération, Définition>, <Itération, Exemple>,
  <Itération, Exercice>
}

```

Et soit cette version simplifiée d'un modèle de domaine qui couvre le domaine de l'algorithmique :

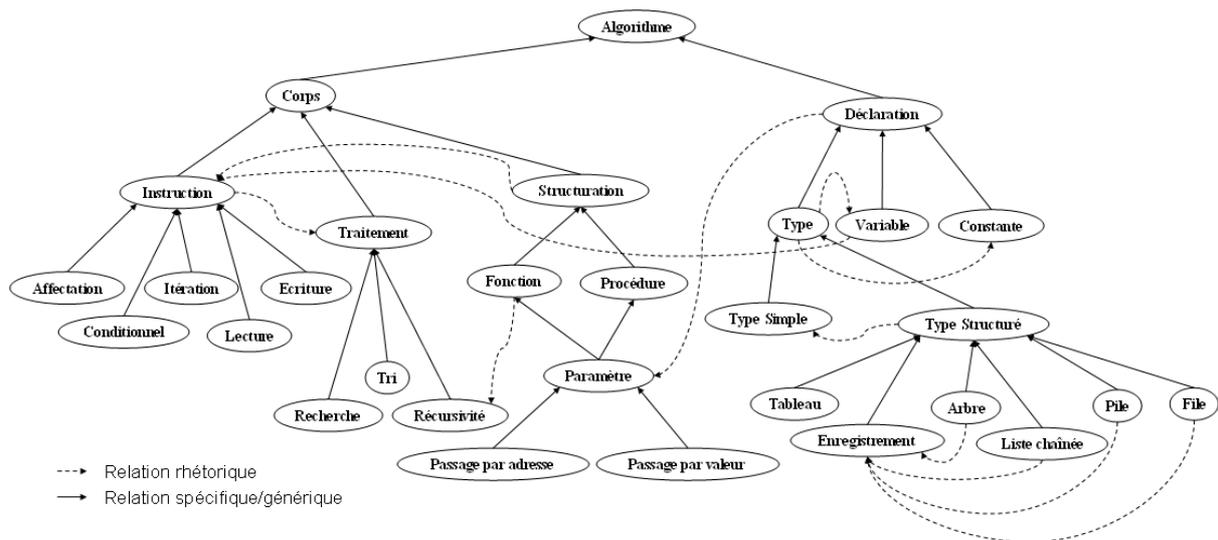


Figure 31 : Modèle de domaine (simplifié)

L'espace sémantique de l'objet d'apprentissage LO_222 est le suivant (c'est un sous-graphe du modèle de domaine) :

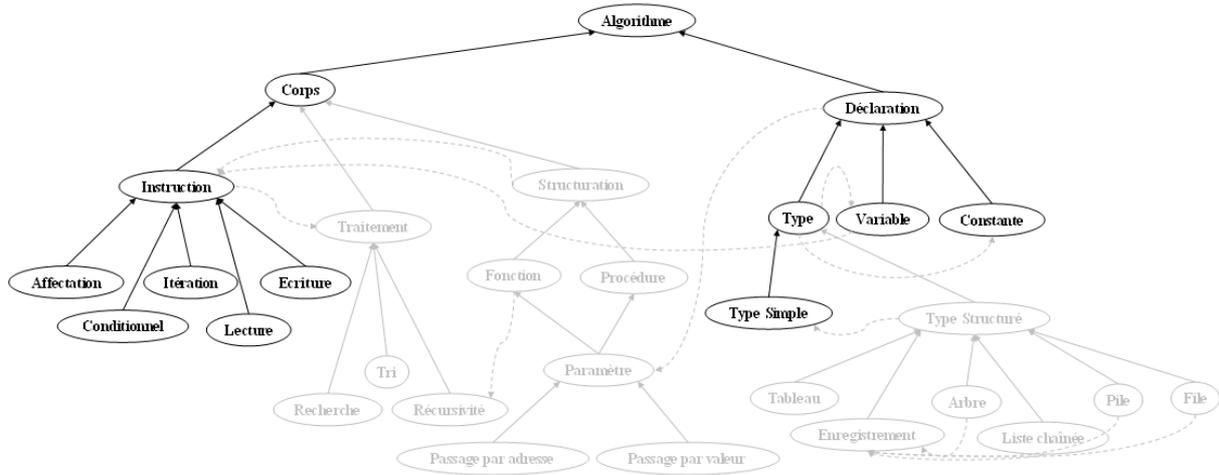


Figure 32 : Représentation graphique d'un espace sémantique

4.2. Calcul de l'espace sémantique

Nous allons noter O'_{clo} l'ensemble des objets d'apprentissage qui appartiennent au graphe de composition concret. Et soit R l'ensemble de toutes les associations de type spécifique/générique dans le modèle de domaine.

L'algorithme suivant permet de calculer l'espace sémantique d'un objet d'apprentissage composé :

Algorithme 9 : Calcul de l'espace sémantique

calculerEspaceSémantique (clo : ObjetComposé)

Debut

Pour tout ($i \in O'_{clo}$) **Faire**

Pour tout ($c \in C_i$) **Faire**

AddConceptToSC(c, SC)

Fin Pour

Fin Pour

Pour tout ($r \in R$) **Faire**

Si fromConcept (r, ci) et toConcept(r,cj) et $\{ci, cj\} \subset SC$ **Alors**

addRelationshipToSR (r, SR)

Fin

Fin

Fin

Afin d'automatiser l'interprétation de l'espace sémantique et d'en tirer les propriétés qui seront présentées à l'auteur de l'objet d'apprentissage, nous allons introduire le concept de composante sémantique.

4.3. Composante sémantique

Une composante sémantique est un groupement de concepts qui sont en proximité. La proximité sémantique se traduit au niveau de l'espace sémantique par une distance entre les concepts qui est inférieure à un seuil fixé. Cette distance correspond au chemin le plus court entre deux concepts donnés pour qu'ils puissent faire partie de la composante sémantique. Les composantes sémantiques forment des graphes connexes.

A titre d'exemple, la distance entre les concepts « Itération » et « Type simple » dans l'exemple ci-dessus est égale à cinq. D'autre part, la distance entre les concepts « Itération » et « Affectation » est deux. Ainsi, une distance faible reflète une proximité sémantique.

Ainsi, nous pouvons ainsi définir un espace sémantique d'un objet d'apprentissage comme un couple $\{CS, s\}$, avec CS est l'ensemble de composantes sémantiques et s est le seuil de proximité sémantique.

Nous pouvons définir une composante sémantique cs_i comme étant un ensemble de concepts C tel que : $\forall \{c_x, c_y\} \subset C, d(c_x, c_y) \leq s$

Prenons comme exemple l'objet d'apprentissage du paragraphe précédent. Définissant, l'espace sémantique comme étant l'ensemble de composantes sémantiques conformes à un seuil s égal à deux. La figure suivante montre les différentes composantes sémantiques au niveau de l'espace sémantique de l'objet d'apprentissage LO_222.

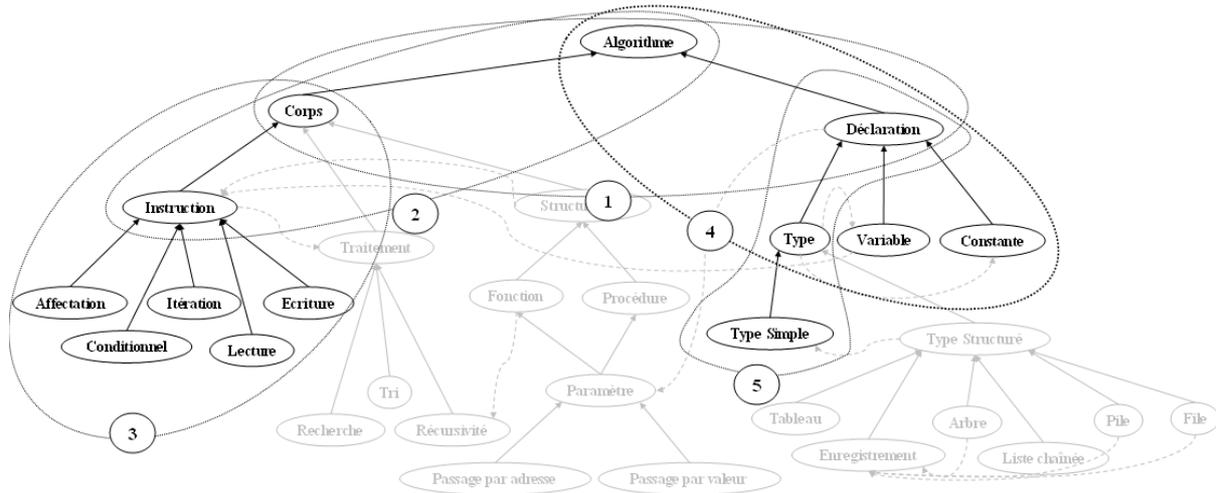


Figure 33 : Composantes sémantiques d'un espace sémantique

Nous allons utiliser ces notions d'espace sémantique et de composante sémantique afin d'analyser les caractéristiques sémantiques des objets d'apprentissage.

4.4. Complexité sémantique du contenu

L'un des facteurs qui peuvent rendre un contenu éducatif compliqué à assimiler est sa densité sémantique. Celle-ci peut être considérée comme étant proportionnelle au nombre de concepts traités au niveau du contenu. Sauf que le fait que certains concepts sont en proximité sémantique doit être pris en compte. En effet, si nous disposons de douze concepts traités au niveau d'un contenu éducatif et que ceux-ci font partie d'une même composante sémantique cela doit être plus simple à assimiler par l'apprenant que si ces concepts font partie de douze composantes sémantiques différentes. Rappelons qu'une composante sémantique reflète une proximité sémantique. Ce qui justifie le fait qu'acquérir douze concepts en proximité sémantique est certainement moins difficile qu'acquérir douze concepts qui font partie de douze composantes sémantiques. C'est-à-dire des concepts qui ne sont pas en proximité sémantique et donc nécessitent de la part de l'apprenant plus d'effort pour les assimiler. Ainsi, la complexité du contenu (d'un point de vue sémantique) peut être évaluée en fonction du nombre des concepts et du nombre de composantes sémantiques.

Soient x le nombre de concepts et y le nombre de composantes sémantiques au sein de l'espace sémantique d'un objet d'apprentissage.

Nous proposons la formule suivante pour évaluer la complexité sémantique du contenu :

$$\left(c_1 \frac{\log(x)}{\log(\text{Max}(\alpha, x))} + c_2 \frac{\log(y)}{\log(\text{Max}(\beta, y))} \right) \times \frac{E}{\sum c_i}$$

Les deux termes de la formule permettent d'avoir une variation logarithmique des valeurs permettant de contribuer à l'évaluation de la complexité du contenu. Si le contenu de l'objet ne traite qu'un seul concept alors forcément nous avons une seule composante sémantique. Les deux termes vont avoir la valeur minimale zéro car $\log(1) = 0$.

La valeur maximale de la complexité nécessite un avis d'experts qui vont décider du nombre de concepts α qui représente un seuil à partir duquel la complexité du contenu est considéré comme maximale. Un avis similaire est nécessaire pour avoir un seuil relatif au nombre de composantes sémantiques maximale β à partir duquel la complexité du contenu est considéré comme maximale. La nature du domaine de l'apprentissage, l'âge des apprenants, le niveau des apprenants et le niveau de détails retenu dans la description du modèle de domaine ont une influence sur les seuils α et β .

Finalement, pour normaliser la valeur maximale à un nous allons diviser la valeur logarithmique de x (respectivement de y) par le maximum entre α et x (respectivement de β et y).

Les coefficients c_1 et c_2 permettent de pondérer l'impact du nombre de concept et du nombre de composantes sémantiques sur la valeur estimative de la complexité sémantique du contenu. La valeur peut s'adapter à une échelle de mesures de 0 à E.

Dans le cas de l'objet LO_222 si on prend les coefficients c_1 et c_2 égaux à 1, une mesure sur une échelle de 10, une valeur α égale à 100 et une valeur β égale à 10, la complexité est calculée ainsi :

$$\left(\frac{\log(13)}{\log(100)} + \frac{\log(5)}{\log(10)} \right) \times \frac{10}{2} = 6,28$$

L'auteur peut en déduire que la complexité sémantique de cet objet d'apprentissage est plutôt moyenne.

Alors que s'il n'y avait que 2 concepts et une composante sémantique, par exemple, la complexité aura la valeur suivante : $\left(\frac{\log(2)}{\log(100)} + \frac{\log(1)}{\log(10)} \right) \times \frac{10}{2} = 0,15$

L'auteur peut en déduire que la complexité sémantique de cet objet d'apprentissage est faible.

D'autre part s'il y avait 50 concepts regroupés en 20 composantes sémantiques, par exemple,

$$\text{la complexité aura la valeur suivante : } \left(\frac{\log(50)}{\log(100)} + \frac{\log(20)}{\log(20)} \right) \times \frac{10}{2} = 9,25$$

L'auteur peut en déduire que la complexité sémantique de cet objet d'apprentissage est très élevée.

4.5. Continuité sémantique du contenu

Passer de l'étude d'un concept à l'étude d'un autre concept pour un apprenant est certainement plus aisé lorsque le passage se fait en faisant la transition via des concepts intermédiaires. Nous appelons cette propriété « continuité sémantique ». Cette facilité de passage d'un concept à un autre, que nous appelons continuité sémantique, est altérée au cas où les concepts sont éparpillés dans l'ontologie de domaine. L'espace sémantique de l'objet par conséquent contient des métriques qui permettent d'estimer le niveau de continuité sémantique d'un contenu.

En fait, nous disposons de deux métriques qui peuvent nous renseigner sur la continuité sémantique du contenu. La première métrique est le nombre de graphes connexes au sein de l'espace sémantique de l'objet d'apprentissage. En fait, si le contenu est parfaitement continu l'espace sémantique va former une seule composante connexe. Cependant, un nombre important de composantes connexes au sein de l'espace sémantique est le signe d'un contenu non continu sémantiquement.

La deuxième métrique est le nombre d'intersections entre les composantes sémantiques par rapport à leur nombre. Une intersection entre deux composantes sémantiques reflète l'existence d'au moins un concept en commun. Ce concept permet de faciliter le passage de l'étude des concepts faisant partie de la première composante sémantique à l'étude des concepts faisant partie de la deuxième composante sémantique.

Afin de produire un indicateur de la cohérence du contenu nous avons défini la formule suivante. Soit x le nombre de graphes connexes au niveau de l'espace sémantique et y le nombre d'intersections entre les n composantes sémantiques. La cohérence est calculée de la façon suivante :

$$\left(c_3 \frac{1}{(1 + \sum d_j)^x} + c_4 \frac{1}{1 + |\min(0, y - (n - 1))|} \right) \times \frac{E}{\sum c_i}$$

En fait, si le nombre de graphes connexes au niveau de l'espace sémantique est égal à un alors le terme $\frac{1}{x}$ prend sa valeur maximale qui est égale à un. Par contre si le nombre de composantes connexes est important alors la valeur de ce terme va tendre vers zéro. La quantité $\sum d_j$ correspond à la somme des distances entre les graphes connexes. La distance entre deux graphes connexes est la longueur du plus court chemin entre les deux graphes. Le coefficient c_3 est utilisé pour pondérer la contribution de ce terme dans l'évaluation de la continuité sémantique de l'objet d'apprentissage.

Le deuxième terme utilisé est $\frac{1}{1 + |\min(0; y - (n - 1))|}$ où n est le nombre de composantes

sémantiques et y est le nombre d'intersections entre elles. En fait, pour n composantes sémantiques le nombre d'intersection minimal pour un contenu qui a une continuité sémantique est égale à $n-1$. Donc, si y est égal à $n-1$ ou plus le terme doit prendre une valeur égale à un. Si le nombre y est strictement inférieur à $n-1$ alors certainement il y en a une discontinuité sémantique. Ainsi, nous allons utiliser la fonction minimum pour avoir zéro dans le cas où la valeur de y est supérieure ou égale à $n-1$ et donc le terme prend une valeur égale à 1. Si y est strictement inférieur à $n-1$ alors $y-(n-1)$ aura une valeur négative qui sera retournée par la fonction minimum et donc le terme va avoir une valeur qui tend vers 0 si la différence entre le nombre d'intersections et le nombre de composantes sémantiques est importante. La valeur absolue de la valeur retournée par la fonction minimum permet d'avoir une valeur positive pour le terme. Le coefficient c_4 est utilisé pour pondérer la contribution de ce terme dans l'évaluation de la continuité sémantique de l'objet d'apprentissage.

Ainsi, les coefficients c_3 et c_4 permettent de pondérer l'impact du nombre de graphes connexes entre les composantes sémantiques d'une part, et le nombre d'intersections entre les composantes sémantique d'autre part sur la valeur estimative de la continuité sémantique du contenu. La valeur peut s'adapter à une échelle de mesures de 0 à E.

Dans le cas de l'objet LO_222 et avec des coefficients c_3 et c_4 égaux à un et une mesure sur une échelle de dix, la continuité est calculée comme suit :

$$\left(\frac{1}{1 + \frac{1}{|\min(0; 7 - (5 - 1))|}} \right) \times \frac{10}{2} = 10$$

L'auteur peut en déduire que son objet d'apprentissage satisfait pleinement la propriété de continuité sémantique.

Par contre si l'objet comporte quatre composantes sémantiques avec une seule intersection. Et si l'espace sémantique de l'objet comporte trois graphes connexes dont la somme des distances qui les sépare est 7, alors nous avons la valeur suivante :

$$\left(\frac{1}{(1+7) \cdot 3} + \frac{1}{1 + |\min(0; 1 - (4-1))|} \right) \times \frac{10}{2} = 1,87$$

Cette valeur reflète une continuité sémantique faible qui mérite d'être examinée par l'auteur.

4.6. Déséquilibre sémantique du contenu

L'espace sémantique d'un objet d'apprentissage décrit les différents concepts traités par cet objet. Cependant, il ne permet pas de savoir si les concepts sont traités d'une façon équilibrée par les objets d'apprentissage réutilisés (au niveau du graphe de composition abstrait) ou si certains concepts ou certaines composantes ont été traités plus que d'autres.

Pour pouvoir étudier cet aspect nous proposons d'étiqueter chaque concept de l'espace sémantique par l'objet d'apprentissage qui le traite. Le nombre d'objets qui traitent un concept représente son « poids » sémantique. De la même façon nous pouvons également déterminer le poids de chaque composante sémantique.

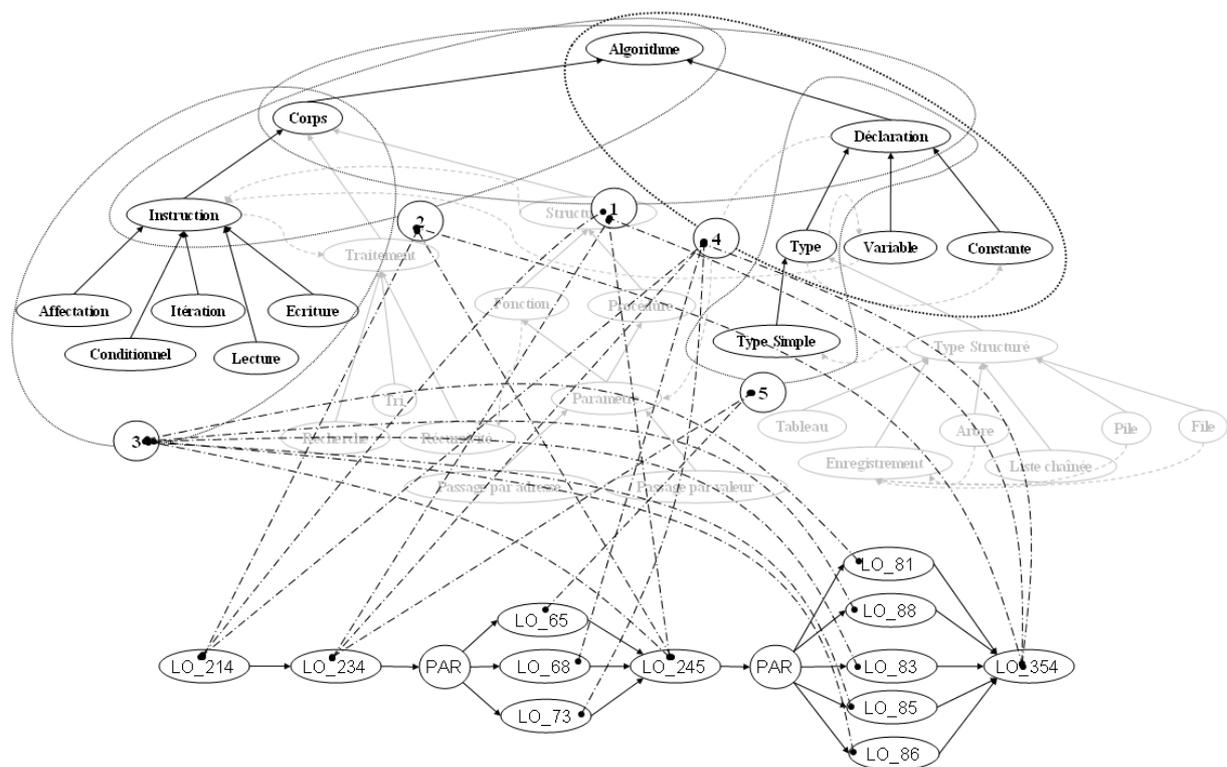


Figure 34 : Etiquetage des composantes sémantiques par les objets d'apprentissage

Cette information doit permettre à l’auteur de l’objet d’apprentissage composé de réexaminer son objet dans le cas où un grand déséquilibre est constaté. Si nécessaire l’auteur peut soit réduire le nombre de concepts traités en retenant les concepts qui ont un poids important et en éliminant ceux qui sont considérés comme marginaux. Sinon, l’auteur peut décider d’ajouter d’autres objets d’apprentissage à son graphe pour rétablir un certain équilibre entre les poids des concepts et des composantes sémantiques.

	LO_214	LO_234	LO_65	LO_68	LO_73	LO_245	LO_81	LO_88	LO_83	LO_85	LO86	LO_354	Tot
1	X	X				X						X	4
2	X					X						X	3
3						X	X	X	X	X	X		6
4	X	X		X	X							X	5
5		X	X										2

Tableau 1 : Poids des composantes sémantiques de l’objet d’apprentissage LO_222

En fait, nous disposons de deux métriques qui peuvent nous renseigner sur le déséquilibre sémantique éventuel du contenu. La première métrique est la différence entre le poids maximal et le poids minimal des différents concepts au niveau de l’espace sémantique.

La deuxième métrique est la différence entre le poids maximal et le poids minimal des différentes composantes sémantiques au niveau de l’espace sémantique de l’objet d’apprentissage.

Afin de produire une estimation du déséquilibre sémantique d’un objet d’apprentissage nous proposons la formule suivante. Soient M le poids maximal et m le poids minimal parmi les poids des concepts au sein de l’espace sémantique. Et soient M' le poids maximal et m' le poids minimal parmi les poids des composantes sémantiques. Soit D la différence maximale possible entre les poids. Cette valeur correspond au nombre d’objets d’apprentissage au sein du graphe de composition abstrait (poids maximal possible) diminuée par un (poids minimal possible). L’estimation est calculée de la façon suivante :

$$\left(c_5 \frac{\log(1 + (M - m))}{\log(1 + D)} + c_6 \frac{\log(1 + (M' - m'))}{\log(1 + D)} \right) \times \frac{E}{\sum c_i}$$

En fait, les deux termes de la formule opèrent de la même façon. S'il y a un grand écart entre le poids minimal et le poids maximal alors la valeur du terme doit tendre vers un. S'il y en a un équilibre sémantique parfait, alors le poids minimal est égal à zéro.

Les coefficients c_5 et c_6 permettent de pondérer la contribution des termes dans l'évaluation de l'équilibre sémantique. La valeur estimative prend une valeur sur une échelle de 0 à E.

Dans le cas de l'objet LO_222 et avec des coefficients c_5 et c_6 égaux à un et une mesure sur une échelle de dix, le déséquilibre sémantique est calculée comme suit :

$$\left(\frac{\log(1 + (3 - 1))}{\log(1 + 11)} + \frac{\log(1 + (6 - 2))}{\log(1 + 11)} \right) \times \frac{10}{2} = 5,45$$

L'auteur peut en déduire que l'objet présente un déséquilibre sémantique moyen qui n'est pas forcément inquiétant.

Toutefois, si l'objet avait la valeur douze comme poids maximal au niveau des concepts, et un comme poids minimal. Et si l'objet avait douze comme poids maximal au niveau des composantes sémantiques et un comme poids minimal, nous avons un déséquilibre maximal qui est révélé par la valeur estimative :

$$\left(\frac{\log(1 + (12 - 1))}{\log(1 + 11)} + \frac{\log(1 + (12 - 1))}{\log(1 + 11)} \right) \times \frac{10}{2} = 10$$

En revanche, si l'équilibre sémantique est idéal entre les concepts et entre les composantes sémantiques alors les valeurs $(M-m)$ et $(M'-m')$ sont égaux à zéro. A ce moment nous n'avons pas de déséquilibre et la valeur estimative donne zéro :

$$\left(\frac{\log(1 + 0)}{\log(1 + 11)} + \frac{\log(1 + 0)}{\log(1 + 11)} \right) \times \frac{10}{2} = 0$$

5. Apports de l'analyse

Nous considérons la phase d'analyse comme un élément clé dans notre approche. Le fait que l'auteur dispose déjà d'un objet d'apprentissage cohérent structurellement et sémantiquement lui procure la sensation que le processus de conception est terminé. Toutefois, pour rendre l'approche effectivement itérative et incrémentale il faut pousser l'auteur à examiner de nouveau son objet en lui facilitant cette tâche.

En effet, nous nous sommes mis comme objectif de fournir d'une façon automatisée des indicateurs divers et variés à propos d'un objet d'apprentissage composé. Ceci étant un moyen

pour permettre à l'auteur de l'objet de pouvoir identifier les éventuelles carences, lacunes et/ou faiblesses au niveau de l'objet par rapport aux objectifs qu'il s'est fixé au départ.

Nous avons démontré que les métriques de bas niveau que nous avons définies et calculées peuvent être combinées et agrégées pour offrir une information de plus haut niveau et orientée métier sous forme d'indicateurs. Toutefois, ceci a été fait à titre indicatif et ne constitue pas le cœur de notre contribution.

En effet, ces métriques constituent une matière première qui doit permettre aux acteurs de l'ingénierie pédagogique de pouvoir étudier la meilleure façon de les interpréter. Ces métriques constituent des éléments de réflexion pour le développement d'approches, méthodes et outils spécifiques permettant d'interpréter ces informations de la façon la plus efficace. Notons en plus que la façon avec laquelle ces informations seront présentées aux auteurs doit être étudiée. Ceci relève de l'ingénierie des interfaces homme/machine.

Un système implémentant notre approche d'assistance pourra exploiter cette analyse de diverses façons. Par exemple, le résultat de l'analyse peut être positionné par rapport à des valeurs de référence. Ces valeurs pourront être les valeurs moyennes de tous les objets présents dans une banque d'objets d'apprentissage, les valeurs moyennes des objets les plus utilisés par les utilisateurs ou encore les valeurs moyennes des objets déjà composés par le même auteur.

Cette phase d'analyse peut être également exploitée au-delà d'un environnement auteur classique. En effet, nous avons étudié son intégration dans une banque d'objets d'apprentissage. Ceci nous a permis de l'intégrer à titre expérimental dans la banque d'objets d'apprentissage COLORS (COoperative Learning Object Repository System) [Farhat et Jemni, 2009] qui sera décrite dans la partie implémentation.

Dans le système COLORS nous proposons à l'auteur suite à l'importation de son objet au niveau de la banque de le soumettre à une phase d'analyse. Les mesures effectués sont ensuite comparées à celles qui correspondent à ceux des dix objets d'apprentissage les plus consultés par les utilisateurs de la banque.

6. Conclusion

Dans ce chapitre nous avons mis en évidence dans un premier temps la possibilité d'extraire des informations implicites à partir des caractéristiques des objets d'apprentissage. Nous avons défini dans ce cadre des métriques qui couvrent les aspects structurels, sémantiques,

métadonnées et niveaux d'abstraction. En se basant sur ces métriques nous avons défini les algorithmes nécessaires pour générer les mesures adéquates.

Nous avons proposé une interprétation de ces valeurs calculées en définissant des fonctions d'agrégation. Ce travail a permis de donner des estimations sur la concordance structurelle de l'objet d'apprentissage composé, sur sa capacité à être adapté, sur sa capacité à être réutilisé et sur sa richesse d'un point de vue didactique.

Nous avons également proposé les notions d'espace sémantique et de composante sémantique. En se basant sur ces notions nous avons défini des formules paramétrables permettant de donner des valeurs estimatives purement relatives aux caractéristiques sémantiques de l'objet d'apprentissage composé sujet de l'analyse. En effet, nous avons pu donner des estimations significatives pour le niveau de complexité, pour la cohérence et pour le déséquilibre sémantiques d'un objet d'apprentissage.

Chapitre VI

ANALYSE DE LA CONFORMITE

1. Introduction

La phase d'analyse de la conformité a pour objectif de vérifier si un objet d'apprentissage est conforme à un patron de conception reflétant un ensemble de règles à respecter. Cela complète les étapes d'analyse par des moyens de contrôle plus forts et qui peuvent le cas échéant être appliqués non pas a posteriori sur les objets déjà produits mais a priori sur les objets en cours de construction. Ainsi, il est important d'intégrer dans notre approche la possibilité d'assister l'auteur pour qu'il puisse vérifier la conformité de son objet soit par rapport à une certaine vision de la qualité, soit par rapport à des principes d'harmonisation édictés par des autorités relatives au domaine de l'apprentissage en ligne.

Le chapitre est organisé comme suit. Nous commençons par donner les principes de notre approche par règles de conformité puis nous présentons le langage tout d'abord informellement par des exemples puis de manière formelle. Une notation graphique permettant aux experts / auteurs de construire ou comprendre des règles est ensuite présentée. Les principes de vérification sont ensuite posés (sachant qu'ils seront détaillés dans le chapitre suivant) et enfin un exemple illustratif permet de montrer tous les langages et notations proposés ici.

2. Principes de la vérification de la conformité

2.1. Introduction

En fait, cette phase de notre approche permet de vérifier la conformité d'un objet d'apprentissage composé par rapport à un patron de conception (i.e. un ensemble de règles). Il faut noter que cette phase se distingue des autres phases par le fait qu'elle ne remet pas forcément en question l'effort de composition d'un objet d'apprentissage composé. En effet, les experts / administrateurs peuvent éventuellement avoir définis plusieurs ensembles de règles de conformité et alors un objet n'a pas à être conforme à tous ces ensembles mais seulement à l'un (au moins) d'entre eux.

Par exemple, supposant qu'un premier ensemble de règles reflète une pédagogie d'apprentissage par compétence et qu'un deuxième ensemble de règles reflète une pédagogie par l'exemple. Si un objet donné est soumis pour une vérification de conformité par rapport au premier ensemble et qu'il n'est pas conforme alors l'auteur peut en déduire qu'il n'est pas destiné à un apprentissage par compétence. En même temps ce même objet peut être conforme au deuxième ensemble de règles et donc être adapté à un apprentissage par l'exemple. Les résultats négatifs d'une analyse de conformité peuvent faire découvrir à l'auteur les éléments à modifier pour devenir conforme (dans l'exemple ci-dessus comment adapter l'objet à un apprentissage par compétences).

La figure ci-dessus résume le processus de vérification de la conformité. Des experts proposent des règles de conformités exprimées via une notation graphique. Celle-ci permet à la fois de simplifier la définition des règles mais aussi la communication des règles aux auteurs. A la notation graphique correspond un langage formel qui permet la traduction des règles dans le langage F-logique. Ainsi, l'auteur peut soumettre son objet d'apprentissage pour une vérification automatique de conformité.

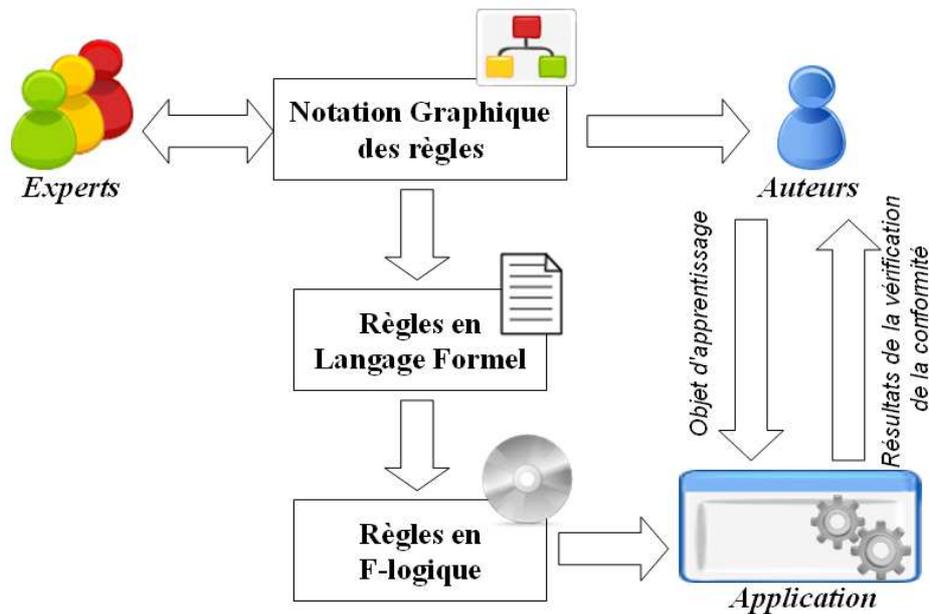


Figure 35 : Différents formats de description des règles de conformité

2.2. Règles de conformité

Les règles de conformité peuvent être définies par un ou plusieurs intervenants. Par exemple, il se peut que certaines règles soient définies au niveau du rectorat, d'autres au niveau de

l'institut d'autres au niveau du département et d'autres par l'auteur lui-même. Il se peut aussi que toutes les règles soient définies par une seule autorité.

Les règles qui peuvent être définies par rapport à un objet d'apprentissage font référence à la description que l'on peut observer. Ainsi, ces règles se basent sur les métriques introduites au niveau de la phase d'analyse et plus généralement sur les métadonnées des objets ainsi que leurs graphes de composition. Rappelons que nous avons classé ces règles en quatre catégories : les règles d'annotation, les règles structurelles, les règles sémantiques et les règles hybrides.

Afin de décrire les règles de conformité il faut définir un langage formel suffisamment expressif pour pouvoir décrire les quatre catégories de règles et flexible pour pouvoir s'adapter aux critères de qualité des experts et auteurs. En effet, on est loin d'avoir une seule vision de la qualité pour les objets d'apprentissage. Dans son rapport intitulé « la qualité en e-learning » [Ehlers et al., 2006] le centre européen pour le développement de la formation professionnelle affirme que 35% des organisations interrogées sur la qualité en e-learning utilisent leurs propres approches alors que 26% utilisent des approches externes. En fait, cette étude montre que 24% des interrogés travaillent dans des organisations qui laissent à leurs collaborateurs le soin de développer la qualité.

2.3. Notation graphique pour les règles

Il est important de prévoir un moyen permettant de définir (pour les experts) et de présenter (pour les auteurs) de manière compréhensible les règles de conformité. La compréhension de ces règles par les auteurs aura comme conséquence qu'il les prendra mieux en compte lors des phases de sélection d'objets à réutiliser et de construction du graphe de composition abstrait. Ceci évitera un nombre élevé d'itérations avant de mettre en place un objet d'apprentissage conforme.

Proposer une notation graphique représente un moyen approprié pour répondre à ce besoin. Cette notation doit offrir à la fois deux propriétés nécessaires pour communiquer clairement les règles même pour les non spécialistes. La première est qu'elle doit représenter les règles d'une façon organisée selon leurs natures : règles hybrides, règles sémantiques, règles structurelles et règles sur les métadonnées. La deuxième est qu'elle doit présenter les règles dans un format succinct et précis compréhensible par les auteurs.

2.4. Vérification automatique des règles

Faire vérifier manuellement la conformité d'un objet d'apprentissage composé par rapport aux règles de conformité présente plusieurs inconvénients. En effet, d'une part nous avons le risque de se tromper ou d'oublier la vérification d'une ou de plusieurs règles et d'autre part c'est une tâche fastidieuse et consommatrice de temps.

Ceci justifie le recours à l'automatisation de cette phase de notre approche afin d'assister les auteurs dans la vérification de leurs objets aux règles de conformité. Pour y parvenir nous avons fait appel aux techniques de la programmation logique afin de pouvoir décider si un objet est en conformité ou pas. Ainsi, les propriétés de l'objet d'apprentissage composé doivent être décrites par des faits et les règles de conformité par des règles logiques. Ensuite, le moteur d'inférence peut décider automatiquement si l'objet est conforme à toutes les règles ou non.

3. Règles de conformité

3.1. Règles pour les métadonnées

Les règles portant sur des métadonnées éducatives (ou règles d'annotation) permettent de vérifier la conformité à un profil d'application LOM. Les vérifications permettent de s'assurer que certaines entrées ou catégories obligatoires sont renseignées et qu'il y en a assez de métadonnées éducatives pour que l'objet soit accessible aux utilisateurs. Ces règles ont été classifiées en trois familles qui peuvent être utilisées séparément ou sous forme d'une combinaison.

3.1.1. Catégories LOM obligatoires

Les métadonnées, comme décrites dans le standard LOM [IEEE, 2002], sont organisées en catégories. En effet, dans le cas de ce standard les éléments à renseigner sont organisés en neuf catégories. Une catégorie est un groupe d'éléments en relation [IEEE, 2002].

Dans un profil d'application LOM on peut procéder par catégorie pour définir ce qui est obligatoire ou pas. Voici un exemple de règles pour imposer le renseignement de certaines catégories de métadonnées :

RM_1 : La catégorie « *General* » est obligatoire.

RM_2 : La catégorie « *Technical* » est obligatoire.

3.1.2. *Éléments LOM obligatoires*

Les profils d'application LOM les plus connus sont définis élément par élément. En effet, pour chaque élément LOM on indique s'il est obligatoire ou optionnel. Pour vérifier la conformité d'un objet d'apprentissage à un profil d'application défini de cette façon il faut définir une règle par élément obligatoire. Voici un exemple :

RM_3 : L'élément « Title » est obligatoire.

3.1.3. *Règles quantitatives*

La troisième famille de règles sur les métadonnées traite l'aspect quantitatif. Il ne s'agit plus de vérifier la conformité à un profil d'application LOM mais de s'assurer que l'auteur a renseigné assez de métadonnées éducatives pour que l'objet d'apprentissage soit accessible et qu'il puisse être découvert suite à une requête de la part de l'utilisateur. Voici deux exemples de règles, dont l'une a comme portée une catégorie LOM alors que l'autre a comme portée toutes les métadonnées :

RM_4 : Au moins 50% des éléments de la catégorie « Educational » doivent être renseignés.

RM_5 : Au moins 20% des éléments de métadonnées éducatives doivent être renseignés.

3.2. Règles structurelles

3.2.1. *Forme du graphe*

Au niveau structurel chaque objet d'apprentissage composé est décrit par un graphe de composition abstrait. Ce graphe est directement conçu par l'auteur de l'objet et sa forme reflète une stratégie de navigation au sein du contenu éducatif.

Une stratégie de navigation peut favoriser les accès alternatifs, les accès parallèles aux ressources. Elle permet également à l'auteur de définir un ordre de navigation permettant d'indiquer quelle ressource doit être étudiée en premier afin d'assimiler au mieux les suivantes.

Ainsi, d'un point de vue métier un ensemble de règles peuvent être proposées à ce niveau. En fait, pour augmenter le degré de personnalisation au niveau d'un objet d'apprentissage composé il faut s'assurer que l'auteur a conçu le graphe de composition avec assez d'opérateurs d'alternatives (ALT). D'autre part, pour permettre à l'apprenant d'avoir une certaine liberté dans le choix du chemin de navigation il faut s'assurer que l'auteur a introduit assez d'opérateurs parallèles (PAR) au sein du graphe.

Voici un exemple de règles structurelles qui permettent d'influencer la stratégie de navigation :

***RS_1** : Au moins deux opérateurs ALT doivent être présent au sein du graphe de composition abstrait.*

***RS_2** : Au moins un opérateur PAR doit être présent au sein du graphe de composition abstrait.*

Ces différentes règles peuvent être également décrites pour le graphe de composition concret. Rappelons que celui-ci est obtenu suite au remplacement de chaque objet d'apprentissage composé au sein du graphe de composition par son graphe de composition abstrait de façon récursive. Ce graphe ne comporte que des objets d'apprentissage non composés (ou atomiques). La stratégie de navigation à ce niveau est une combinaison de la stratégie décrite par l'auteur au niveau du graphe de composition abstrait et des stratégies de navigation des auteurs des objets d'apprentissage composés réutilisés. L'usage de règles à ce niveau est justifié par le fait de vouloir avoir un certain contrôle sur les propriétés du graphe de composition concret qui présente une facette de l'objet d'apprentissage non directement visible pour l'auteur. A titre d'exemples, voici quelques règles illustratives :

***RS_3** : Au plus cinq opérateurs ALT doivent être présent au sein du graphe de composition concret.*

***RS_4** : Au plus cinq opérateurs PAR doivent être présent au sein du graphe de composition concret.*

Le graphe concret peut se décomposer en un ensemble de graphes affichables qui peuvent être empruntés par les apprenants qui vont accéder à cet objet d'apprentissage. Certaines portions sont communes à tous les graphes affichables d'un objet d'apprentissage composé alors que d'autres parties sont spécifiques et s'adaptent mieux à certains profils d'apprenants. Le nombre de graphes affichables représente un aspect significatif d'un point de vue métier qui peut être contrôlé via des règles. Par exemple :

***RS_5** : Au moins trois graphes affichables par objet d'apprentissage composé.*

***RS_6** : Au plus six graphes affichables par objet d'apprentissage composé.*

3.2.2. Dimensions et densité du graphe de composition

Les règles structurelles peuvent également porter sur la taille du graphe permettant ainsi de contrôler ses dimensions ainsi que sa densité. Ceci permet de traduire certaines règles métier

qui concernent le volume de contenu dispensé aux apprenants via un objet d'apprentissage composé.

La taille du graphe est exprimée en termes de largeur et de profondeur. La profondeur d'un graphe correspond au plus long chemin. La largeur du graphe correspond à la plus grande distance entre deux sommets. Les distances sont mesurées en nombre d'arcs. Des règles portant sur les dimensions d'un graphe de composition peuvent être formulées par exemple comme suit :

RS_7 : la profondeur du graphe de composition abstrait ne doit pas dépasser vingt.

RS_8 : la largeur du graphe de composition concret ne doit pas dépasser cinq.

La densité du graphe en terme de nombre de sommets de type objets d'apprentissage est une propriété supplémentaire qui peut être contrôlée au niveau des différents graphes associés à un objet d'apprentissage composé. Des règles comme les suivantes peuvent être formulées :

RS_9 : Le nombre de sommets de type objet d'apprentissage ne doit pas dépasser 50 au niveau du graphe de composition concret.

RS_10 : Le nombre de sommets de type objet d'apprentissage au niveau de chaque graphe affichable ne doit pas dépasser vingt.

Les graphes affichables représentent pour les apprenants la partie visible de l'objet d'apprentissage composé. Ceci justifie le fait de vouloir s'assurer que certaines règles ont été respectées. A ce niveau, les règles peuvent imposer une certaine similarité entre les graphes affichables. A titre d'exemples voici deux règles illustratives :

RS_11 : La différence entre la plus petite profondeur et la plus grande profondeur des différents graphes affichables ne doit pas dépasser 30%.

RS_12 : La différence de densité en sommets entre le graphe affichable le plus dense et le moins dense ne doit pas dépasser 30%.

3.2.3. Niveaux d'abstraction

Les règles peuvent porter sur des propriétés liées aux niveaux d'abstraction du graphe de composition d'un objet d'apprentissage composé. Le nombre de niveau d'abstraction peut être limité pour éviter le risque d'une discordance structurelle importante entre le graphe de composition abstrait et le graphe de composition concret qui peut influencer de façon négative la stratégie de navigation choisie par l'auteur. La règle suivante permet d'exprimer ceci :

RS_13 : *Le nombre de niveaux d'abstraction ne doit pas dépasser quatre.*

Certaines règles peuvent limiter les aspects en relation avec la discordance structurelle. En effet, il est possible de cibler des propriétés précises. Les différences en termes de nombre de sommets, d'opérateurs PAR et d'opérateurs ALT entre le graphe de composition abstrait et le graphe de composition concret peuvent être ainsi contrôlées. Les différences entre ces deux graphes en termes de profondeur, de largeur et de densité peuvent également être contrôlées. Voici des exemples de règles :

RS_14 : *Le nombre de sommets du graphe de composition concret ne doit pas dépasser le double du nombre de sommets du graphe de composition abstrait.*

3.2.4. Similarité des graphes affichables

A un même objet d'apprentissage composé est associé un certain nombre de graphes affichables qui correspondent aux différents parcours possibles du graphe de composition concret. Ces graphes affichables sont en quelque sorte des versions différentes d'un même objet. Ainsi, ils doivent présenter une certaine similarité. Cette similarité peut être contrôlée via des règles spécifiques. Ces règles doivent spécifier le degré de similarité en donnant son étendue minimum. Voici quelques exemples de règles :

RS_15 : *Tous les graphes affichables d'un objet d'apprentissage composé doivent avoir en commun au moins les trois premiers sommets.*

RS_16 : *Tous les graphes affichables doivent avoir en commun au moins 50% des leurs sommets (sommets du graphe de composition concret sans les opérateurs ALT).*

3.3. Règles sémantiques

Concernant les aspects sémantiques les règles peuvent porter sur le contenu, les pré-requis et la fonction d'acquisition. Au niveau du contenu et pour des considérations pédagogiques il est intéressant de pouvoir s'assurer que le contenu n'est pas trop chargé en concepts, que certains rôles éducatifs sont présents et éventuellement que certains rôles éducatifs sont assurés pour chaque concept traité. Quant aux pré-requis, il est généralement souhaitable qu'ils ne soient pas trop nombreux ni trop restrictifs. En effet, ceci peut rendre l'objet inaccessible à la plupart des apprenants et donc aller à l'encontre de son rôle éducatif. Finalement, la fonction d'acquisition ne doit pas être trop forte en offrant à l'apprenant un niveau élevé sur plusieurs concepts et plusieurs rôles simultanément.

3.3.1. Contenu

Le contenu d'un objet d'apprentissage est décrit par un ensemble de couples <concept, rôle>. A ce niveau le nombre de concepts différents qui sont traités peut être contrôlé afin d'éviter un contenu trop chargé. Ceci doit permettre de favoriser une meilleure assimilation par les apprenants. Ceci peut être exprimé via des règles semblables à celle-ci :

***RSe_1** : Le nombre de concepts traités au niveau du contenu ne doit pas dépasser cinq.*

Les rôles éducatifs assurés par l'objet d'apprentissage composé peuvent être également contrôlés pour s'assurer par exemple de la présence de certains rôles spécifiques. Voici un exemple de règles de ce type :

***RSe_2** : Les rôles « exercice » et « exemple » doivent être présents au niveau du contenu.*

Il est également envisageable de vouloir s'assurer de la présence d'un ou de plusieurs rôles pour chaque concept traité au niveau de l'objet d'apprentissage composé. Voici une règle de ce type à titre d'exemple :

***RSe_3** : Le rôle « définition » doit être assuré pour chaque concept dans le contenu.*

Il est possible également de procéder d'une façon plus générale en définissant une règle limitant le nombre de doublets qui décrivent le contenu. Voici un exemple :

***RSe_4** : le nombre d'entrées qui décrivent le contenu ne doivent pas dépasser dix.*

3.3.2. Pré-requis

Pour les pré-requis ils sont décrit par des triplets de la forme <concept, rôle, niveau>. S'assurer que les pré-requis ne sont pas trop restrictifs peut être contrôlé en appliquant certaines règles.

Le nombre de concept traités au niveau des pré-requis peut faire le sujet d'une règle permettant de s'assurer qu'il ne dépasse pas un certain seuil. Par exemple, voici un exemple d'une règle qui put être formulée afin d'assurer cette propriété :

***RSe_5** : Le nombre de concepts différents demandés au niveau des pré-requis ne doit pas dépasser cinq.*

Le nombre de rôles différents demandés au niveau des pré-requis peut également être contrôlé pour s'assurer que l'objet reste accessible aux apprenants. Il est ainsi possible de limiter le nombre de rôles différents pour mettre un seuil supérieur. Voici un exemple de formulation d'une telle règle :

***RSe_6** : Le nombre de rôles éducatifs différents demandés au niveau des pré-requis ne doit pas dépasser trois.*

Le niveau de maîtrise d'un concept suivant un certain rôle, décrit par un triplet au niveau des pré-requis, permet d'indiquer si l'objet sera accessible à un nombre restreint ou important d'apprenants. En effet, si le niveau de maîtrise demandé est élevé alors moins d'apprenants seront en mesure d'accéder à l'objet d'apprentissage. Certaines règles peuvent contrôler cet aspect comme la règle suivante :

***RSe_7** : Le nombre de pré-requis avec un niveau de maîtrise « élevé » ne doit pas dépasser trois.*

Le nombre de pré-requis de façon générale peut également être contrôlé via une règle qui limite le nombre d'entrées. Voici un exemple :

***RSe_8** : Le nombre d'entrées qui décrivent les pré-requis nécessaires à l'apprenant ne doit pas dépasser cinq.*

3.3.3. Fonction d'acquisition

La fonction d'acquisition est un ensemble de triplets <concept, rôle, niveau> qui représente ce qu'un apprenant a acquis grâce au contenu d'un objet d'apprentissage. Cette information permet de mettre-à-jour le profil de l'apprenant.

Ce qui peut être contrôlé à ce niveau c'est principalement l'aspect quantitatif. En fait, proposer trop de connaissances à la fois à l'apprenant est souvent considéré comme étant à l'encontre d'une meilleure assimilation des concepts traités par le contenu. Ceci peut être contrôlé via une règle de la forme suivante :

***RSe_9** : Le nombre de triplets dans la fonction d'acquisition ne doit pas dépasser cinq.*

Le contrôlé peut être raffiné en se focalisant sur le nombre de triplets dont le niveau de connaissance est « élevé ». Par exemple, voici une règle qui permet d'appliquer ce type de contrôle :

***RSe_10** : Le nombre de triplets dans la fonction d'acquisition ayant comme niveau de connaissance « élevé » ne doit pas dépasser trois.*

3.4. Règles hybrides

Les règles hybrides représentent une façon plus expressive pour contrôler des aspects qui mettent en jeu la sémantique et la structure simultanément. Ainsi, des règles didactiques plus

élaborées que celles purement structurelles ou purement sémantiques peuvent être exprimées et contrôlées.

3.4.1. Rôles du sommet racine et des sommets feuilles

Le sommet racine et les sommets feuilles ont une importance particulière. En effet, le sommet racine est le premier objet d'apprentissage qui sera offert aux apprenants. Concernant les sommets feuilles sont les derniers objets d'apprentissage qui sont offerts aux apprenants. Il est donc intéressant de vouloir leur imposer des rôles précis via des règles spécifiques. Voici deux règles à titre d'exemple :

RH_1 : *Le sommet racine doit avoir le rôle « introduction ».*

RH_2 : *Chaque sommet feuille doit avoir le rôle « exercice ».*

3.4.2. Patron de conception

Les règles de conception peuvent être très strictes en imposant un patron de conception commun aux objets d'apprentissage avec cependant une certaine marge de liberté. Un patron de conception représente une structure « flexible » du graphe de composition couplée avec des règles portant sur le contenu. Ceci permet d'avoir des objets d'apprentissage très homogènes et qui présente une structure et une sémantique homogènes. Voici deux patrons de conception possibles :

RH_3 : *L'objet doit commencer par un objet d'apprentissage simple avec le rôle « Introduction » suivi par un objet d'apprentissage avec le rôle « présentation ». Il doit être suivi par trois alternatives d'objets d'apprentissage ayant le rôle « Exemple ». Ensuite, il faut un accès parallèle à deux objets d'apprentissage avec le rôle « Exercice ». Puis, l'objet doit présenter un objet d'apprentissage avec le rôle « Conclusion ».*

RH_4 : *L'objet doit commencer par une séquence de trois objets d'apprentissage simples et se terminer par un objet d'apprentissage avec le rôle « Conclusion ».*

4. Langage formel pour la description des règles

4.1. Éléments du langage

Afin de pouvoir décrire les règles d'une façon formelle il faut définir un langage capable de supporter les différentes règles de conformité. Ainsi, nous avons défini un langage qui suit la grammaire décrite ci-dessous.

En fait, chaque ensemble de règles est décrit par un identificateur, éventuellement une spécification du graphe de composition abstrait et une ou plusieurs règles de conformité.

Rules := Identifieur = [Graph_specification ;] [Rule ;]^+

Le graphe est décrit par l'ensemble de ses arcs. Chaque arc est décrit par un couple de sommets. Le premier est le sommet de départ et le deuxième c'est le sommet d'arrivée.

*Graph_specification := Edge [, Edge]**

Edge := Node Node

Chaque sommet peut représenter l'un des éléments suivants : un objet d'apprentissage (quelconque, simple ou composé), un opérateur de graphe ou un sommet générique ANY qui représente tout graphe valide. Les opérateurs de graphe sont ALT pour les alternatives, PAR pour les accès parallèles et ANY_Op qui représente l'un ou l'autre des deux opérateurs.

Node := Id_Node : LO / SLO / CLO / Graph_Op / ANY

Graph_Op := ALT / PAR / ANY_Op

Les règles sont simples ou complexes. Chaque règle simple consiste en une fonction dont le résultat est comparé à une constante ou au résultat de l'évaluation d'une autre fonction. Une règle complexe est formée d'une quantification qui porte sur un ensemble généré par une fonction et qui doit vérifier soit une propriété décrite par un quantificateur qui porte sur un ensemble généré également par une fonction, soit une comparaison entre le résultat d'une fonction qui s'applique sur l'ensemble généré et une constante.

Rule := SimpleRule / ComplexRule

SimpleRule := LearningObject->Characteristic->Function Comparator [Function / Constant]

LerningObject := LO / Id_Node

Characteristic := Metadata / AbstractCG / ConcreteCG / DeliveryG / Prerequisite / Content / AcquisitionFunction

Comparator := = / ≠ / < / > / ≤ / ≥ / IN / NOT IN / ⊇ / ⊆

ComplexRule := Quantification Function => [Quantification Function / Function Comparator Constant]

Quantification := ∀ var IN / ∃ var IN

Constant := string or numbers or booleans

4.2. Fonctions

Chaque fonction a un nom et d'éventuels paramètres donnés entre parenthèse. Chaque fonction retourne un résultat et s'applique à une caractéristique d'un objet d'apprentissage :

$LearningObject \rightarrow Characteristic \rightarrow FunctionName ([Parameter [, Parameter]^*])$
--

L'objet d'apprentissage est soit l'objet d'apprentissage composé sujet de la vérification de la conformité ou un objet d'apprentissage utilisé dans le graphe de composition abstrait. Dans le premier cas il faut noter l'objet par LO (représente l'objet en cours de validation) et dans le deuxième cas il faut donner l'identifiant du sommet qui représente l'objet au sein du graphe.

La liste des caractéristiques sur lesquelles portent les règles est la suivante :

« **Metadata** » : désigne l'ensemble des métadonnées informationnelles de l'objet d'apprentissage.

« **AbstractCG** » : désigne le graphe de composition abstrait de l'objet d'apprentissage.

« **ConcreteCG** » : désigne le graphe de composition concret de l'objet d'apprentissage.

« **DeliveryG** » : désigne l'ensemble de graphes affichables de l'objet d'apprentissage.

« **Content** » : désigne l'ensemble des doublets <contenu, rôle> qui donnent une description sémantique du contenu de l'objet d'apprentissage.

« **Prerequisites** » : désigne l'ensemble des triplets <contenu, rôle, niveau> qui donnent une description sémantique des pré-requis nécessaires pour aborder cet objet d'apprentissage.

« **AcquisitionFunction** » : désigne l'ensemble des triplets <contenu, rôle, niveau> qui donnent une description sémantique des acquis apportés au profil de l'apprenant par cet objet d'apprentissage.

Nous allons introduire par la suite les différentes fonctions ainsi que des exemples de leur utilisation pour chaque famille de règles.

4.2.1. Fonctions pour les règles de métadonnées

Nous allons utiliser la fonction *list(<paramètre>)* avec la caractéristique « Metadata ». Elle peut prendre comme paramètre le terme « Categories » et alors elle génère l'ensemble des catégories de métadonnées renseignées au sein de l'objet. Sinon, elle prend comme paramètre le terme « Entries » afin de générer l'ensemble des entrées de métadonnées renseignées.

A titre d'exemples voici comment on note les règles présentées dans la section précédente :

RM_1 : *LO -> Metadata -> list('Categories') \supseteq {'General'}*

RM_2 : *LO -> Metadata -> list('Categories') \supseteq {'Technical'}*

RM_3 : *LO -> Metadata -> list('Entries') \supseteq {'Title'}*

Pour les règles quantitatives qui portent sur le pourcentage des entrées de métadonnées renseignées nous allons utiliser la fonction **percentageOfFilledEntries**(<paramètre>) avec la caractéristique « Metadata ». Elle donne un pourcentage. S'il s'agit du pourcentage par rapport à tous les entrées, le paramètre à indiquer est « All ». Par contre s'il s'agit du pourcentage par rapport à tous les entrées d'une catégorie le paramètre est alors le nom de la catégorie de métadonnées.

RM_4 : *LO -> Metadata -> percentageOfFilledEntries('Educational') \geq 50%*

RM_5 : *LO -> Metadata -> percentageOfFilledEntries('All') \geq 20%*

4.2.2. Fonctions pour les règles structurelles

Les règles structurelles portent sur trois caractéristiques : le graphe de composition abstrait (**AbstractCG**), le graphe de composition concret (**ConcreteCG**) et les graphes de composition (**DeliveryG**).

Une première fonction qui s'applique à tous ces caractéristiques est la fonction **count**(<paramètre>). Elle permet de comptabiliser les opérateurs d'alternatives au sein du graphe si 'ALT' est donné comme paramètre. Elle permet de comptabiliser le nombre d'opérateurs parallèle si le paramètre donné est 'PAR'. Si le paramètre donné est 'LO' ou 'SLO' ou 'CLO' alors elle comptabilise respectivement le nombre d'objets d'apprentissage ou le nombre d'objet d'apprentissage simples ou le nombre d'objets d'apprentissage composés au sein du graphe. Finalement, le paramètre 'ALL' permet de comptabiliser le nombre de sommets au sein du graphe.

Grace à cette fonction les règles suivantes peuvent être exprimées :

RS_1 : *LO -> AbstractCG -> count('ALT') \geq 2*

RS_2 : *LO -> AbstractCG -> count('PAR') \geq 1*

RS_3 : *LO -> ConcreteCG -> count('ALT') \leq 5*

RS_4 : *LO -> ConcreteCG -> count('PAR') \leq 5*

RS_9 : *LO* -> *ConcreteCG* -> *count('LO')* ≤ 50

RS_10 : *LO* -> *DeliveryG* -> *count('LO')* ≤ 20

Deux fonctions sans paramètres ***width()*** et ***depth()*** permettent de retourner respectivement la largeur et la profondeur d'une des trois catégories de graphes associés à un objet d'apprentissage composé.

Grace à cette fonction les règles suivantes peuvent être exprimées :

RS_7 : *LO* -> *AbstractCG* -> *depth()* ≤ 20

RS_8 : *LO* -> *AbstractCG* -> *width()* ≤ 5

Les graphes affichables ont leurs propres fonctions. La fonction ***number()*** permet de donner le nombre de graphes affichables. Alors que la fonction ***difference(<paramètre>)*** permet de retourner le pourcentage de différence entre la valeur la plus petite et la plus grande entre les graphes affichables par rapport à l'un des paramètres suivants : 'Depth' pour la profondeur, 'Width' pour la largeur, 'LO' pour le nombre d'objets d'apprentissage, 'PAR' pour le nombre d'opérateurs parallèle et 'ALL' pour le nombre total de sommets.

La fonction ***similarity(<paramètre>)*** permet de retourner suivant le paramètre donné une évaluation concernant la similarité entre les graphes affichables. Si le paramètre donné est 'Beginning' la fonction retourne le nombre des sommets qu'on trouve au début de tous les graphes affichables. Si le paramètre donné est 'ALL' alors la fonction doit retourner le pourcentage de sommets partagés entre les graphes affichables par rapport à l'ensemble de leurs sommets. Ses sommets sont ceux du graphe de composition concret sans les opérateurs ALT.

Grace à ces fonctions il est possible d'exprimer les contraintes suivantes qui portent sur les graphes affichables :

RS_5 : *LO* -> *DeliveryGraph* -> *number()* ≥ 3

RS_6 : *LO* -> *DeliveryGraph* -> *number()* ≤ 6

RS_11 : *LO* -> *DeliveryGraph* -> *difference('Depth')* ≤ 30%

RS_12 : *LO* -> *DeliveryGraph* -> *difference('ALL')* ≤ 30%

RS_15 : *LO* -> *DeliveryGraph* -> *similarity('Beginning')* ≥ 3

RS_16 : *LO* -> *DeliveryGraph* -> *similarity('ALL')* ≥ 50%

Le graphe de composition abstrait dispose de deux fonctions spécifiques. La première est la fonction *numberOfAbstractionLevels()*. Cette fonction donne le nombre de niveaux d'abstraction. La deuxième est la fonction *similarityToConcreteCG()* qui permet de donner le pourcentage de similarité entre le graphe de composition abstrait et le graphe de composition concret en termes d'objets d'apprentissage.

Ces deux fonctions permettent d'exprimer les règles suivantes :

RS_13 : LO -> AbstractCG -> numberOfAbstractionLevels() ≤ 4

RS_14 : LO -> AbstractCG -> similarityToConcreteCG() ≤ 50%

4.2.3. Fonctions pour les règles sémantiques

Les règles sémantiques portent sur trois caractéristiques : le contenu (*Content*), les pré-requis (*Prerequisites*) et la fonction d'acquisition (*AcquisitionFunction*).

Une première fonction qui s'applique à tous ces caractéristiques est la fonction *number()* qui permet de retourner le nombre d'entrées relatives à chacune des trois catégories de métadonnées sémantiques.

Cette fonction permet d'exprimer les règles suivantes :

RSe_4 : LO -> Content -> number() ≤ 10

RSe_8 : LO -> Prerequisites -> number() ≤ 5

RSe_9 : LO -> AcquisitionFunction -> number() ≤ 5

Une deuxième fonction applicable à ces caractéristiques est la fonction *count(<paramètre>)*. Si le paramètre est 'Concept' ou 'Role' alors la fonction retourne respectivement le nombre de concepts ou le nombre de rôles distincts au sein de la catégorie de métadonnées considérée.

RSe_1 : LO -> Content -> count('Concept') ≤ 5

RSe_5 : LO -> Prerequisites -> count('Concept') ≤ 5

RSe_6 : LO -> Prerequisites -> count('Role') ≤ 3

Nous proposons également la fonction *list(<paramètre>)* qui permet selon le paramètre de lister soit l'ensemble des concepts avec le paramètre 'Concept', soit l'ensemble des rôles avec le paramètre 'Role', soit l'ensemble de niveaux avec le paramètre 'Level'.

Cette fonction permet d'exprimer des règles de conformité comme la suivante :

RSe_2 : $LO \rightarrow Content \rightarrow list('Role') \supseteq \{ 'Exercice', 'Exemple' \}$

RSe_3 : $\forall x \text{ IN } LO \rightarrow Content \rightarrow list('Concept') \Rightarrow \exists \langle x, 'Définition' \rangle \text{ IN } LO \rightarrow Content \rightarrow list()$

RSe_7 : $\forall \langle x, y, 'Elevé' \rangle \text{ IN } LO \rightarrow Prerequisites \rightarrow list() \Rightarrow count() \leq 3$

RSe_10 : $\forall \langle x, y, 'Elevé' \rangle \text{ IN } LO \rightarrow AcquisitionFunction \rightarrow list() \Rightarrow count() \leq 3$

En fait les règles RSe_3, RSe_7 et RSe_10 sont des exemples de règles complexe où on fait appel aux quantificateurs.

4.2.4. Formulation des règles hybrides

Les règles hybrides associent à la fois des aspects structurels et sémantiques. Elles ont une forte capacité expressive. Voici comment les règles hybrides données comme exemple précédemment sont exprimées :

RH_1 : *Le sommet racine doit avoir le rôle « introduction ».*

$R_{001} =$

$N1 : LO \quad N2 : ANY ;$

...

$N1 \rightarrow Content \rightarrow list (Role) \supseteq \{ 'Introduction' \},$

...

RH_2 : *Chaque sommet feuille doit avoir le rôle « exercice ».*

$R_{002} =$

$N1 : ANY \quad N2 : LO$

...

$N2 \rightarrow Content \rightarrow list (Role) \supseteq \{ 'Conclusion' \}$

...

RH_3 : *L'objet doit commencer par objet d'apprentissage simple avec le rôle « Introduction » suivi par un objet d'apprentissage avec le rôle « présentation ». Il doit être suivi par trois alternatives d'objets d'apprentissage ayant le rôle « Exemple ». Ensuite, il faut un accès parallèle à deux objets d'apprentissage avec le rôle « Exercice ». Puis, l'objet doit présenter un objet d'apprentissage avec le rôle « Conclusion ».*

$R_{003} =$

N1 : SLO N2 : LO,
N2 : LO N3 : ALT,
N3 : ALT N4 : LO,
N3 : ALT N5 : LO,
N3 : ALT N6 : LO,
N4 : LO N7 : PAR,
N5 : LO N7 : PAR,
N6 : LO N7 : PAR,
N7 : PAR N8 : LO,
N7 : PAR N9 : LO,
N8 : LO N10 : LO,
N9 : LO N10 : LO ;
N1 -> Content -> list (Role) ⊇ {‘Introduction’},
N2 -> Content -> list (Role) ⊇ {‘Présentation’},
N4 -> Content -> list (Role) ⊇ {‘Exemple’},
N5 -> Content -> list (Role) ⊇ {‘Exemple’},
N6 -> Content -> list (Role) ⊇ {‘Exemple’},
N8 -> Content -> list (Role) ⊇ {‘Exercice’},
N9 -> Content -> list (Role) ⊇ {‘Exercice’},
N10 -> Content -> list (Role) ⊇ {‘Conclusion’} ;

RH_4 : *L’objet doit commencer par une séquence de trois objets d’apprentissage simples et se terminer par un objet d’apprentissage avec le rôle « Conclusion ».*

R_004 =

N1 : SLO N2 : SLO,
N2 : SLO N3 : SLO,
N3 : SLO N4 : ANY,
N4 : ANY N5 : LO ;
N5 -> Content -> list (Role) ⊇ {‘Conclusion’} ;

4.3. Expressivité du langage

Nous avons étudié l’expressivité du langage par rapport à nos objectifs, et nous avons réussi à formuler toute les règles qui nous intéressent. Toutefois, il faut noter qu’il ne supporte pas des règles qui intègrent des contraintes conditionnelles.

Exemple :

« Si le nombre d'objets d'apprentissage d'objets dépasse dix au niveau du graphe de composition abstrait alors le nombre d'opérateurs ALT au niveau de ce graphe ne doit dépasser trois. »

Ceci ne présente pas une limite par rapport à notre contexte d'utilisation du moment qu'il est possible de définir plusieurs ensembles de contraintes. Chaque ensemble peut être conçu pour une catégorie particulière d'objets d'apprentissage. En effet, nous pouvons par exemple considérer que les objets d'apprentissage dont le graphe de composition comprend plus que dix objets sont des objets de grande taille. Et il est possible de définir un ensemble de règles pour les objets de grande taille qui spécifie que le nombre d'opérateurs ALT doit être supérieur à trois. Ce profil peut être formulé de la façon suivante :

<pre>R_005 = ... LO -> AbstractCG -> count ('LO') > 10, LO -> AbstractCG -> count ('ALT') > 3, ...</pre>
--

Dans cet extrait de profil on peut constater qu'uniquement les objets d'apprentissage qui comportent plus que dix objets au niveau de leurs graphes de composition abstraits sont traités par le reste des règles de conformité. Et ils ne sont considérés comme valides que si le nombre d'opérateurs ALT est supérieur à trois.

Les règles itératives sont une autre forme de règles non supportées par notre langage formel. En effet, certaines règles que nous pouvons énoncer nécessitent l'exécution d'un traitement itératif.

Exemple :

« Le graphe de composition abstrait ne doit pas comporter de cycles. »

Cette règle ne peut être exprimée avec notre langage formel. En fait, plusieurs règles de validité par rapport au modèle SIMBAD nécessitent des traitements itératifs. Ceci explique le fait que nous avons isolé ces règles au niveau d'une phase à part et que nous l'avons pas considéré comme une forme de règles de conformité.

Il faut également noter que les règles doivent être basées sur des propriétés de l'objet d'apprentissage. Et donc certaines règles qui touchent à des propriétés relatives par exemple à

l'apprenant ou au modèle de domaine (et donc pas à l'objet d'apprentissage) ne peuvent pas être exprimées.

Exemple :

Tous les concepts présents dans l'objet d'apprentissage composé doivent être d'un niveau inférieur à trois dans le modèle de domaine (i.e. des concepts de haut niveau).

En fait, prendre en compte dans le langage formel des entités autres que les objets d'apprentissage, comporte plus de risques que d'avantages. En effet, le langage risque de devenir trop complexe alors que ce type de règles semble trop restrictif et pas fondamental par rapport à notre objectif de départ.

5. Notation graphique

5.1. Objectif

Communiquer les règles à respecter aux concepteurs des objets d'apprentissage composés est une étape importante pour permettre aux auteurs de comprendre les règles et donc d'anticiper dès la conception pour favoriser la production d'objets conformes.

Une représentation graphique des règles doit permettre d'abord de les organiser par catégorie. En plus, elle doit permettre de simplifier la formulation de ces règles grâce à une notation à la fois expressive et concise. Ceci doit éviter une mauvaise compréhension ou interprétation des règles.

Certaines des règles hybrides, et spécialement les patrons de conception, sont naturellement exprimées via la notation graphique. Nous définissons ci-après les éléments d'une représentation graphique qui tient compte de ces différents éléments.

5.2. Conteneur pour les règles

L'idée de la notation graphique est de structurer les règles autour du graphe de composition. Certaines règles portent sur l'objet tout entier alors que d'autres portent sur un sommet particulier du graphe.

Un conteneur va spécifier dans sa première partie (appelée règles hybrides) la structure du graphe de composition sur lequel il s'applique. Si cette partie est laissée vide c'est qu'il n'y a pas de contraintes particulières sur la forme du graphe de composition. Si une structure est donnée alors elle est décrite entre autres par des conteneurs.

La portée du conteneur englobant est l'objet d'apprentissage composé à contrôler. Alors que la portée d'un conteneur utilisé pour décrire la structure d'un graphe de composition est le sommet qu'il représente (le plus souvent le sommet racine ou un sommet feuille).

Les autres parties du conteneur contiennent les autres règles qui s'appliquent sur le graphe et classifiées en règles sémantiques, structurelles et sur d'annotation.

Au conteneur est associée une étiquette qui lui attribut un identifiant unique (il permet de le désigner d'une façon unique et donc l'ensemble de règles associées) et un type (objet simple (SLO), complexe(CLO), quelconque(LO) ou n'importe quel graphe valide (ANY)).

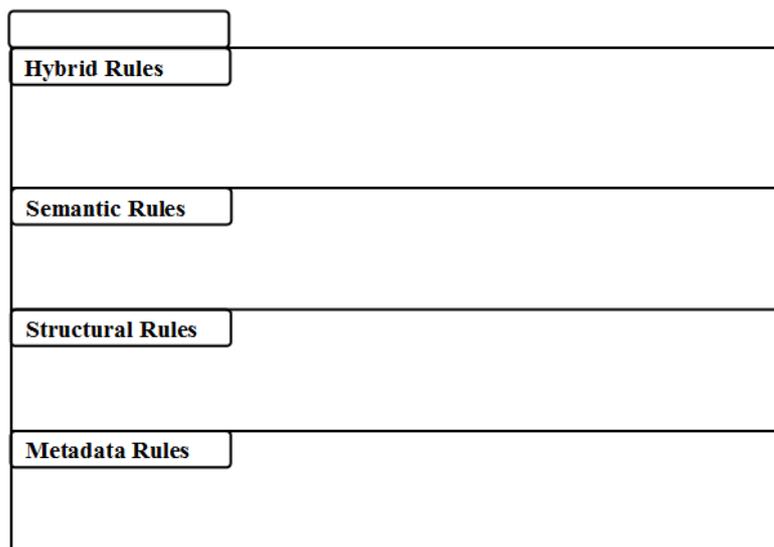


Figure 36 : Conteneur utilisé dans la notation graphique

5.3. Patrons de conception

L'expression des règles hybrides est basée sur une présentation graphique. Celle-ci contient uniquement deux formes : les conteneurs et les opérateurs. Ces derniers sont représentés par des formes circulaires. Chaque forme prend soit le libellé PAR pour représenter un opérateur « parallèle », soit le libellé ALT pour représenter l'opérateur « alternatif » ou le libellé ANY_Op pour représenter tout opérateur.

A titre d'exemple soit un ensemble de règles de conformité (R_150) qui comporte uniquement la règle hybride *RH_3*.

RH_3 : *L'objet doit commencer par objet d'apprentissage simple avec le rôle « Introduction » suivi par un objet d'apprentissage avec le rôle « présentation ». Il doit être suivi par trois alternatives d'objets d'apprentissage ayant le rôle « Exemple ». Ensuite, il faut*

un accès parallèle à deux objets d'apprentissage avec le rôle « Exercice ». Puis, l'objet doit présenter un objet d'apprentissage avec le rôle « Conclusion ».

Voici l'expression de R_150 en langage formel :

R_150= N1 : SLO N2 : LO, N2 : LO N3 : ALT, N3 : ALT N4 : LO,
 N3 : ALT N5 : LO, N3 : ALT N6 : LO, N4 : LO N7 : PAR,
 N5 : LO N7 : PAR, N6 : LO N7 : PAR, N7 : PAR N8 : LO,
 N7 : PAR N9 : LO, N8 : LO N10 : LO, N9 : LO N10 : LO ;

- N1 -> Content -> list (Role) \supseteq {'Introduction'},
- N2 -> Content -> list (Role) \supseteq {'Présentation'},
- N4 -> Content -> list (Role) \supseteq {'Exemple'},
- N5 -> Content -> list (Role) \supseteq {'Exemple'},
- N6 -> Content -> list (Role) \supseteq {'Exemple'},
- N8 -> Content -> list (Role) \supseteq {'Exercice'},
- N9 -> Content -> list (Role) \supseteq {'Exercice'},
- N10 -> Content -> list (Role) \supseteq {'Conclusion'} ;

Et voici la notation graphique pour cet exemple :

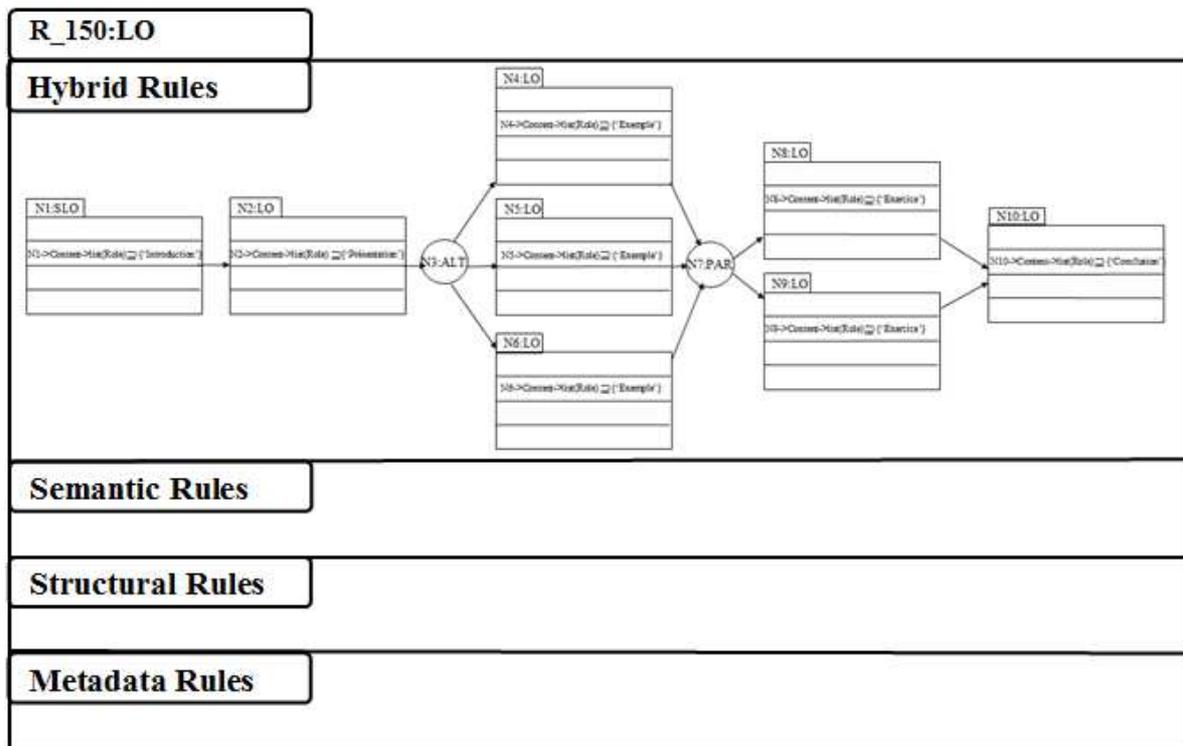


Figure 37 : Exemple de notation graphique pour les règles de conformité

5.4. Expressivité de la notation graphique

En fait, les règles sémantiques, structurelles et d'annotation sont exprimées de la même façon qu'avec le langage formel. Ainsi, on a à ce niveau le même niveau d'expressivité que le langage formel. L'apport de la notation graphique à ce niveau consiste en l'organisation de ces règles par catégorie afin de faciliter leur compréhension par les utilisateurs.

L'intérêt majeur de cette notation est l'expression des règles hybrides. Celles-ci se formulent beaucoup plus simplement autour du graphe de composition. Il reste donc à discuter de l'expressivité de la notation graphique par rapport aux règles hybrides, ce qui revient à discuter de la capacité à décrire tout patron de conception valide avec cette notation.

En fait, la description des règles hybrides repose sur une structure qui est décrite par un ensemble d'arcs. Chaque arc a un sommet de départ et un sommet d'arrivée. Et les sommets se déclinent en des objets d'apprentissage, des opérateurs ou de sommets génériques comme le montre cet extrait de la grammaire du langage formel :

$\begin{aligned} \text{Graph_specification} &:= \text{Edge } [, \text{Edge }]^* \\ \text{Edge} &:= \text{Node Node} \\ \text{Node} &:= \text{Id_Node} : \text{LO} / \text{SLO} / \text{CLO} / \text{Graph_Op} / \text{ANY} \\ \text{Graph_Op} &:= \text{ALT} / \text{PAR} / \text{ANY_Op} \end{aligned}$
--

La notation graphique permet de représenter tous ces éléments et de leur associer les caractéristiques que l'on souhaite. En effet, puisque la notation graphique est capable de représenter chaque type de sommet et ainsi que les arcs alors elle est capable d'exprimer toute règle hybride conformément à la grammaire du langage formel.

Cette équivalence entre la notation graphique et le langage formel permet de garantir la faisabilité de la traduction des règles de la première forme vers la deuxième et vice versa.

6. Exemple illustratif

Supposons qu'une université a décidé d'adopter une solution d'apprentissage en ligne pour ses étudiants et pour son personnel. Le système SIMBAD est retenu comme support de formation en ligne. Durant une période d'essai l'administration a constaté un taux d'abandon assez important. Un sondage fait auprès des apprenants a montré que la majorité des critiques concernent les objets d'apprentissage mis à leur disposition.

Cette université peut réagir en demandant à des experts en conception d'objets d'apprentissage et en pédagogie, de définir des règles auxquelles tout objet d'apprentissage doit être conforme pour qu'il soit accepté et adopté.

Ceci va engendrer la définition d'un ensemble de règles en langage naturel. Ces règles doivent être traduites en une notation adaptée pour leur communication aux auteurs potentiels d'objets d'apprentissage. Ces mêmes règles doivent être également traduites en un langage informatique afin d'assurer leur vérification automatique à chaque fois qu'un objet est proposé par un auteur.

6.1. Règles en langue naturelle

Les règles, comme nous l'avons mentionné, peuvent être le résultat de l'accumulation de directives édictées par plusieurs entités. Par exemple, les deux règles suivantes peuvent être définies à l'échelle d'un établissement universitaire :

(R1) Le nombre de concept traités au niveau du contenu ne doit pas dépasser cinq concepts.

(R2) Le nombre de niveaux d'abstraction ne doit pas dépasser quatre niveaux.

Par la première règle l'établissement peut vouloir s'assurer que le contenu des objets d'apprentissage ne présente pas une surcharge cognitive importante pour les apprenants. En fait, c'est l'un des facteurs qui peuvent justifier un taux important d'abandon.

Alors que par la deuxième règle, l'établissement souhaite éviter que la composition du graphe concret soit très différente de la composition du graphe abstrait ce qui peut engendrer certains problèmes. En effet, l'auteur peut constater que ce qu'il obtient au niveau concret ne correspond pas à son objectif initial. Ceci peut entraîner l'abandon de la conception par réutilisation des objets d'apprentissage et donc des coûts plus élevés en termes d'efforts pour les enseignants et donc pour l'établissement.

A l'échelle des départements d'autres règles plus spécifiques aux disciplines enseignées et à leurs spécificités peuvent être définies. Voici un exemple de trois règles qui peuvent être définies par un département:

(R3) Le nombre d'objets d'apprentissage au niveau du graphe concret ne doit pas dépasser vingt objets.

(R4) Chaque graphe affichable doit inclure des objets ayant les rôles « Introduction », « Exercice » et « Exemple ».

(R5) *Le nombre des graphes affichables ne doit pas dépasser cinq.*

Le ou les responsables d'un module spécifique peuvent également proposer des règles supplémentaires auxquelles les objets d'apprentissage qui traitent ce module doivent être conformes. Les deux règles suivantes peuvent être définies à ce niveau :

(R6) *Le graphe de composition abstrait doit commencer par un objet d'apprentissage simple ayant le rôle « introduction » et doit contenir un objet avec le rôle « Exercice ».*

(R7) *Tous les graphes affichables doivent partager les deux premiers objets d'apprentissage.*

Les règles sont donc définies par un ou par plusieurs intervenants. Elles peuvent être issues d'une approche pédagogique et/ou d'une expertise. Leurs énoncés sont formulés en langage naturel ce qui donne certes une flexibilité mais en contre partie peut engendrer des risques de mauvaise interprétation ou de confusion. Ce risque prend plus d'ampleur lorsque leur nombre devient élevé.

6.2. Règles en langage formel

Pour formuler les règles suivant notre notation il faut les classer suivant leurs catégories. Pour notre cas d'étude nous avons des règles qui couvrent trois catégories parmi les quatre possibles. En fait, cet ensemble de règles ne s'intéresse pas aux aspects relatifs aux métadonnées mais à des aspects structurels, sémantiques et hybrides.

En ce qui concerne l'aspect structurel nous avons quatre règles (R2), (R3), (R5) et (R7). Elles sont notées de la façon suivante :

(R2) : $LO \rightarrow AbstractCG \rightarrow abstractionLevels() \leq 4$

(R3) : $LO \rightarrow ConcreteCG \rightarrow count('LO') \leq 20$

(R5) : $LO \rightarrow DeliveryGraph \rightarrow number() \leq 5$

(R7) : $LO \rightarrow DeliveryGraph \rightarrow similarity('Beginning') \geq 2$

Pour ce qui relève de la sémantique nous avons deux règles (R1) et (R4). Elles sont notées de la façon suivante :

(R1) : $LO \rightarrow Content \rightarrow count('Concept') \leq 5$

(R4) : $LO \rightarrow DeliveryGraph \rightarrow list('Role') \supseteq \{ 'introduction', 'Exemple', 'Exercice' \}$

Finalement, la règle (R6) est une règle hybride qui est notée de la façon suivante :

(R6) : N1 : SLO N2 : ANY ;
 N1 -> Content -> list ('Role') \supseteq { 'Introduction' },
 N2 -> Content -> list ('Role') \supseteq { 'Exercice' } ;

6.3. Notation graphique pour les règles

L'ensemble des règles est décrit sous la forme suivante :

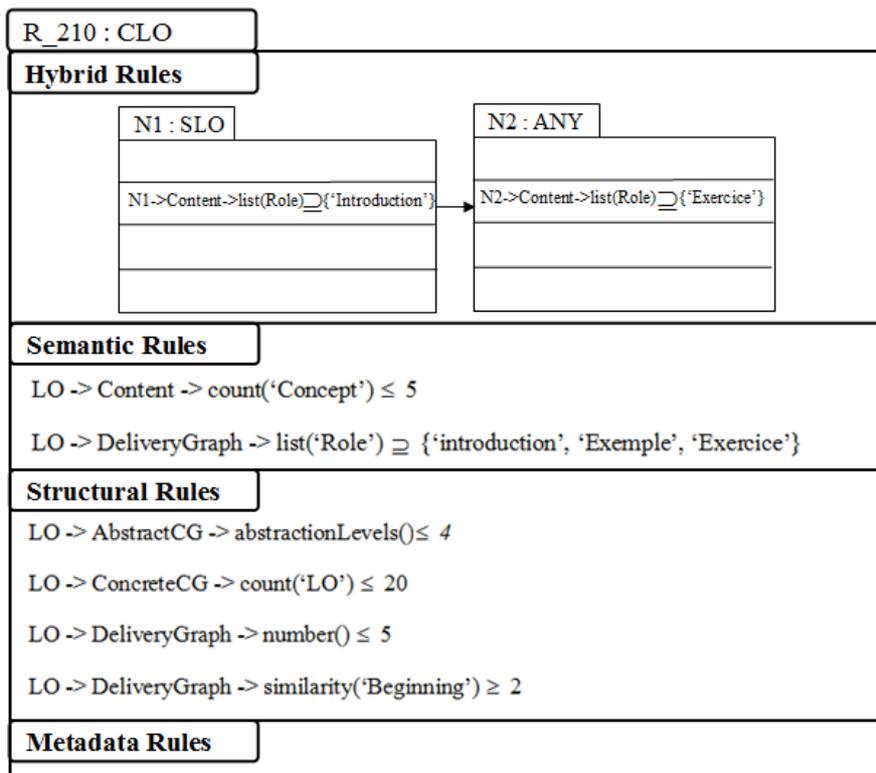


Figure 38 : Notation graphique de règles de conformité

6.4. Vérification automatique des règles

Plusieurs arguments justifient le recours à l'automatisation de la vérification des règles de conformité. D'abord, la vérification manuelle comporte plusieurs risques tels que l'oubli de la vérification de la conformité par rapport à une règle ou l'erreur en considérant ce qui est conforme comme non conforme ou l'inverse. De plus, si le nombre de règles est élevé ou le nombre d'objets d'apprentissage est élevé l'automatisation doit permettre de gagner un temps considérable. Nous pouvons ajouter à ceci le fait que dans un processus de conception itératif et incrémental d'objets d'apprentissage composés, comme c'est le cas dans notre approche d'assistance, faire passer l'objet au niveau de chaque itération au test de la conformité d'une façon non automatisée peut décourager l'auteur à adopter cette approche.

Ainsi, l'automatisation est un élément clé dans notre approche. Pour y parvenir nous avons fait recours à la programmation en F-logique (logique de frames) pour formuler les règles. Ensuite, la vérification automatique se fait par un système tel qu'Ontobroker⁹ dans notre cas. Les détails concernant la F-logique et Ontobroker seront présentés dans le chapitre suivant sur la partie expérimentale.

Voici la façon avec laquelle les différentes règles seront exprimées en F-logique :

(R1) *FORALL LO,R,C,N LO:R1Conforms<- LO[hasContent@(R)->>C] and count(LO,C,N) and lessorequal(N,5).*

(R2) *FORALL LO,G,L,N LO:R2Conforms<- LO[hasGraph->G] and p_g_levels(G,L) and count(G,L,N) and less(N,4).*

(R3) *FORALL LO,G,V,N LO:R3Conforms<- LO[hasConcreteGraph->G] and p_cg_vertices(G,V) and V:AtomicLearningObject and count(G,V,N) and less(N,20).*

(R4) *FORALL LO,G LO:R4Conforms<- LO[hasDeliveryGraph->>G] and p_dg_existsRole(G,"introduction") and p_dg_existsRole(G,"exercise") and p_dg_existsRole(G,"example").*

(R5) *FORALL LO,G,D,N LO:R5Conforms<- LO[hasGraph->G] and p_g_dg(G,D) and count(G,D,N) and less(N,5).*

(R6) *FORALL LO,R,C1,C2,V LO:R6Conforms<- LO[hasContent@("exercise")->>C1] and LO[hasRoot ->V] and V[hasContent@("introduction") ->> C2].*

(R7) *FORALL LO,R,G,N LO:R7Conforms<- LO[hasDGrah->G] and p_dg_beginSimilarity(G,N) and lessorequal(2,N).*

Il faut noter que nous avons défini des prédicats spécialisés. Nous avons donné des noms à ces prédicats qui commencent par « p_ ». Par exemple, p_g_levels(X,Y) est un prédicat qui indique que le graphe de composition X à Y niveaux d'abstraction. Sans la définition de ces prédicats la formulation de la conformité aux règles sera une tâche très difficile avec les prédicats de base de la F-logique.

⁹ www.ontoprise.de/en/home/products/ontobroker/

7. Conclusion

Tout au long de ce chapitre nous avons traité les différents éléments relatifs à la phase d'analyse de la conformité des objets d'apprentissage composés par rapport à un patron de conception (ensemble de règles). Nous avons souligné l'importance de cette phase dans un processus de conception d'objets d'apprentissage. Nous avons également présenté une étude des différents aspects qui peuvent être contrôlés par des règles de conformité avec des exemples à l'appui.

Par la suite, nous avons introduit le langage formel de description de règles via sa grammaire et en reprenant les différents exemples énoncés en langage naturel en les décrivant en langage formel. Nous avons également introduit une notation graphique plus adaptée pour la communication des règles de conformité aux auteurs. Les différents éléments de structuration et de description de patrons de conception ont été introduits. L'expressivité du langage formel et de la notation graphique a été discutée. Enfin, nous avons terminé le chapitre avec un exemple complet qui dévoile les différents éléments relatifs à la vérification de la conformité.

Dans le chapitre suivant nous allons, entre autres, présenter et discuter les différents éléments relatifs à l'implémentation et à l'expérimentation du module de la vérification de la conformité et plus largement de notre approche.

Chapitre VII :

MODELISATION ET PROTOTYPE

1. Introduction

Dans ce chapitre nous décrivons l'implémentation que nous avons faite des différentes propositions faites dans ce travail. L'intégration logicielle est faite dans le logiciel COLORS que nous avons développé et qui est un environnement auteur d'objets d'apprentissage SCORM et IMS travaillant sur un ensemble de banques d'objets distantes. Après avoir décrit COLORS, nous décrivons son extension aux objets SIMBAD. La plus grande partie des analyses présentées dans ce travail est implémentée via la logique des frames. Nous allons présenter le langage F-logique et la plateforme OntoBroker que nous avons utilisés pour la mise en œuvre de notre approche. Puis, nous allons présenter notre modélisation des objets SIMBAD conformément à ce formalisme. Enfin nous décrivons l'implémentation des différentes analyses en se focalisant sur les règles de conformité.

2. Système COLORS

2.1. Contexte

De nos jours le concept d'objet d'apprentissage est considéré comme l'un des concepts les plus prometteurs pour l'avenir de la formation en ligne. L'un des principaux apports de ce concept est le fait de promouvoir la réutilisation des contenus éducatifs. En fait, la mise en place des formations en ligne est marquée par le coût élevé de la réalisation d'un contenu sous une forme numérisée. En plus, force est de constater une forte ressemblance entre les contenus des formations similaires. En conséquence, la réutilisation de l'existant est une piste prometteuse pour tout organisme souhaitant mettre en place une formation en ligne avec des facteurs de succès et de rentabilité notamment un retour sur investissement et une capitalisation de l'existant.

Cependant, il est nécessaire d'allier aux objets d'apprentissage les outils informatiques nécessaires pour une exploitation réelle, efficace et rentable. Parmi les outils concernés nous

pouvons citer les outils d'édition (RELAOD, eXe, etc.), les LCMS¹⁰ (MOODLE, INES, CLAROLINE, etc.) et les banques d'objets d'apprentissage (eRIB, PLANET, ARIADNE, DOOR, etc). C'est aux banques d'objets d'apprentissage que nous nous somme intéressés.

Les banques d'objets d'apprentissage représentent une composante fondamentale dans l'infrastructure logicielle de toute institution adoptant la formation en ligne. Leur rôle principal est de rassembler en un endroit unique, au moins d'un point de vue logique et pas forcément physique, les objets d'apprentissage. Il est alors plus facile de trouver, de gérer et d'utiliser les objets d'apprentissage à partir d'un point d'accès unique. Ce point d'accès, qui est la banque d'objets d'apprentissage, permet d'éviter une anarchie inévitable avec des contenus redondants, invisibles ou fictifs.

Toutefois, ce n'est pas le concept intrinsèque de banque d'objets d'apprentissage qui a suscité notre attention, d'un point de vue recherche, mais plutôt le questionnement sur son utilisation dans un cadre de coopération inter-organisationnelle et à grande échelle. Encore une fois, cette perspective à été depuis quelques années étudiée. En effet, le concept même d'objets d'apprentissage se manifeste comme catalyseur pour la réutilisation et l'interopérabilité. D'autre part, ces deux éléments représentent les ingrédients de base pour toute coopération entre les institutions œuvrant dans la sphère de la formation en ligne. Ainsi, le concept d'objets d'apprentissage doit être un élément clé pour favoriser une synergie entre ces institutions. Ceci explique la mise en place de projets de création de réseaux de banques d'objets d'apprentissage de part et d'autre du monde.

Cependant, les réseaux qui ont été mis en place souffrent d'une contextualisation forte au point qu'il est difficile de les réutiliser ailleurs. C'est-à-dire ils n'ont pas été faits généralement dans une optique de transfert ou d'adaptation à d'autres bénéficiaires. Autrement dit, la seule possibilité qui peut être négociée c'est celle d'intégrer ces réseaux déjà développés sans avoir la possibilité de créer ses propres réseaux. Notre réponse à cette situation c'est le projet intitulé COLORS pour **CO**operative **L**earning **O**bject **R**epository **S**ystem.

En fait, l'objectif est de permettre à plusieurs établissements universitaires de partager à la fois leurs objets d'apprentissage en interne tout en ayant la possibilité d'intégrer un réseau de

¹⁰ LCMS : Learning Content Management System

banques d'objets d'apprentissage dans un cadre de partenariat et de coopération avec des partenaires externes. Il s'agit de permettre la réutilisation des objets d'apprentissage localement et éventuellement à grande échelle. C'est ainsi que nous avons décidé de mettre en place une infrastructure logicielle de partage d'objets d'apprentissage à une échelle multi-organisationnelle supportant l'esprit de coopération et d'ouverture. Afin d'atteindre cet objectif COLORS doit être évolutif, interopérable avec l'existant, flexible et portable. Présentons tout d'abord COLORS avant de revenir sur la façon avec laquelle nous avons répondu à ces objectifs.

2.2. Présentation

COLORS est une banque d'objets d'apprentissage Open source (License GNU GPL : GNU GENERAL PUBLIC LICENSE Version 2, June 1991) développée en langage PHP à base du code source de DOOR¹¹ qui est également une banque d'objets Open Source. Par rapport à son prédécesseur, COLORS garde le fait qu'il y a trois acteurs : l'administrateur, l'auteur et le simple utilisateur. Côté fonctionnalités, COLORS a hérité les fonctions de gestion des utilisateurs, des objets d'apprentissage et les fonctions de recherche et d'exploration du contenu de la banque.

Toutefois, COLORS se distingue par une prise en compte native de la langue arabe. Elle est interopérable avec MOODLE¹² grâce à un plugin spécifique. En plus, elle est complètement conforme aux spécifications IMS Content Package 1.1.3, SCORM 1.2 et SCORM 2004. Cependant, l'originalité la plus marquante de COLORS est qu'elle offre la possibilité à une banque COLORS de s'intégrer à un ou plusieurs réseaux de banques d'objets d'apprentissage COLORS. De plus amples détails sont donnés dans l'annexe C.

¹¹ <http://door.sourceforge.net>

¹² <http://moodle.org>

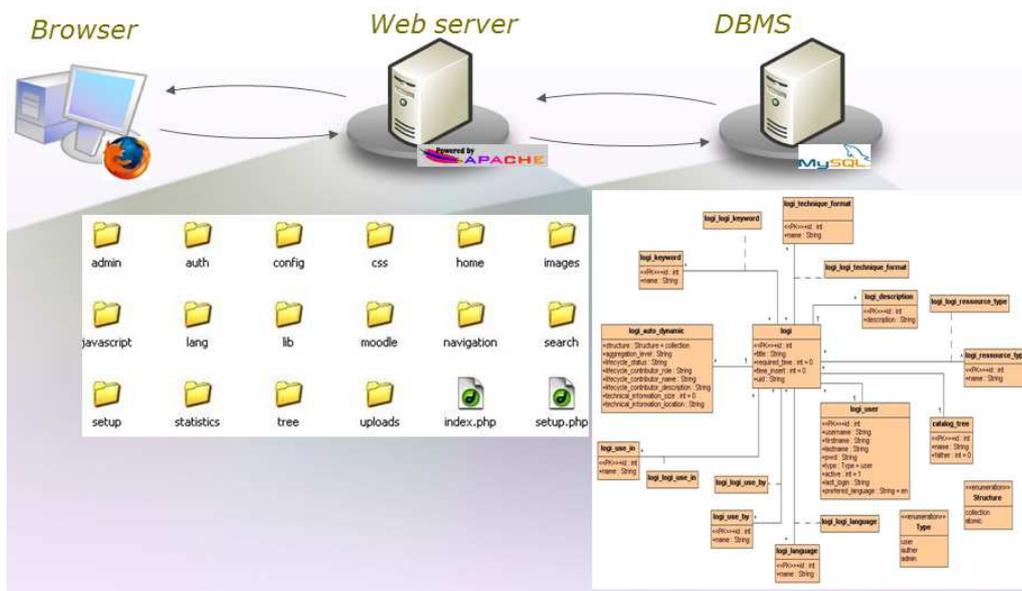


Figure 39 : Architecture logicielle du système COLORS

2.3. Coordinateur de banques COLORS

En fait, pour mettre en place un réseau de partage d'objets d'apprentissage il faut plusieurs instances de COLORS et un coordinateur pour le réseau de banques. Ce coordinateur, que nous avons développé, est une application Web open source simple à administrer. Elle joue le rôle d'un médiateur entre les banques d'un même réseau. Elle permet de gérer la procédure d'inscription permettant à des institutions de mutualiser leurs objets d'apprentissage. C'est au niveau de cette composante, qui forme le cœur d'un réseau de banques d'objets d'apprentissage, que nous avons implémenté une approche de fédération de ressources et que nous avons fixé les règles de propagation des requêtes et d'agrégation des résultats.

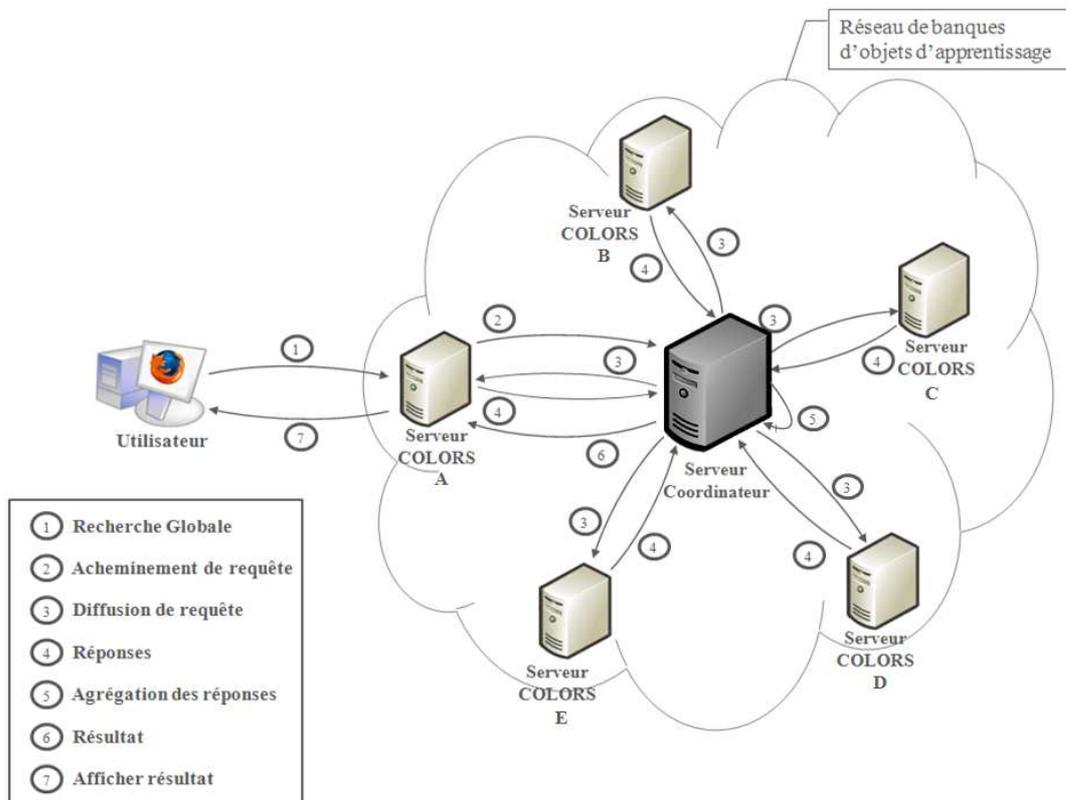


Figure 40 : Gestion des requêtes au niveau d'un réseau de banques COLORS

Ce coordinateur supporte deux types d'acteurs. Le premier c'est l'administrateur qui a comme rôle principale de gérer (activer, bloquer ou supprimer) les instances de COLORS qui forment le réseau de banques, de consulter des statistiques, de modifier les conditions d'utilisation, de reconfigurer le système, et de gérer les paramètres de son compte. Le deuxième acteur c'est tout utilisateur ou utilisateur potentiel. Il pourra consulter les conditions d'utilisation fixées par la communauté qui détient le réseau de banques. Il peut également consulter la liste des institutions qui forment le réseau.



Figure 41 : Interface de gestion des banques d'un réseau COLORS par le coordinateur

2.4. Assistance aux auteurs via COLORS

Les banques d'objets d'apprentissage offrent aux auteurs des fonctionnalités classiques liées au dépôt des objets d'apprentissage. Elles consistent le plus souvent à l'importation de l'objet et à la gestion des métadonnées. Avec COLORS nous souhaitons offrir aux auteurs des outils supplémentaires issus de notre approche d'assistance.

En fait, dans la version actuelle du système COLORS nous avons introduit un module simple d'analyse qui porte uniquement sur les métadonnées éducatives. L'objectif étant d'étudier dans un premier temps la façon avec laquelle l'analyse pour la compréhension des objets d'apprentissage peut être intégrée et présentée comme un service supplémentaire aux auteurs. Dans ce contexte nous avons inclus un module permettant de comparer les propriétés de l'objet par rapport à ceux des dix objets les plus consultés dans la banque COLORS [Farhat et Jemni, 2009].

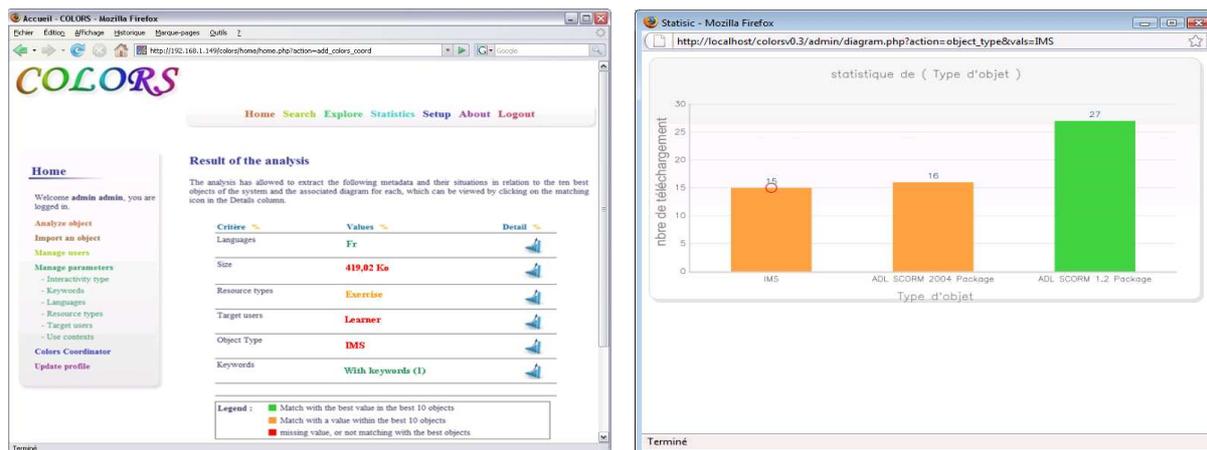


Figure 42 : Interfaces COLORS relatives à l'analyse des propriétés des objets d'apprentissage

Comme le montre la capture d'écran à gauche, les résultats de l'analyse sont affichés par critère et en trois couleurs. La couleur verte indique que pour ce critère l'objet a une valeur qui correspond à celle de l'objet le plus consulté dans la banque. Si la couleur est orangée, la valeur correspond à l'une des valeurs des dix objets les plus consultés. La couleur rouge indique que la valeur ne correspond à aucune valeur des dix objets les plus téléchargés. L'auteur peut consulter les détails en cliquant sur le lien correspondant. Par exemple, il peut voir les détails concernant le type de l'objet d'apprentissage. Ainsi, l'auteur peut constater que la plupart des utilisateurs de cette banque d'objets d'apprentissage préfèrent des objets au format SCORM 1.2 plutôt que IMS ou SCORM 2004 comme le montre la capture d'écran à droite. Ceci peut être expliqué par le fait que les utilisateurs utilisent plutôt des systèmes de gestion de l'apprentissage compatible avec les objets SCORM 1.2. Donc, pour favoriser l'usage de ses objets d'apprentissage l'auteur a intérêt à concevoir des objets dans ce format.

2.5. Extension de COLORS aux objets SIMBAD

Notre approche d'assistance prend réellement intérêt dès que la structure et la sémantique des objets d'apprentissage atteint un certain niveau de complexité. C'est le cas des objets composés SIMBAD. Ainsi, nous avons proposé une extension pour le système COLORS qui prévoit la gestion des objets composés SIMBAD. Pour cela, nous avons remplacé le module d'analyse en PHP de COLORS par un ensemble de modules construits avec la logique des frames et le logiciel Ontobroker. Ces modules supposent que les objets d'apprentissage en entrées soient conformes au format SIMBAD (exprimé en F-logique). Nous avons donc ajouté un module de traduction SCORM et IMS vers SIMBAD, ce qui permet d'analyser un grand nombre d'objets d'apprentissage, même si cette analyse est moins fouillée que celle des objets SIMBAD.

Voici l'architecture que nous proposons :

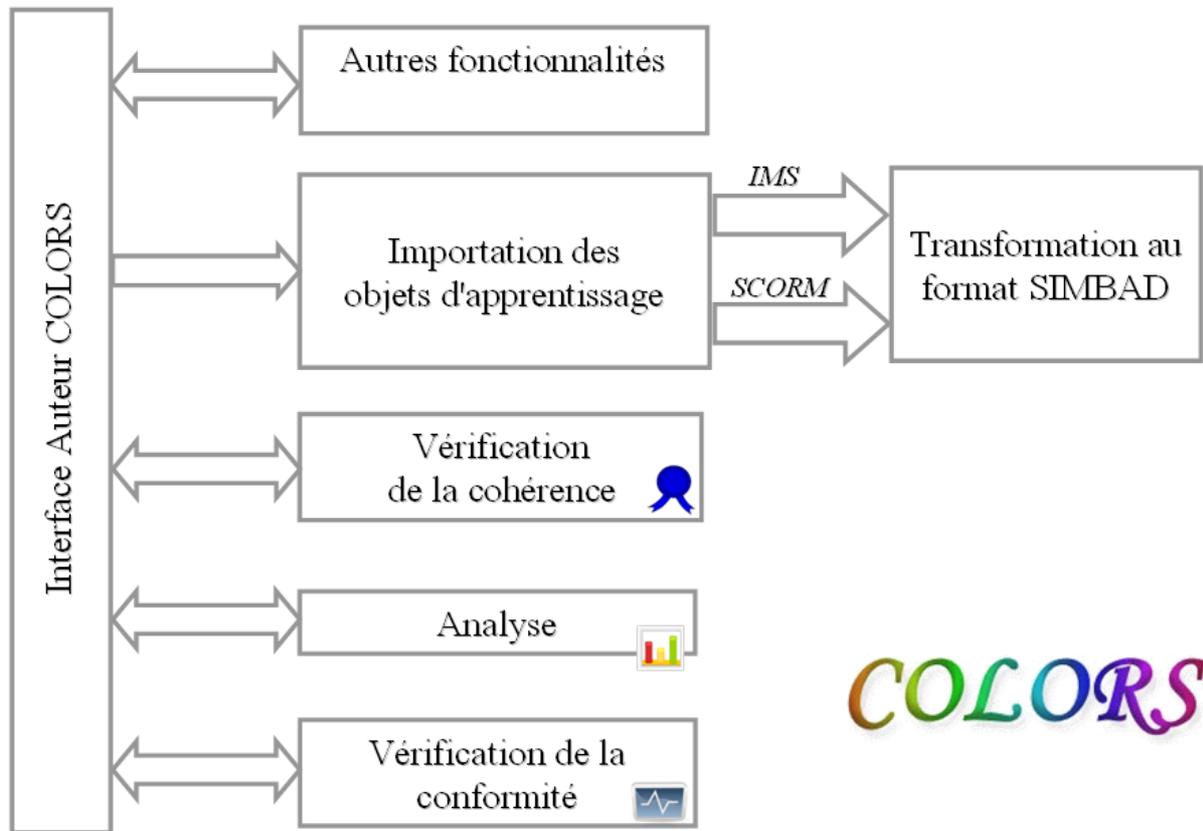


Figure 43 : Architecture de l'extension de COLORS

L'implémentation de notre approche est basée sur l'utilisation de la programmation logique. Ainsi, nous allons introduire le langage F-logique et le moteur d'inférence OntoBroker.

3. Logique des frames

3.1. Introduction

D'après [Fensel 2000] si la logique des prédicats offre comme primitive de modélisation la notion de prédicat, d'autres approches proposent une modélisation basée sur d'autres primitives. En effet, il existe des approches basées sur les « Frames » (classes) et les attributs comme primitives de modélisation. La logique des Frames, ou F-logique, fait partie de ces approches.

L'utilisation de la F-logique est motivée tout d'abord par sa capacité à répondre à nos besoins en termes d'expression et d'inférence. En effet, elle est capable d'exprimer les différentes composantes et propriétés d'un objet d'apprentissage. En plus, elle est capable d'inférer

différentes informations permettant la mise en place de notre approche d'assistance aux auteurs d'objets d'apprentissage composés.

Outre les caractéristiques intrinsèques de la F-logique, son utilisation dans le cadre de cette thèse a été favorisée par le fait qu'il s'agit d'un formalisme déjà utilisé au sein du projet SIMBAD. Ceci nous permet de profiter de l'expertise et du savoir faire existant. D'autre part, nous disposons d'une licence du système OntoBroker qui est un moteur d'inférences efficace pour la F-logique.

3.2. Présentation de la F-logique

D'après [Fensel 2000] la F-logique permet la spécification de bases de données orientées objet, de systèmes à base de classes et de faire de la programmation logique. Son principal apport, par rapport à des approches concurrentes c'est l'intégration de constructeurs conceptuels de modélisation : les classes, les attributs, les restrictions de domaine et de portée, l'héritage et les axiomes. Les axiomes logiques permettent d'exprimer, dans le cas de modélisation d'ontologies, les relations entre les éléments de cette ontologie et leurs instances. Toutes ces primitives de modélisation sont introduites dans un cadre logique cohérent.

L'alphabet de la F-logique consiste en un ensemble de symboles de fonctions et un ensemble de variables. Un terme est un terme usuel de la logique de premier ordre composé de symboles de fonctions et de variables.

3.3. Langage F-logique

Le langage associé à la F-logique est constitué d'un ensemble de formules construites à partir des symboles de l'alphabet. Comme dans plusieurs autres logiques, les formules peuvent être construites à partir de formules simples par l'utilisation des connecteurs usuels (non / et /ou) et des quantificateurs universel et existentiel [Ontoprise 2006b].

Avec le langage F-logique il est possible de définir trois types d'expressions : les faits, les règles et les requêtes.

- Exemple de fait : LO_001 est un objet d'apprentissage

LO_001 : LearningObject.

L'élément central à considérer est l'objet d'apprentissage qui se décline en deux variantes : les objets simples et les objets composés. Il est décrit par un contenu, des pré-requis et une fonction d'acquisition. Ces éléments représentent les métadonnées sémantiques à qui sont associées les métadonnées LOM constituées par des entrées organisées en catégories.

A chaque objet d'apprentissage composé est associé un graphe de composition abstrait. Celui-ci comporte des arcs et des sommets. Ces derniers se déclinent en opérateurs et sommets avec leurs différentes variantes comme décrit par le diagramme de classe ci-dessous.

4.2. Objet d'apprentissage

En F-logique on utilise la notion de signature pour définir les méthodes applicables aux instances de classes. Ainsi, la signature déclare la méthode et spécifie les restrictions concernant les paramètres et les résultats.

Pour distinguer les signatures des données, les flèches utilisées pour les signatures sont doubles (=) et leurs sommets indiquent si le résultat est une valeur (=>) ou un ensemble de valeurs (=>>). Par exemple, pour spécifier que la méthode « hasContent » est applicable à un objet d'apprentissage et qu'elle a pour paramètre un rôle et comme résultat des concepts, nous devons utiliser la signature suivante :

```
LearningObject [hasContent@(Role) ==>> Concept].
```

Une instance d'un objet d'apprentissage prend la forme suivante :

```
PrimitiveLearningObject :: LearningObject
LO_001 : PrimitiveLearningObject.
LO_001 [hasContent@(introduction) -> database].
```

Ici nous avons spécifié que LO_001 est une instance de la classe « PrimitiveLearningObject ». Cette classe est une sous-classe de la classe « LearningObject » et donc hérite de ses méthodes. Ainsi, nous avons pu spécifier que cet objet a comme contenu le concept « database » avec le rôle « introduction ». Voici les signatures relatives aux objets d'apprentissage :

```
LearningObject
[
    hasContent@(Role) ==>> Concept;
    hasPrerequisite@(Role,KnowledgeLevel) ==>> Concept;
    hasAcquisitionFunction@(Role,KnowledgeLevel) ==>> Concept;
```

```

        otherCharacteristics =>> MetadataEntry

    ].

PrimitiveLearningObject :: LearningObject.
PrimitiveLearningObject
    [
        hasURL=>String
    ].

ComposedLearningObject :: LearningObject.
ComposedLearningObject
    [
        hasGraph => Graph;
    ].

```

Par la suite il faut définir les instances des objets d'apprentissage conformément à la signature.

L'instanciation d'un objet d'apprentissage simple prend ainsi la forme suivante :

```

lo_001 : PrimitiveLearningObject.
lo_001
[
    hasContent@(introduction) -> database;
    hasPrerequisite@(introduction, medium) -> informatics ;
    hasAcquisitionFunction@(introduction, beginner)->> database ;
    hasURL -> "http://localhost/lo/lo_001.html" ;
    otherCharacteristics ->> {lo_001_Category, lo_001_Entry, lo_001_Title,
lo_001_Title, lo_001_Language}
].

```

L'instanciation d'un objet d'apprentissage composé prend la forme suivante :

```

lo_901 : ComposedLearningObject.
lo_901
[

```

```

hasPrerequisite@() -> ;

hasContent@(introduction) -> database;

hasAcquisitionFunction@(introduction, beginner) -> database;

hasGraph -> graph_901;

otherCharacteristics ->> {lo_901_Catalog, lo_901_Entry, lo_901_Title,
lo_901_Language}

].

```

4.3. Graphe de composition

Chaque objet d'apprentissage composé est décrit par un graphe de composition abstrait (« *Graph* »). Celui-ci est décrit par l'ensemble des arcs (« *Transition* ») qui le décrit. De plus, chaque graphe de composition se distingue par un sommet racine et des sommets feuilles qui sont des objets d'apprentissage.

```

Graph :: ContentNode.

Graph
[
    hasRoot => LearningObject;
    hasEnd =>> LearningObject;
    hasTransition =>> Transition;
].

```

Chaque arc à une origine et une destination qui sont des sommets (« *Vertex* »). Chaque sommet est soit un opérateur (« *Operator* ») ou un objet d'apprentissage (« *LearningObject* »). Voici la signature que nous avons définie pour un graphe en F-logique :

```

Transition :: GraphObject.

Transition
[
    toVertex => Vertex;
    fromVertex => Vertex;
].

```

```
Operator :: Vertex .
LearningObject :: Vertex .
```

D'autres éléments relatifs à SIMBAD ont été décrits en F-logique tels que le modèle de domaine ou les métadonnées LOM. Nous allons les présenter au niveau de l'annexe A.

4.4. Exemple illustratif

Supposant que l'objet d'apprentissage composé LO_901, introduit dans la section 3.1 de ce chapitre, a le graphe de composition abstrait suivant :

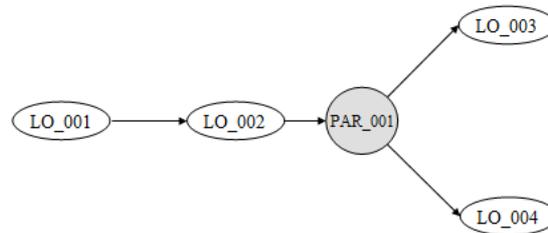


Figure 45 : Exemple d'un graphe de composition abstrait

Voici la description du graphe ci-dessus :

```
graph_001 : Graph.
graph_001 [
  hasRoot -> LO_001;
  hasEnd ->> {LO_003, LO_004};
  hasTransition ->> {transition_001, transition_002, transition_003,
transition_004}
].
transition_001 : Transition.
transition_001 [
  fromVertex -> LO_001;
  toVertex -> LO_002
].
transition_002 : Transition.
transition_002 [
  fromVertex -> LO_002;
  toVertex -> op_par_001
].
```

```

transition_003 : Transition.
transition_003 [
    fromVertex -> op_par_001;
    toVertex -> LO_003
].

transition_004 : Transition.
transition_004 [
    fromVertex -> op_par_001;
    toVertex -> LO_004
].

```

5. Analyse des objets SIMBAD

5.1. Architecture

Afin, de mettre en œuvre les différents mécanismes d'assistance aux auteurs des objets d'apprentissage composés nous avons mis au point un ensemble de modules développés en langage Java et basés sur l'utilisation de la programmation logique via le langage F-logique. Le schéma suivant décrit l'architecture suivie :

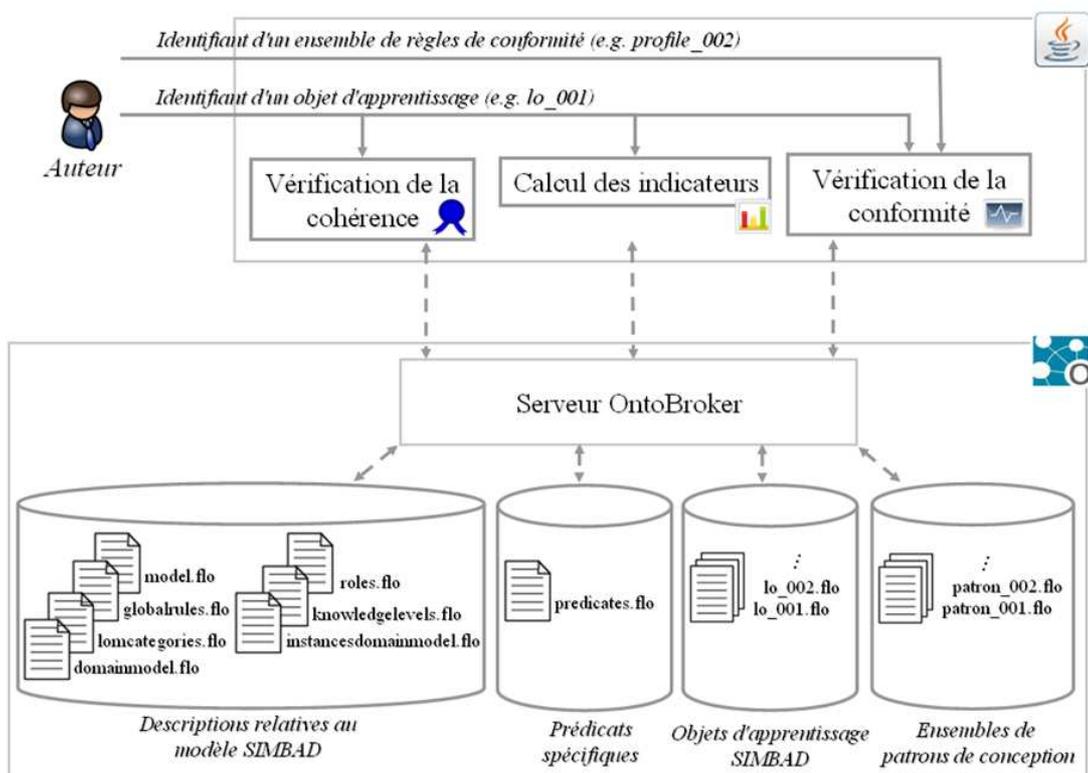


Figure 46 : Architecture expérimentale

Il s'agit d'une architecture client/serveur avec une partie client développée en langage Java et en une partie serveur formée d'OntoBroker le moteur d'inférence associé au langage F-logique.

Le langage Java est utilisé principalement pour simplifier la manipulation pour l'utilisateur (il n'a pas à connaître le langage F-logique). Toutefois, les algorithmes que nous utilisons pour la vérification de la cohérence, l'analyse multicritères et sémantique et la vérification de la conformité nécessite un langage qui supporte les expressions conditionnelles et itératives. Ainsi, les indicateurs et les données nécessaires pour l'exécution des traitements sont demandées à OntoBroker via des requêtes en F-logique.

Le serveur OntoBroker est lancé avec une configuration basée sur un ensemble de fichiers comportant les éléments nécessaire pour pouvoir assister l'auteur dans la composition d'un objet d'apprentissage composé.

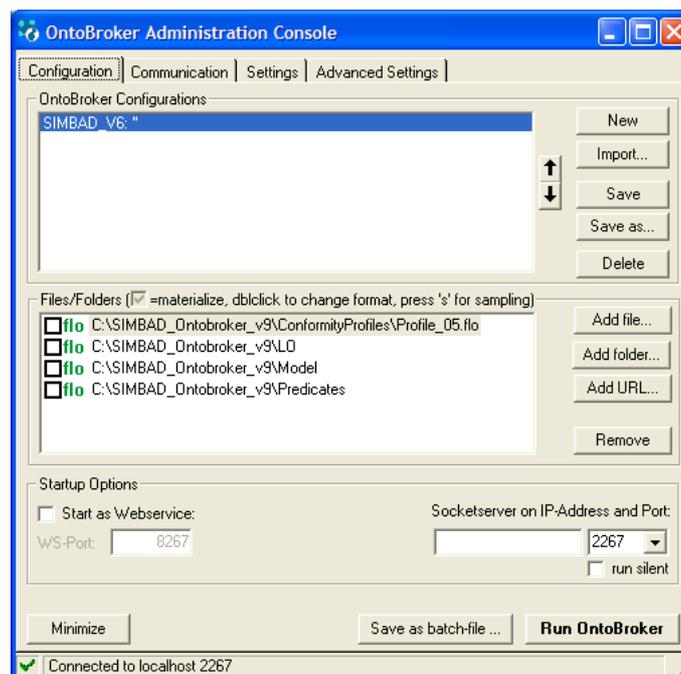


Figure 47 : Console d'administration OntoBroker 5.0

Les fichiers sont organisés en quatre répertoires :

- *Model* : Il contient sept fichiers qui décrivent les éléments du modèle SIMBAD
 - *Model.flo* : contient les signatures nécessaires pour définir des instances d'objets d'apprentissage, de graphes de composition et de métadonnées.
 - *GlobalRules.flo* : contient les règles qui permettent de classer les objets d'apprentissage en conformes ou pas conformes aux règles de conformité.

- *LOMCategories.flo* : contient la signature et les instances des catégories et des entrées de métadonnées LOM.
 - *DomainModel.flo* : contient les signatures relatives au modèle de domaine en termes de concepts et de relations inter-concepts.
 - *InstancesDomainModel.flo* : contient les instances relatives au modèle de domaine.
 - *Roles.flo* : contient la définition des rôles éducatifs.
 - *KnowledgeLevels.flo* : contient les niveaux de connaissances à considérer pour la définition des pré-requis et des fonctions d’acquisition.
- *LO* : contient les fichiers qui décrivent les différents objets d’apprentissage.
 - *Predicates* : contient les différents prédicats de haut niveau qui ont été défini pour simplifier les manipulations spécifiques aux objets d’apprentissage.
 - *ConformityProfiles* : contient les différents ensembles de règles de conformité.

5.2. Implémentation

Nous avons implémenté la partie client de l’architecture en langage Java afin d’expérimenter les différents algorithmes proposés et pour tester les mécanismes d’assistance aux auteurs que nous avons proposés.

Le diagramme de classe proposé dans cette section montre les différentes classes que nous avons implémentées et les relations établies entre elles. Pour des raisons de lisibilité du diagramme nous avons montré uniquement les attributs et les méthodes publiques.

La classe « *ComposedLearningObject* » est la classe centrale dans notre implémentation. Elle offre les différents services d’assistance via ses méthodes. A cette classe est associé la classe « *CompositionGraph* » qui offre les services relatifs aux aspects structurels des graphes de composition. Deux classes héritent d’elle : « *AbstractCompositionGraph* » et « *ConcreteCompositionGraph* ». La première offre en plus des services spécifiques au graphe de composition abstrait. La deuxième est relative au graphe de composition concret généré par la classe « *ConcreteGraphGenerator* ». La classe nommée « *DeliveryGraphGenerator* » permet de générer les graphes affichables à partir d’un graphe de composition concret.

A l'objet d'apprentissage composé est associé les classes « LOM » et « SemanticMetadata » qui héritent de la classe « Metadata ». Chacune (des deux classes) offre des services relatifs à chaque famille de métadonnées via leurs méthodes respectives.

La classe « SemanticSpace » offre des services relatifs à l'espace sémantique de l'objet d'apprentissage composé. Elle fait appel aux services relatifs aux métadonnées sémantiques et au graphe de composition abstrait.

La classe « OntoBrokerClient » permet aux différents objets de notre application de communiquer avec le serveur OntoBroker. Cette classe présente plusieurs services dont la possibilité d'envoyer des requêtes ainsi que d'ajouter ou de supprimer des faits de la base de connaissances OntoBroker.

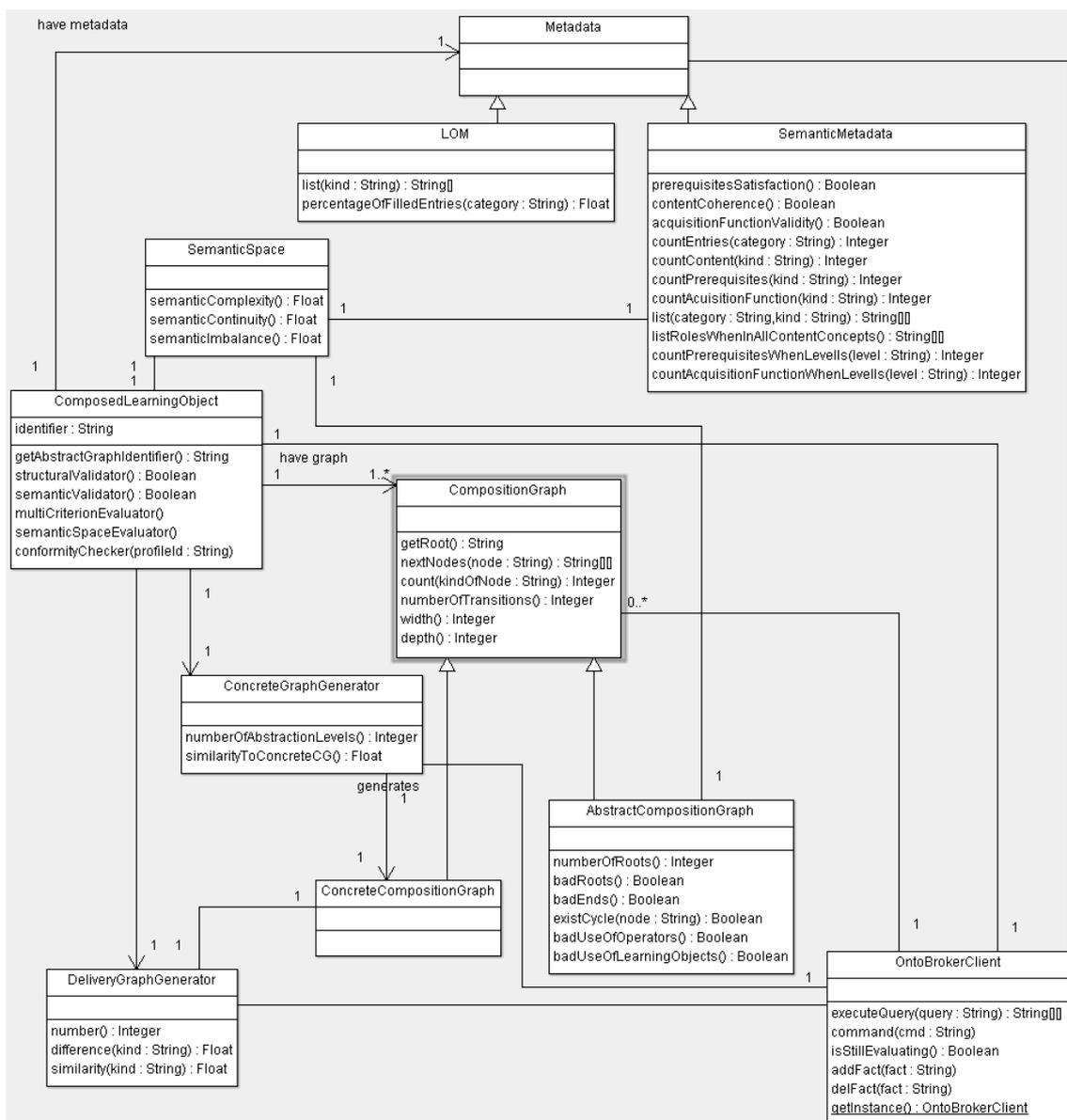


Figure 48 : Diagramme de classe du prototype implémenté

6. Vérification de la cohérence

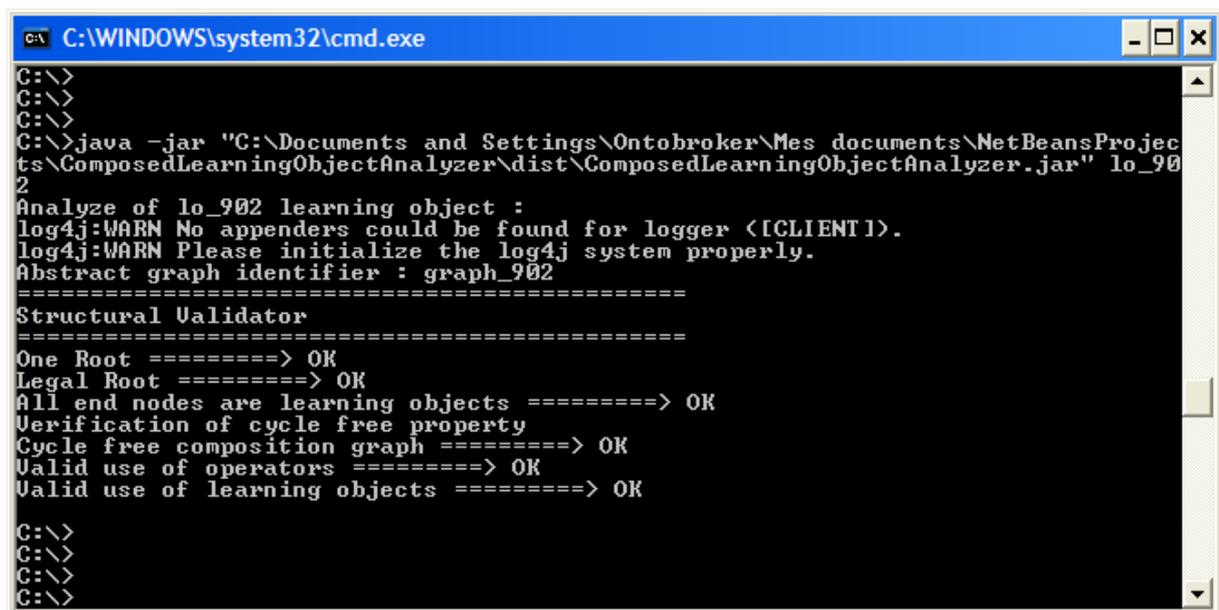
6.1. Expérimentation

Pour expérimenter le module de vérification de la cohérence (sémantique et structurelle) nous avons défini un ensemble d'objets d'apprentissage chacun avec une anomalie spécifique parmi les anomalies structurelles et sémantiques possibles. Nous avons soumis ces différents objets à notre prototype afin de vérifier la capacité des algorithmes que nous avons défini à détecter les incohérences.

6.2. Test et résultats

Concernant les anomalies structurelles nous avons soumis les objets suivants à la vérification automatique de la cohérence structurelle :

LO_902 : Objet d'apprentissage sans anomalies.



```
C:\WINDOWS\system32\cmd.exe
C:\>
C:\>
C:\>
C:\>java -jar "C:\Documents and Settings\Ontobroker\Mes documents\NetBeansProjects\ComposedLearningObjectAnalyzer\dist\ComposedLearningObjectAnalyzer.jar" lo_902
Analyze of lo_902 learning object :
log4j:WARN No appenders could be found for logger [CLIENT].
log4j:WARN Please initialize the log4j system properly.
Abstract graph identifier : graph_902
=====
Structural Validator
=====
One Root =====> OK
Legal Root =====> OK
All end nodes are learning objects =====> OK
Verification of cycle free property
Cycle free composition graph =====> OK
Valid use of operators =====> OK
Valid use of learning objects =====> OK
C:\>
C:\>
C:\>
C:\>
```

Figure 49 : Vérification de la conformité structurelle de l'objet LO_902

LO_1001 : Il comporte deux composantes connexes.

LO_1002 : Il comporte un sommet de type objet d'apprentissage qui est à l'origine de deux arcs.

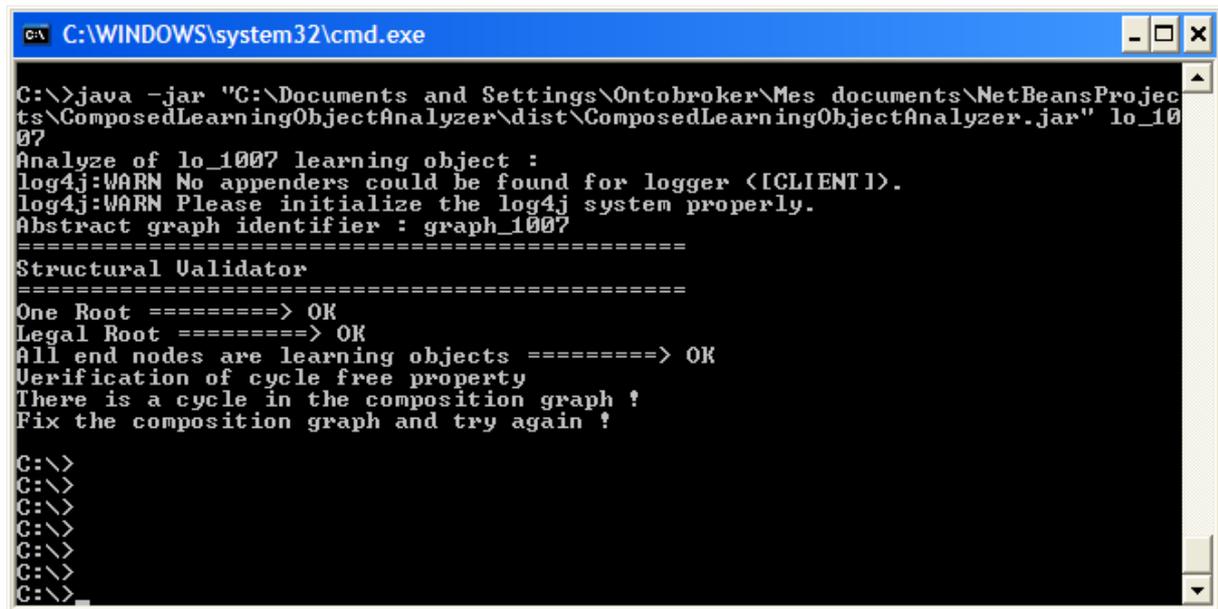
LO_1003 : Il comporte un sommet feuille dans son graphe de composition abstrait qui est de type opérateur.

LO_1004 : Il comporte plusieurs sommets racines.

LO_1005 : Il comporte un sommet racine qui est de type opérateur.

LO_1006 : Il comporte un sommet de type opérateur qui est à l'origine d'un seul arc.

LO_1007 : Il comporte un cycle au sein du graphe de composition abstrait.



```
C:\WINDOWS\system32\cmd.exe
C:\>java -jar "C:\Documents and Settings\Ontobroker\Mes documents\NetBeansProjects\ComposedLearningObjectAnalyzer\dist\ComposedLearningObjectAnalyzer.jar" lo_1007
Analyze of lo_1007 learning object :
log4j:WARN No appenders could be found for logger [CLIENT].
log4j:WARN Please initialize the log4j system properly.
Abstract graph identifier : graph_1007
=====
Structural Validator
=====
One Root =====> OK
Legal Root =====> OK
All end nodes are learning objects =====> OK
Verification of cycle free property
There is a cycle in the composition graph !
Fix the composition graph and try again !

C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
```

Figure 50 : Vérification de la conformité structurelle de l'objet LO_1007

Les différentes simulations ont montré que les algorithmes que nous avons définis permettent de détecter les différentes anomalies structurelles qui peuvent être présentes dans un graphe de composition abstrait.

Pour les aspects sémantiques nous avons préparé deux objets d'apprentissage avec des anomalies sémantiques :

LO_1011 : Un objet d'apprentissage au sein du graphe de composition abstrait comporte un pré-requis non satisfait ni par les objets qui le précèdent ni par les pré-requis de l'objet composé.

LO_1012 : Le graphe de composition abstrait comporte un objet d'apprentissage dont le contenu comporte un concept qui n'est pas parmi les concepts du contenu de l'objet composé.

Ces deux anomalies sémantiques ont été également détectées par l'analyseur. Ce qui permet de valider les algorithmes qui ont été défini pour assurer cet objectif.

7. Assistance à la compréhension

7.1. Analyse générale

Afin de vérifier que la phase d'assistance à la compréhension permet d'avoir des indicateurs significatifs nous avons préparé un ensemble d'objets d'apprentissage composés ayant chacun des caractéristiques précises par rapport aux aspects suivant :

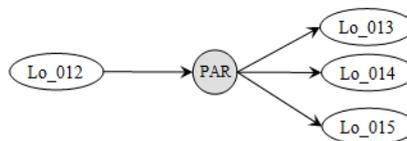
- Cohérence structurelle
- Capacité à être adapté
- Capacité à être réutilisé
- Consistance didactique

Ensuite, nous avons soumis ces objets d'apprentissage composés au processus d'analyse automatique. Nous avons fixé les coefficients utilisés dans les formules à la valeur un et les mesures sont ramenées à une échelle de zéro à dix.

7.2. Tests et résultats

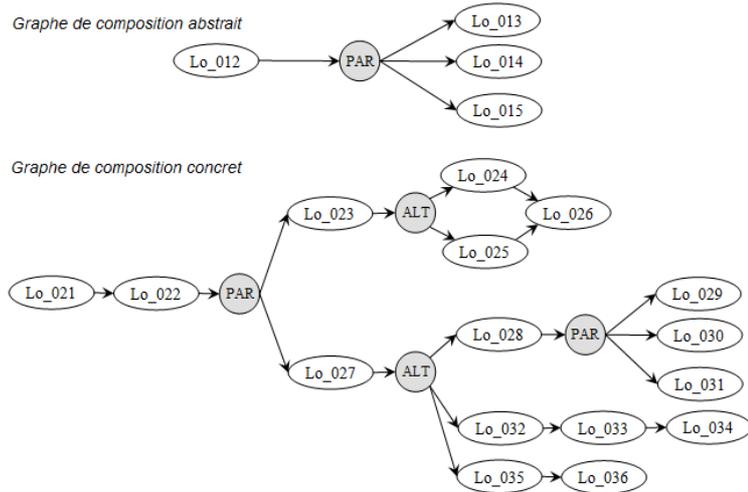
L'objectif des tests que nous avons mené sur l'aspect analyse générale consiste à vérifier que les résultats retournés correspondent bien à la vraie nature des objets d'apprentissage. Afin, d'effectuer cette évaluation nous avons préparé des objets d'apprentissage composés avec des caractéristiques particulières afin de vérifier si les valeurs estimatives calculées des indicateurs les caractérisent correctement. Voici la liste des objets expérimentés et leurs caractéristiques :

LO_920 : Concordance structurelle élevée.



Cet objet est formé d'objets d'apprentissage simples. Ainsi, les graphes de composition abstrait et concret sont identiques.

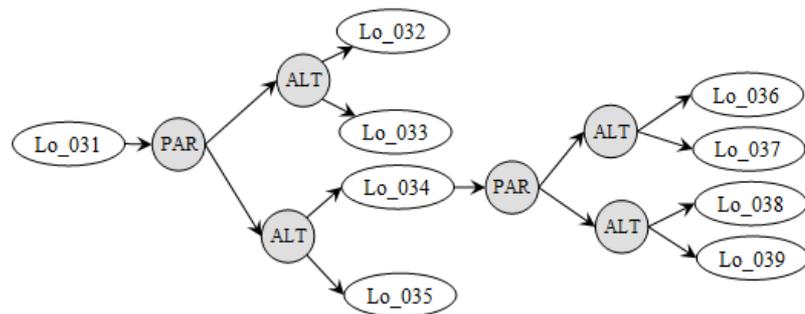
LO_921 : Concordance structurelle faible.



L'objet a été conçu de telle façon qu'il y en a une différence importante entre les graphes de composition abstrait et concret comme le montre la figure.

LO_922 : Capacité à être adapté élevée.

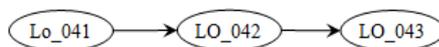
Graphe de composition concret



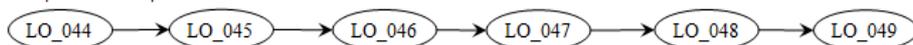
L'objet Lo_922 a le graphe de composition concret suivant qui produit dix graphes affichables différents.

LO_923 : Capacité à être adapté faible.

Graphe de composition abstrait



Graphe de composition concret



Le graphe de composition concret ne comporte aucun opérateur et donc la capacité d'adaptation est nulle.

LO_924 : Capacité à être réutilisé élevée.

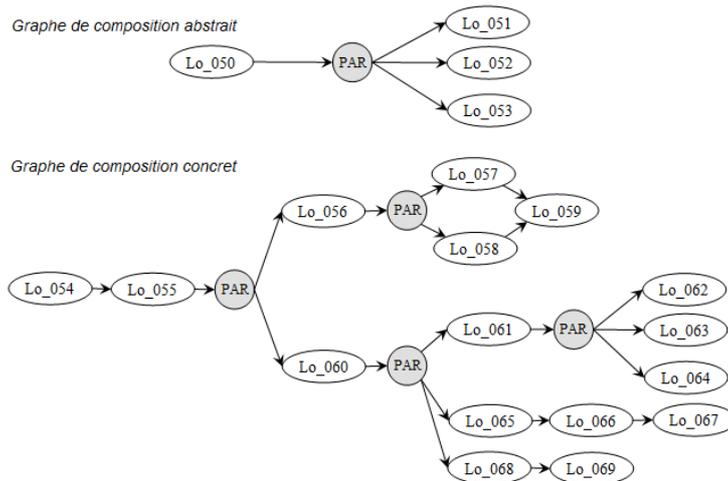
Cet objet est identique au niveau structurel à l'objet LO_921. Cependant son contenu traite un seul concept et ne demande aucun pré-requis. En plus, toutes les métadonnées sont renseignées.

LO_925 : Capacité à être réutilisé faible.

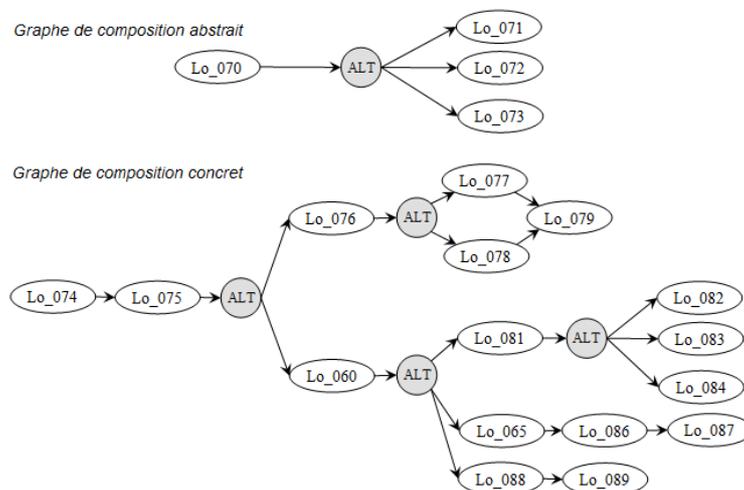
Cet objet a aussi les mêmes caractéristiques structurelles que l'objet LO_921. En revanche, son contenu et ses pré-requis portent sur dix concepts. La seule entrée de métadonnées renseignée est le titre.

LO_926 : Richesse didactique élevée.

Le graphe de composition concret de cet objet ne comporte pas d'opérateurs ALT et donc on a un seul graphe affichable. En plus, il comporte quatre rôles par concept et quatre sommets par rôle en moyenne.



LO_927 : Richesse didactique faible.



Cet objet a en moyenne un rôle par concept et quatre objets par rôle. En plus, seulement les deux premiers sommets sont partagés par tous les graphes affichables.

Si une valeur est entre 0 et 3,32 alors elle est considérée comme faible (F). Si elle est entre 3,33 et 6,65 alors elle est considérée comme moyenne (M). Finalement, si la valeur est entre 6,66 et 10 alors elle est considérée comme élevée (E).

Le résultat des tests est résumé dans le tableau ci-dessous. Chaque colonne comporte les mesures estimatives relatives à un objet d'apprentissage composé. Le titre de la colonne indique l'identifiant d'objet alors que chaque ligne indique l'aspect vérifié. La dernière ligne indique si les valeurs obtenues correspondent ou pas aux caractéristiques de chaque objet.

	<i>LO_920</i>	<i>LO_921</i>	<i>LO_922</i>	<i>LO_923</i>	<i>LO_924</i>	<i>LO_925</i>	<i>LO_926</i>	<i>LO_927</i>
<i>Concordance structurelle</i>	E (10)	F (3,16)	M (5,31)	F (1,83)	F (3,16)	F (3,16)	F (3,16)	F (3,16)
<i>Capacité à être adapté</i>	F (1,25)	M (5,25)	E (7,83)	F (0)	M (5,25)	M (5,25)	M (5,25)	M (5,25)
<i>Capacité à être réutilisé</i>	M (6,26)	M (6,26)	M (6,26)	M (6,26)	E (9,8)	F (2,19)	M (6,26)	M (6,26)
<i>Richesse didactique</i>	M (5)	E (7,5)	F (0,18)	M (5)	M (5,16)	M (5,16)	E (8,33)	F (2,8)
Correspondance	<input checked="" type="checkbox"/>							

Comme on peut le constater l'évaluation automatique donne des estimations correctes concernant les caractéristiques des objets d'apprentissage. La précision des valeurs peut être améliorée selon le contexte, le domaine et l'avis des experts en réglant les coefficients de chaque formule d'évaluation.

7.3. Analyse spécifique de la sémantique du contenu

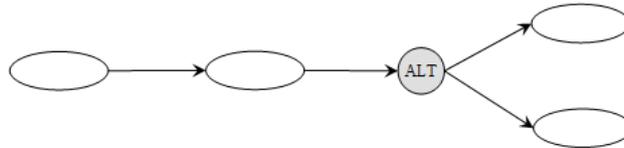
Afin d'expérimenter l'analyse spécifique de la sémantique du contenu nous avons procédé de la même façon que dans l'expérimentation de l'analyse multicritères. En effet, nous avons préparé des objets d'apprentissage avec des caractéristiques particulières par rapport aux trois propriétés objets de l'analyse : complexité sémantique, continuité sémantique et déséquilibre sémantique du contenu.

Par la suite nous avons soumis ces différents objets à l'analyse afin de vérifier que les valeurs des indicateurs calculés automatiquement sont bien ceux attendu. Nous avons fixé les valeurs des coefficients à un et les mesures sont ramenées à une échelle de zéro à dix.

7.4. Tests et résultats

Pour ces tests nous avons considéré des espaces sémantiques de taille 2. Pour calculer la complexité sémantique du contenu nous avons considéré que le seuil de complexité maximal pour les concepts (α) est égal à 10 et que le seuil de complexité maximal pour les composantes sémantiques (β) est également égal à 10.

Tous les objets utilisés dans cette simulation ont le même graphe de composition abstrait :



Voici la liste des objets expérimentés et leurs caractéristiques, sachant que nous avons utilisé l'ontologie de l'algorithmique (voir le chapitre V) :

LO_930 : Complexité sémantique élevée.

En fait, l'espace sémantique de cet objet comporte dix concepts répartis en dix composantes sémantiques. Ces valeurs sont celles des seuils maximaux de complexité α et β .

LO_931 : Complexité sémantique faible.

L'espace sémantique de cet objet contient un seul concept et donc on a une seule composante sémantique.

LO_932 : Continuité sémantique élevée.

L'espace sémantique de cet objet contient treize concepts et cinq composantes sémantiques. Toutefois, les concepts font partie d'un seul graphe connexe et on a sept intersections entre les composantes sémantiques.

LO_933 : Continuité sémantique faible.

L'espace sémantique de cet objet (qui contient vingt quatre concepts) est éclaté en trois graphes connexes et il n'y a qu'une seule intersection entre les quatre composantes sémantiques.

LO_934 : Déséquilibre sémantique élevée.

Le poids maximal des concepts est quatre alors que le poids minimal est un. Ces mêmes valeurs correspondent aux poids maximal et minimal des composantes sémantiques. Le

graphe de composition abstrait de l'objet comporte quatre sommets de type objet d'apprentissage.

LO_935 : Déséquilibre sémantique faible.

Les poids maximal et minimal des concepts et des composantes sémantiques est un. Le graphe de composition abstrait de l'objet comporte quatre sommets de type objet d'apprentissage.

Si une valeur est entre 0 et 3,32 alors elle est considérée comme faible (F). Si elle est entre 3,33 et 6,65 alors elle est considérée comme moyenne (M). Finalement, si la valeur est entre 6,66 et 10 alors elle est considérée comme élevée (E).

	<i>LO_930</i>	<i>LO_931</i>	<i>LO_932</i>	<i>LO_933</i>	<i>LO_934</i>	<i>LO_935</i>
<i>Complexité</i>	E (10)	F (0)	E (8,49)	E (8,01)	M (5,73)	F (3,01)
<i>Continuité</i>	F (0,5)	M (10)	E (10)	F (1,81)	M (3,33)	M (10)
<i>Déséquilibre</i>	M (4,3)	F (0)	M (4,3)	M (4,3)	E (8,61)	F (0)
<i>Correspondance</i>	<input checked="" type="checkbox"/>					

Comme on peut le constater l'évaluation automatique donne des estimations correctes concernant les caractéristiques des objets d'apprentissage au niveau sémantique. La précision des valeurs peut être améliorée selon le contexte, le domaine et l'avis des experts en réglant les coefficients de chaque formule d'évaluation et en réglant les seuils α et β .

8. Vérification de la conformité

8.1. Expérimentation

Afin d'expérimenter la vérification automatique de la conformité d'un objet d'apprentissage composé à un ensemble de règles de conformité nous avons considéré les règles suivants définissant un patron de conception :

R_210 =

N1 : SLO N2 : ANY ;

LO -> Content -> count('Concept') \leq 5 ;

LO -> AbstractCG -> abstractionLevels() \leq 4 ;

LO -> ConcreteCG -> count('LO') \leq 20 ;
 LO -> DeliveryGraph -> list('Role') \supseteq { 'introduction', 'Exemple', 'Exercice' } ;
 LO -> DeliveryGraph -> number() \leq 5 ;
 N1 -> Content -> list ('Role') \supseteq { 'Introduction' } ;
 N2 -> Content -> list ('Role') \supseteq { 'Exercice' } ;
 LO -> DeliveryGraph -> similarity('Beginning') \geq 2 ;

Ensuite, nous avons transformé ces règles en F-logique de la façon suivante :

FORALL LO,R,C,N

LO:R1Conforms <- LO[hasContent@(R)->>C] and count(LO,C,N) and lessorequal(N,5).

Tout objet d'apprentissage qui a un contenu qui traite le concept C selon le rôle R, le nombre de concepts dans le contenu de LO est N et N est inférieur ou égal à 5 alors c'est un objet conforme à la première règle de conformité.

Les autres règles fonctionnent selon le même principe :

FORALL LO, G, L, N

LO:R2Conforms <- LO[hasGraph->G] and p_g_levels(G,L) and count(G,L,N) and less(N,4).

FORALL LO, G, V, N

LO:R3Conforms <- LO[hasConcreteGraph->G] and p_cg_vertices(G,V) and V:AtomicLearningObject and count(G,V,N) and less(N,20).

FORALL LO, G

LO:R4Conforms <- LO[hasDeliveryGraph->>G] and p_dg_existsRole(G,"introduction") and p_dg_existsRole(G,"exercise") and p_dg_existsRole(G,"exemple").

FORALL LO, G, D, N

LO:R5Conforms <- LO[hasGraph->G] and p_g_dg(G,D) and count(G,D,N) and less(N,5).

FORALL LO, R, C1, C2, V

LO:R6Conforms <- LO[hasContent@("exercise")->>C1] and LO[hasRoot ->V] and V[hasContent@("introduction") ->> C2].

FORALL LO, R, G, N

LO:R7Conforms<- LO[hasDGrah->G] and p_dg_beginSimilarity(G,N) and lessorequal(2,N).

Pour le moment nous n'avons pas proposé un module automatique pour transformer des règles en langage formel vers des règles F-logique. Nous nous sommes plutôt intéressés à la faisabilité de la vérification automatique de la conformité. La définition des prédicats de haut-niveau que nous avons utilisés pour implémenter les règles de conformité sont données au niveau de l'annexe B.

8.2. Tests et résultats

Afin de vérifier la capacité du système à détecter la conformité ou la non-conformité à un patron de conception, nous avons défini un ensemble d'objets d'apprentissage dont chacun viole l'une de ses règles :

LO_1021 : Objet qui traite sept concepts au niveau du contenu et donc qui viole la première règle de conformité.

LO_1022 : Objet qui comporte cinq niveaux d'abstraction et donc qui viole la deuxième règle de conformité.

LO_1023 : Objet qui comporte vingt trois objets d'apprentissage au niveau du graphe de composition concret et donc qui viole la troisième règle de conformité.

LO_1024 : Objet qui comporte un graphe affichable qui ne comporte pas le rôle « Exemple » et donc qui viole la quatrième règle de conformité.

LO_1025 : Objet qui comporte six graphes affichables et donc qui viole la cinquième règle de conformité.

LO_1026 : Objet avec un graphe de composition dont le sommet racine est un objet d'apprentissage composé et donc qui viole la sixième règle de conformité.

LO_1027 : Objet dont les graphes affichables ne partagent que le premier sommet et donc qui viole la septième règle de conformité.

La soumission de ces objets à notre système a permis de vérifier que les algorithmes qui ont été définis permettent effectivement de vérifier la conformité d'un objet d'apprentissage composé à un ensemble de règles de conformité.

9. Conclusion

Au niveau de ce chapitre nous avons étudié la possibilité d'utiliser F-logique comme support de modélisation et l'OntoBroker comme moteur d'inférence afin de vérifier notre approche d'assistance aux auteurs d'objets d'apprentissage composés.

Comme approche de test nous avons effectué des simulations avec à chaque fois un ensemble d'objets d'apprentissage afin de vérifier le comportement du système. Les résultats ont permis de confirmer que le système retourne les résultats attendu.

Nous avons également présenté dans ce chapitre le système COLORS qui est une banque d'objet d'apprentissage. Nous avons montré via ce système la possibilité d'intégrer certains éléments de notre approche dans des systèmes outre les environnements de conception d'objets d'apprentissage. Pour le moment nous avons réussi d'intégrer dans COLORS certains éléments relatifs à l'analyse comparative des métadonnées informationnelles. Ceci s'explique par le fait que pour le moment les formats utilisés pour coder les objets d'apprentissage ne comportent pas une structure complexe et pas de métadonnées sémantiques.

CONCLUSION GENERALE

Les travaux de recherche que nous avons présentés concernent le domaine de la formation en ligne. Nous nous sommes intéressés particulièrement à la composition des objets d'apprentissage par réutilisation et assemblage d'objets existants. Si ce mode de composition présente plusieurs avantages il se confronte à plusieurs problématiques. C'est dans ce cadre que se situe l'approche SIMBAD qui propose un modèle structurel et une description sémantique appropriés à ce mode de composition d'objets d'apprentissage.

Toutefois, si les éléments structurels et sémantiques proposés présentent un bon support pour la composition par réutilisation et assemblage, ils introduisent de nombreux risques. La solution que nous apportons dans ce contexte est une approche d'assistance pour les auteurs d'objets d'apprentissage composés.

1. Bilan des travaux

Nos travaux de recherche ont permis tout d'abord d'identifier, suite à l'étude de l'état de l'art sur les objets d'apprentissage, les phases nécessaires pour assister les auteurs dans le processus de conception d'objets d'apprentissage par réutilisation d'objets existants. Ces phases sont : la recherche d'objets à réutiliser, la composition du nouvel objet d'apprentissage, l'aide à la compréhension, la génération automatique de certaines métadonnées et la vérification de conformité par rapport à un patron de conception. Si les phases de recherche, de composition et de génération automatique de métadonnées ont fait l'objet de certains travaux, les phases d'aide à la compréhension et la vérification de la conformité représentent le cœur de ce travail de recherche.

Toutefois, nous avons commencé par étudier l'aspect cohérence structurelle et sémantique des objets composés par assemblage d'objets existants. Concernant la cohérence structurelle elle est fortement liée au modèle de composition tel que SIMBAD et peut être contrôlée au moment même de la composition. C'est au niveau de la cohérence sémantique que nous avons identifié dans un premier temps les risques et dans un deuxième temps les règles à respecter.

Dans ce cadre nous avons également proposés les algorithmes permettant de vérifier la cohérence de la composition.

Concernant l'aide à la compréhension de l'objet d'apprentissage composé nous avons distingué entre deux types d'analyses : l'analyse générale et l'analyse spécifique de la sémantique du contenu. Dans le cadre de l'analyse générale nous avons proposé des métriques permettant de récupérer des indicateurs sur les propriétés de l'objet. Par la suite, nous avons proposé des fonctions d'agrégation permettant d'examiner le degré de concordance entre le niveau abstrait et le niveau concret du graphe de composition, la capacité de l'objet à être adapté par rapport aux profils d'apprenants, la capacité à être réutilisé dans la composition d'autres objets et la consistance didactique de l'objet. Pour l'analyse spécifique de la sémantique du contenu nous avons introduit les notions d'espace sémantique et de composante sémantique. Nous avons utilisé ces deux notions afin de pouvoir donner des estimations concernant la complexité, la continuité et le déséquilibre sémantique du contenu.

Nous avons également proposé une phase pour la vérification de la conformité de l'objet composé par rapport à un patron de conception. Dans ce cadre nous avons étudié tout d'abord la nature des règles qui peuvent être proposées. Ainsi, nous avons identifié des règles qui couvrent les aspects métadonnées, structurels, sémantiques et hybrides. Ensuite, nous nous sommes penché sur la formulation et la communication de ces règles. C'est dans ce cadre que nous avons introduit un langage formel et une notation graphique.

2. Résultats obtenus

Au delà de l'étude du processus de composition des objets d'apprentissage par réutilisation et le fait d'en déduire que l'assistance aux auteurs peut être assurée à plusieurs niveaux appelés phases, nous avons obtenu divers résultats.

En effet, nous avons proposés des *métriques* afin de calculer des *indicateurs* élémentaires sur les propriétés d'un objet d'apprentissage composé. Ces indicateurs couvrent les propriétés structurelles, sémantiques et le renseignement des métadonnées. Dans ce cadre les notions d'*espace sémantique* et de *composante sémantique* ont été introduites et définies.

De plus, nous avons proposé diverses *formules* qui permettent d'utiliser les métriques pour calculer les indicateurs qui couvrent plusieurs aspects, à savoir le degré de concordance entre le niveau abstrait et le niveau concret du graphe de composition, la capacité de l'objet à être adapté par rapport aux profils d'apprenants, la capacité à être réutilisé dans la composition

d'autres objets, la consistance didactique de l'objet, la complexité sémantique du contenu, la continuité sémantique du contenu et le déséquilibre sémantique du contenu. Ces différentes formules relatives aux indicateurs peuvent être adaptées au contexte d'utilisation en modifiant les coefficients et donc l'impact des métriques sur les valeurs des indicateurs.

Nous avons également proposés plusieurs *algorithmes*. En effet, le premier algorithme proposé est celui qui permet de transformer un objet SCORM en un objet SIMBAD, ce qui nous permet de pouvoir utiliser nos propositions sur un plus grand nombre d'objets (même si on ne peut pas alors tirer profit de toute la richesse du modèle SIMBAD). Nous avons également défini des algorithmes permettant de vérifier les différents aspects relatifs à la cohérence structurelle et sémantique des objets d'apprentissage composés. En plus, nous avons défini l'algorithme permettant de calculer l'espace sémantique d'un objet d'apprentissage donné.

Nous avons également proposé un langage et une notation pour exprimer les patrons de conception. Le langage formel permet de passer d'un énoncé en langage naturel en un énoncé précis et concis formulé en se basant sur une grammaire BNF. La notation est graphique et se distingue par le fait qu'elle est plus adaptée pour la communication des règles aux auteurs. En effet, elle offre une organisation par catégorie des règles et une explication visuelle des règles hybrides dont l'énoncé en langage formel est compliqué.

L'ensemble des algorithmes proposés a été incorporé dans un environnement auteur construit en utilisant la programmation logique via le langage F-logique et le moteur d'inférence associé OntoBroker. Cela nous a permis de commencer à valider les métriques et algorithmes proposés en les appliquant sur un ensemble d'objets d'apprentissage composés construits de manière adhoc. Cet environnement auteur est pour l'instant indépendant mais pourrait être couplé à une banque d'objets d'apprentissage comme celle de COLORS que nous avons développé par ailleurs et qui permet de fédérer un ensemble de banques stockées sur des sites distants. Cela fait d'ailleurs partie de nos perspectives.

3. Perspectives

La première perspective de notre travail est la validation en situation « réelle » de nos propositions. Pour cela nous comptons tout d'abord sélectionner un benchmark d'objets d'apprentissage composés représentatifs de la diversité de situations que nous avons décrits dans ce travail que nous allons faire analyser par un groupe d'experts et comparer cette analyse à celle produite par nos outils. Cela nous permettra de mieux calibrer nos métriques et

les formules d'indicateurs en plus de valider globalement nos propositions. Ensuite nous pourrions conduire une analyse d'usage en mettant à disposition d'auteurs de nos institutions (tant à Télécom SudParis qu'à l'Université Virtuelle de Tunis) notre environnement. Cette analyse d'usage devrait nous permettre de mieux comprendre comment les auteurs construisent leurs cours, quels types d'analyse ils/elles privilégient et à quels critères de « qualité » ils/elles sont les plus sensibles. De plus, cela nous donnera des retours intéressants sur les aspects d'interface de nos propositions, visualisation des résultats d'analyse, notation graphique pour les règles de conformité, aspect itératif du processus.

Concernant les patrons de conception, plusieurs réflexions restent à mener. Tout d'abord il va falloir valider l'expressivité du langage conformément aux usages attendus. Ensuite, il pourrait être intéressant d'étudier comment intégrer ces règles (ou tout au moins les règles hybrides) dans le processus même de construction d'un objet composé. En effet, il est dommage d'attendre la fin du processus d'analyse pour vérifier l'adéquation à un ensemble de règles. La définition des règles risque d'être perçue comme complexe par les experts. La notation graphique est une première réponse à ce problème. Une deuxième réponse pourrait être apportée par des analyses statistiques sur les objets déjà existants et fortement utilisés. Ces analyses pourraient fournir des « profils » type de « bons » objets exprimés sous forme de règles de conformité.

Les indicateurs (produits par les formules que nous avons définies) peuvent être régulés. En effet, chaque expert peut choisir les valeurs de coefficients qui lui semblent appropriés pour chaque formule. En plus, certaines formules offrent des paramètres supplémentaires permettant d'influencer leur comportement (tels que les seuils de complexité dans le calcul de la complexité sémantique du contenu). Ainsi, une de nos perspectives consiste à étudier le processus de choix de ces paramètres. En particulier comment faut-il procéder dans un contexte donné pour choisir les valeurs qui permettent d'obtenir des valeurs d'indicateurs les plus fidèles à la réalité. Et est-ce que ces formules qui définissent les indicateurs sont universelles ou plutôt dépendantes de certains facteurs ? Si c'est le deuxième cas, alors quels sont ces facteurs et de quelle façon influencent-ils les valeurs des indicateurs ?

La définition des patrons de conception par des intervenants divers et à des échelles différents (Institution, Département, etc.) peut entraîner un risque d'incohérence. En effet, il se peut que par inadvertance deux règles contradictoires au sein d'un même patron de conception soient édictées par deux instances différentes. Ainsi, il nous semble que dans ce contexte il faut étudier le problème de la cohérence des règles de conformité.

Plusieurs perspectives se présentent également au niveau implémentation. En particulier, nous comptons améliorer le système COLORS. Dans un futur proche nous comptons intégrer un module de génération automatique de métadonnées sémantiques pour les objets IMS et SCORM importés. Pour cela, nous comptons utiliser des techniques de fouille de textes et les appliquer aux métadonnées et aux informations textuelles disponibles au niveau de l'objet d'apprentissage.

Références

[ADL 2004a] Advanced Distributed Learning (ADL), “Sharable Content Object Reference Model (SCORM®) 2004 2nd Edition Overview”, [Online:] <http://www.adlnet.org/>, 2004.

[ADL 2004b] Advanced Distributed Learning (ADL), “Sharable Content Object Reference Model (SCORM®) Content Aggregation Model (CAM) Version 1.3.1”, [Online:] <http://www.adlnet.org/>, 2004.

[ADL 2004c] Advanced Distributed Learning (ADL), “Sharable Content Object Reference Model (SCORM®) Run-Time Environment (RTE) Version 1.3.1”, [Online:] <http://www.adlnet.org/>, 2004.

[ADL 2004d] Advanced Distributed Learning (ADL), “Sharable Content Object Reference Model (SCORM®) Sequencing and Navigation (SN) Version 1.3.1”, [Online:] <http://www.adlnet.org/>, 2004.

[ADL 2009] Advanced Distributed Learning (ADL), “Sharable Content Object Reference Model (SCORM®) 2004 4th Ed. Compliance Requirements Version 1.0”, 2009.

[AFNOR 2006] AFNOR, « Technologies de l’information pour l’éducation, la formation et l’apprentissage – Profil français d’application du LOM (LOMFR) – Métadonnées pour l’enseignement », Norme NF Z76-040, 2006.

[Azevedo et al., 2008] Azevedo, I. Carrapatoso, E. Carvalho, C.V. “A framework to scaffold the reuse of learning objects”. In the proceedings of ICALT’08, Santander, Spain, 1-5 July, 2008.

[Bertin 2006] Bertin, G. “*Sharable Content Object Reference Metadata (SCORM)*”, Babel - edit -, L’indexation des ressources pédagogiques numériques, ENSSIB, [en ligne] <http://babel.enssib.fr/document.php?id=83>, 2004.

[Bouras 2003] Bouras, Ch. Nani, M. Tsiatsos, Th. “*Building Reusable and Interactive e-learning content using Web*”, The 2nd International Conference on Web-based Learning (ICWL 2003), [Online] <http://ru6.cti.gr/ru6/publications/5605906.pdf>, 2003.

[Bourda 2001] Bourda, Y. “*Objets pédagogiques, vous avez dit objets pédagogiques ?*”, Actes du congrès GUTenberg, 2001.

[Bouzeghoub et al. 2003] Bouzeghoub, A., Carpentier, C., Defude, B., Duitama, F., “A model of reusable educational component for the generation of adaptive courses”. Proc. of the First international Workshop on Semantic Web for Web-based Learning in conjunction with CAISE'03 conference, Klagenfurt, Austria, 2003.

[Bouzeghoub et al. 2004] Bouzeghoub, A., Defude, B., Ammour, S., Duitama, F. AND Lecocq, C., 2004. A RDF description model for manipulating learning objects. In: 4th IEEE International Conference on Advanced Learning Technologies ICALT' 2004 (Joensuu, Finland, 2004).

[Bouzeghoub et al. 2006] Bouzeghoub, A. Defude, B. Duitama, F. Lecocq, C. « Un modèle de description sémantique de ressources pédagogiques basé sur une ontologie de domaine », Revue STICEF, Volume 12, 2005, ISSN : 1764-7223, mis en ligne le 15/03/2006, <http://sticef.org>

[Bouzeghoub et Selmi, 2009] Bouzeghoub, A. and Selmi, M. “Authoring Tool for Structural and Semantic Coherence Validation of Composed Learning Objects”, the 9th IEEE International Conference on Advanced Learning Technologies (ICALT 2005), Riga, Latvia, July 14-18, 2009.

[Broeglin, 2003] Broeglin, D. « Composition incrémentale de documents Virtuels », Colloque EIAH2003 sur les environnements informatiques pour l'apprentissage humain, en ligne (consulté le 26/12/2009) : http://archive.eiah.univ-lemans.fr/EIAH2003/Pdf_annexes/Broeglin.pdf, 2003.

[Brugia, 2005] Brugia, M. “Pratiques innovantes en formation et enjeux pour la professionnalisation des acteurs”, Cedefop Panorama series n°119, ISBN 92-896-0341-0, 2005.

[Brusilovsky, 1996] Brusilovsky, P., « Methods and techniques of adaptive hypermedia user modeling and user adapted interaction », Special issue on adaptive hypertext and hypermedia, 1996.

[Brusilovsky et al., 1996] Brusilovsky, P. Schwarz, E. and Weber. G. « An intelligent tutoring system on World Wide Web”. Third International Conference on Intelligent Tutoring Systems, ITS-96, in ELM-ART, 1996.

[Brusilovsky et al., 1998] Brusilovsky, P. Eklund, J. and Schwarz, E. “Web-based education for all: A tool for developing adaptive courseware”. Proceedings of Seventh International World Wide Web Conference, 14-18 April, 1998.

[Brusilovsky, 2003] Brusilovsky, P. “Adaptive navigation support in educational hypermedia: The role of student knowledge level and the case for meta-adaptation”. British Journal of Educational Technology, Vol. 34, n°4, pp. 487-497, 2003.

[Burgos et al., 2005] Burgos, D. Arnaud, M. Neuhauser P. et Koper, R. “IMS Learning Design : la flexibilité pédagogique au service des besoins de l'e-formation”, consulté le 13/09/2009 et accessible à : <http://www.epi.asso.fr/revue/articles/a0512c.htm>

[Catteau et al., 2006] Catteau, O. Vidal, P. and Broisin, J. “A generic representation allowing for expression of learning object and metadata lifecycle”. In the proceedings of the 6th IEEE Internatioanl Conference on Advanced Learning Technologies ICALT'06, Kerkrade, The Netherlands, 5-7 July, 2006.

[Catteau, 2008] Catteau, O. «Le Cycle de Vie de l'Objet Pédagogique et de ses Métadonnées », Thèse soutenue à l'Université de Toulouse III – Paul Sabatier, 2008.

[Celebrate, 2003] Celebrate, “Celebrate Application Profile, version 1.1”, [On line]: http://goedel.uiah.fi/projects/calibrate/attachment/wiki/MetadataStandards/CELEB_app_profile.pdf, 2003.

[CELTS profile, 2003] CELTS application profile, accessible de : http://mdlet.jtc1sc36.org/doc/SC36_WG4_N0059.pdf, 2003.

[Cheniti, 2008] Cheniti, L. “Assessment personalization in the semantic Web”, Thèse de doctorat, Faculté des Sciences de Tunis, 2008.

[Cherhal, 2004] Cherhal, E. « Le Dublin Core (DC) ». Babel - edit -, L'indexation des ressources pédagogiques numériques (journée d'étude du 16/11/ 2004). ENSSIB - janvier 2006 [en ligne] <http://babel.enssib.fr/document.php?id=145>

[Chiali et al., 2006] Chiali, S. Eberichi, Z. and Malki, M. “A dynamic composition of adaptive course”, Proceedings of world academy of science, engineering and technology, Volume 11, ISSN 1307-6884, 2006.

[Chorfi et Jemni, 2004] Chorfi, H. and Jemni, M. “PERSO: Towards an adaptative e-learning system”. Journal of Interactive Learning Research , 15 (4), pp 433-447, 2004.

[Chorfi, 2007] Chorfi, H. “Personnalisation de cours dans un environnement d’Enseignement à Distance à travers Internet”, Thèse de doctorat, Ecole Nationale des Sciences de l’Informatique, Université de la Mannouba, 2007.

[Chouinard, 2002] Chouinard, R. “Les normes et standards de la formation en ligne : état des lieux et enjeux”, Sous-comité sur les technologies de l’information et de la communication (SCTIC), accessible à : <http://www.crepuq.qc.ca/IMG/pdf/norm-0210-d-rapport.pdf>, 2002.

[Collier et Robson, 2002] Collier, G. and Robson, R. “e-Learning interoperability standards”, available at: http://www.sun.com/solutions/documents/white-papers/ed_interop_standards.pdf, Sun Microsystems, Inc., 2002.

[CREPUQ, 2003] CREPUQ, Novasys inc., « La description normalisée des ressources : vers un patrimoine éducatif », NORMETIC, version 1.0, 2003.

[De Bra et al., 1999] De Bra, P. Brusilovsky, P. and Houben, G. « Adaptive Hypermedia : From Systems to Framework ». ACM Computing Surveys, Symposium Edition, 1999.

[De La Passardière et Jarraud, 2004] De La Passardière, B. Jarraud, P. « ManUeL, un profil d’application du LOM pour C@mpusSciences », revue Sticef , Recueil 2004, vol.11, p. 11-57, ISBN : 2-7342-1000-2, 2004.

[De La Passardière et Jarraud, 2005] De La Passardière, B. Jarraud, P. « LOM et l’indexation de ressources scientifiques. Vers de bonnes pratiques pour l’Université en Ligne », Actes de la conférence EIAH 2005, ISBN : 2-7342-0999-3, 2005.

[Defude et Farhat, 2005] Defude, B. and Farhat, R. "A Framework to Design Quality-Based Learning Objects", 5th IEEE International Conference on Advanced Learning Technologies (ICALT 2005), Kaohsiung, Taiwan, 5-8th July, 2005.

[Derntl and Motschnig-pitrik, 2003] Derntl, M. and Motschnig-pitrik, R. “Conceptual Modeling of Reusable Learning Scenarios for Person-centered E-learning”, in proceedings of Int. Workshop for Interactive Computer-Aided Learning (ICL’03), 2003.

[DOC’INSA, 2004] DOC’INSA, « Schéma LOM pour l’INSA de Lyon », Animation du Campus Ouvert de l’académie de Lyon (ANCOLY), Version en ligne : http://www.ancoly.net/normesetstandards/docs/LOM_INSA.pdf, 2004.

[Downes, 2001] Downes, S. “Learning objects: resources for distance education Worldwide”. International review of research in open and distance learning, ISSN: 1492-3831, July 2001. Accessible at: <http://irrodl.org/content/v2.1/downes.html>

[Downes, 2004] Downes, S. “The learning marketplace: meaning, metadata and content syndication in the learning object economy”. Version en ligne: <http://www.downes.ca/files/book3.htm>, 2004.

[Duitama et al., 2002] Duitama, F. Bouzeghoub, A. Carpentier, C. and Defude, B. « Une plate-forme de télé-enseignement a base de composants réutilisables et personnalisables ». In : 18èmes journées Bases de Données avancées, Evry, France, 2002.

[Duitama et al., 2003a] Duitama, F. Defude, B. Bouzeghoub, A. and Carpentier C. “Educational component model for adaptive Web-based courses”. In: Proceeding of IADIS WWW/Internet 2003 Conference, Algarve, Portugal, 2003.

[Duitama et al., 2003b] Duitama, F. Defude, B. Bouzeghoub, A. and Carpentier, C. “A framework for the generation of adaptive courses based on semantic metadata”. In: Proceeding of the Workshop on Metadata and Adaptability in Web-based Information Systems in conjunction with the 9th International Conference on Object-Oriented Information Systems OOIS 2003, Geneva, Switzerland, 2003.

[Duitama, 2005] Duitama, F. “Un modèle de composants pédagogiques pour des cours adaptatifs sur le Web”, Thèse de doctorat, Institut National des Télécommunications d’Evry, 2005.

[Duval et al., 2002] Duval, E. Hodgins, W. Stuart, S. Stuart, L. “Weibel, Metadata Principles and Practicalities”, D-Lib Magazine, Volume 8 Number 4, Online: <http://www.dlib.org/dlib/april02/weibel/04weibel.html>, 2002.

[Duval et al., 2006] Duval, E. Smith, N. and Coillie, M. “Application profiles for learning”, 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006), Kerkrade, The Netherlands, July 5-7, 2006.

[Ehlers et al., 2006] Ehlers, U. Goertz, L. Hildebrandt, B. et Pawlowski, J. “La qualité en e-learning : Utilisation et diffusion des approches qualité en e-learning en Europe”, Cedefop Panorama series n°121, ISBN 92-896-0411-5, 2006.

[Ely, 2005] Ely, M. « Normes et plates-formes de la formation en ligne : la participation des PVD ». Actes du colloque Initiatives 2005 [en ligne], Débat thématique 2, 2 mars 2007. Disponible sur Internet : <http://www.initiatives.refer.org/Initiatives-2005/document.php?id=170>.

[Farhat, 2004] Farhat, R. « Un modèle de typage pour les métadonnées éducatives : Application au projet SIMBAD », Rapport de master recherche, Université d'Evry Val d'Essonne, 2004.

[Farhat et al., 2006a] Farhat, R. Jemni, M. et Defude, B. "*Indicateurs de Qualité dans les Objets d'apprentissage*", 4^{ème} conférence internationale JTEA, Hammamet, Tunisie, 12-14 Mai, 2006.

[Farhat et al., 2006b] Farhat, R. Jemni, M. and Defude, B. "Toward Pedagogy-based Learning Objects", International Conference on Distance Education (ICODE 2006), Muscat, Oman, , 27-29th March, 2006.

[Farhat et al., 2007a] Farhat, R. Jemni, M. and Defude, B. "An approach to evaluate learning objects properties", Interactive Mobile and Computer Aided Learning international conference (IMCL 2007), Amman, Jordan, 18-20 April, 2007.

[Farhat et al., 2007b] Farhat, R. Jemni, M. and Defude, B. "Technology assisted improvement of learning objects quality", the first international conference on Information and Communication Technology and Accessibility (ICTA 2007), Hammamet, Tunisia, 12-14 April, 2007.

[Farhat et al., 2009] Farhat, R. Defude, B. and Jemni, M. "Authoring by Reuse for SCORM like Learning Objects", the 9th IEEE International Conference on Advanced Learning Technologies (ICALT 2009), Riga, Latvia, 14-18 July, 2009.

[Farhat et al., 2010] Farhat, R. Defude, B. and Jemni, M. "Checking semantic consistency of SCORM like learning objects", 10th IEEE International Conference on Advanced Learning Technologies (ICALT 2010), Sousse, Tunisia, 5-7 July, 2010.

[Farhat et Jemni, 2005] Farhat, R. and Jemni, M. "*A Tool to Edit and Control LOM based Learning Object's Metadata*", 1st International E-Business Conference (IEBC 2005), Hammamet, Tunisia, 23-25 June, 2005.

[Farhat et Jemni, 2006] Farhat, R. et Jemni, M. "Evaluation de la Qualité des Objets d'apprentissage", 3^{ème} Symposium International en EAD, Hammam Sousse, Tunisie, 25 - 27 mai, 2006

[Farhat et Jemni, 2009] Farhat, R. and Jemni, M. "COLORS: COoperative Learning Object Repository System", the second international conference on Information and Communication Technology and Accessibility (ICTA 2009), Hammamet, Tunisia, 7-9 Mai, 2009.

- [Fensel, 2000] Fensel, D. “Relating Ontology Languages and Web Standards“. In: Informatik und Wirtschaftsinformatik. Modellierung 2000, Foelbach Verlag, 2000.
- [Ghebghoub et al., 2008] Ghebghoub, O. Abel, M. and Moulin, C. “Learning Object Indexing Tool Based on LOM Ontology”. In the proceedings of ICALT’08, Santander, Spain, 1-5 July, 2008.
- [Garlatti et Prié, 2004] Garlatti, S. et Prié, Y. « Adaptation et personnalisation dans le Web sémantique », in Revue I3 Information-Interaction - Intelligence, Numéro spécial Web sémantique, 2004.
- [Gruber, 1993] Gruber, T. R. “A translation approach to portable ontologies. Knowledge Acquisition”, Vol. 5, No. 2, 199-220, 1993. Accessible à : http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html
- [Gançarski et al., 2007] Gançarski, A. Bouzeghoub, A. Defude, B. Lecocq, C. “Iterative search of composite learning objects”. In IADIS International Conference WWW/Internet 2007, Vila Real, Portugal, 5-8 octobre 2007.
- [Henze et al., 2004] Henze, N. Dolog, P. and Nejd, W. “Reasoning and Ontologies for Personalized E-Learning in the Semantic Web”, Educational Technology & Society, 7 (4), p. 82-97, 2004.
- [Honey et Mumford, 1982] Honey, P. and Mumford, A. “Manual of Learning Styles”. Maidenhead, 1982.
- [IEEE, 2002] IEEE Computer Society, Learning Technology Standards Committee. "IEEE Standard for Learning Object Metadata, IEEE Std 1484.12.1." The Institute of Electrical and Electronics Engineers, Inc., 2002.
- [IMS, 2002] IMS Global Learning Consortium, Inc. “IMS Learning Design Information Model”, 2002.
- [IMS, 2003a] IMS Global Learning Consortium, Inc. “IMS Simple Sequencing Best Practice and Implementation Guide – Version 1.0 Final Specification”, 2003.
- [IMS, 2003b] IMS Global Learning Consortium, Inc. “IMS Simple Sequencing Information and Behavior Model – Version 1.0 Final Specification”, 2003.
- [IMS, 2003c] IMS Global Learning Consortium, Inc. “IMS Learning Design Best Practice and Implementation Guide – Version 1.0 Final Specification”, 2003.

[IMS, 2004] IMS Global Learning Consortium, Inc. “IMS Content Packaging Information Model – Version 1.1.4 Final Specification”, 2004.

[Janssens, 2002] Janssens, J. « Innovations in lifelong learning, capitalising on ADAPT », Cedefop, Panorama series n° 25, ISBN 92-896-0129-9, 2002.

[Koper, 2001] Koper, R. “Modeling units of study from a pedagogical perspective – The pedagogical meta-model behind EML”, Educational Technology Expertise Center, Open University of the Netherlands, 2001.

[L’Allier, 1997] L’Allier, J. “Frame of Reference: NETg’s Map to the Products, Their Structure and Core Beliefs”, accessible à : <http://www.netg.com/research/whitepapers/frameref.asp>

[Lawless et al., 2008] Lawless, S. Hederman, L. and Wade, V. “OCCS: Enabling the dynamic discovery, harvesting and delivery of educational content from open corpus sources”. In the proceedings of ICALT’08, Santander, Spain, 1-5 July, 2008.

[Latorre et al., 2008] Latorre, M. Garcia-Sevilla, F. Lopez, E. Martin, S. Gil, R. Lamas, M. Caeiro, M. Castro, M. and Peire, J. “A good practice example on learning object reutilization”. In the proceedings of ICALT’08, Santander, Spain, 1-5 July, 2008.

[Lehmann et al., 2006] Lehmann, L. Aqqal, A. Rensing, C. Chikova, P. Leyking, K. and Steinmetz, R. “A content modeling approach as basis for the support of the overall content creation process”. In the proceedings of the 6th IEEE Internatioanl Conference on Advanced Learning Technologies ICALT’06, Kerkrade, The Netherlands, 5-7 July, 2006.

[Lehmann et al., 2008] Lehmann, L. Hildebrandt, T. Rensing, C. and Steinmetz, R. “Capture, Management and Utilization of Lifecycle Information for Learning Ressources”, IEEE Transactions on learning technologies, Vol. 1, No. 1, 2008.

[Man et Thomson, 1987] Mann, W. and Thomson, S. “Rhetorical Structure Theory: a theory of text organisation”, Technical Report RS-87-190, Information Science Institute.

[Mellon, 2003] Mellon, C. « SCORM, Simple sequencing templates & models », Learning Systems Architecture Lab, 2003.

[Moreno-Ger et al., 2006] Moreno-Ger, P. Martinez-Ortiz, I. Sierra, J. and Fernandez-Manjon, B. “A descriptive markup approach to facilitate the production of e-learning content”. In the proceedings of the 6th IEEE Internatioanl Conference on Advanced Learning Technologies ICALT’06, Kerkrade, The Netherlands, 5-7 July, 2006.

- [Najjar et Duval, 2003] Najjar, J. Duval, E. « Mapping between ARIADNE and LOM », téléchargé le 14/10/2005, <http://www.ariadnee.org/common/docs/MappingBetweenAriadneAndLomV1.pdf>, 2003.
- [Nelson, 1965] Nelson, T. “A File Structure for the Complex, The Changing and The Intermediate”, ACM 20th National Conference, 1965.
- [Nesic et al., 2008] Nesic, S. Gasevic, D. and Jazayeri, M. “Samantic Document Management for Collaborative Learning Object Authoring”. In the proceedings of ICALT’08, Santander, Spain, 1-5 July, 2008.
- [Ochoa and Duval, 2007] Ochoa, X. and Duval, E. “Relevance Ranking Metrics for Learning Objects”. In the proceedings of the Second European Conference on Technology Enhanced Learning, EC-TEL, ISBN 978-3-540-75194-6, Crete, Greece, 2007.
- [Ontoprise, 2006a] “OntoBroker User Guide, version 4.3”, Ed. Ontoprise GmbH, 2006.
- [Ontoprise, 2006b] “How to write F-logic programs“, Tutorial series, Ed. Ontoprise GmbH, 2006.
- [Ostyn, 2003] Ostyn, C. “Cooking up a SCORM - A SCORM 1.2 Content Cookbook for Developers”, Learning Standards Strategist Click2learn, Inc., 2003. [Online:] http://www.scorm.com/wp-content/assets/cookbook/CookingUpASCORM_v1_2.pdf
- [Paquette et Rosca, 2002] Paquette, G. and Rosca, I. “Organic Aggregation of Knowledge Objects in Educational Systems”, Canadian Journal of Learning Technologies, Volume 28-3, 2002.
- [Passardière et al., 2004] De La Passardière, B. Jarraud, P. « ManUeL, un profil d’application de LOM pour C@mpuSciences », Revue STICEF – Sciences et Technologies de l’Information et de la Communication pour l’Education et la Formation, Volume 11, ISSN 176-7223, accessible à : http://sticef.univ-lemans.fr/num/vol2004/passardiere-11/sticef_2004_passardiere_11.htm, 2004.
- [Passardière et Jarraud, 2005] De La Passardière, B. Jarraud, P. « LOM et l’indexation de ressources scientifiques : vers des bonnes pratiques pour l’Université en Ligne », actes de la conférence EIAH, ISBN 2-7342-0999-3, 2005.
- [Polsani, 2003] Polsani, P. « Use and abuse of reusable learning objects », Journal of digital information, Volume 3 Issue 4, Article No. 164, 2003.

- [Prié et Garlatti, 2004] Prié, Y. Garlatti, S. « Annotations et métadonnées dans le Web sémantique », in Revue I3 Information-Interaction - Intelligence, Numéro Hors-série Web sémantique, 2004.
- [Ranwez, 2000] Ranwez, S. « Composition Automatique de Documents Hypermédia Adaptatifs à partir d'Ontologies et de Requêtes Intentionnelles de l'Utilisateur ». Thèse de doctorat en informatique, Université Montpellier II, 2000.
- [Ranwez et Ranwez, 2001] Ranwez, V. et Ranwez, S. « Algorithme pour la génération de documents personnalisés structurés ». Nîmes TIC'2001, 2001.
- [Rebai et al., 2008] Rebai, I. De La Passardière, B. and Labat J. « A Formalism to Describe Open Standards in order to Generate Application Profiles », in the proceedings of the 16th International Conference on Computers in Education, Taipei, Taiwan, 2008.
- [Rensing et al., 2005] Rensing, C. Bergsträßer, S. Hildebrandt, T. Meyer, M. Zimmermann, B. Faatz, A. Lehmann, L. and Steinmetz, R. "Re-Use and Re-Authoring of Learning Resources - Definitions and Examples". Technical report no. KOM-TR-2005-02, 2005.
- [Robert et al., 2003] Robert, M. Letourneur, S. et Roy, Y. « La description normalisée des ressources : vers un patrimoine éducatif », CREPUQ, Novasys Inc., accessible à : <http://profetic.org/normetic>, ISBN 2-89574-021-6, 2003.
- [Robson et McElroy, 2008] Robson, R. and McElroy, P. "Transforming to next generation digital learning content", Eduworks, White paper, accessible online: www.adobe.com/resources/elearning/pdfs/95010540_digitallearning_wp_ue.pdf, 2008.
- [Robson et Ostin, 2003] Robson, R. et Ostin, C. « La normalisation et ses défis : Le programme du comité pour les normes en technologie de l'apprentissage de l'IEEE », Eduworks, accessible en ligne : www.eduworks.com/Documents/Publications/LTSC_Versailles_3_14_03_fr.pdf, 2003.
- [Rojas and Defude, 2008] Rojas, J. Defude, B. "Improving Learning Objects Quality with Learning Styles", Proc IEEE ICALT '08, July 1-5, Santander, Spain, Los Alamitos, CA : IEEE Computer Society, 2008, pp. 496-497
- [Stojanovic et al., 2001] Stojanovic, L. Staab, S. and Studer, R. « eLearning based on the Semantic Web », World Conference on the WWW and Internet (WebNet2001), Orlando, Florida, USA, 2001

[Seyfried, 2007] Seyfried, E. « Indicateurs de qualité dans l'enseignement et la formation professionnels », Cedefop, Panorama series n° 146, ISBN 978-92-896-0474-1, 2007.

[Simard, 2002] Simard, C. « Normalisation de la formation en ligne, Enjeux, tendances et perspectives », Agence Universitaire Francophone, BAN, NordSud.org, 2002.

[Strijker, 2004] Strijker, A. « Reuse of Learning Objects in Context: Human and Technical Aspects », Thèse soutenue à l'Université de Twente, Enschede, PrintPartners Ipskamp, ISBN: 09-365-2090-9, 2004.

[Tazi et Altawki, 1999] Tazi, S. et Altawki, Y. « Création de documents virtuels, Cas des supports de cours », Workshop sur les Documents Virtuels Personnalisables, Journées IHM'99, Montpellier, 1999.

[Touzani et al., 2009] Touzani, N. Braham, R. and Khemaja, M. "Pedagogical scenario generation framework for elearning in the semantic web", the second international conference on Information and Communication Technology and Accessibility (ICTA 2009), Hammamet, Tunisia, 7-9 Mai, 2009.

[Trigano et Giacomini-Pacurar, 2004] Trigano, P. and Giacomini-Pacurar, E. "CEPIAH, a method for Evaluation and Design of Pegagogical Hypemedia", Higher Education in Europe, vol. XXIX, num. 1, pp. 115-132, April, 2004.

[UK LOM Core, 2003] UK Metadata for Education Group, "UK Learning Object Metadata Core", [On line]: http://www.cetis.ac.uk/profiles/uklomcore/uklomcore_v0p1.doc, 2003.

[Vidal et Broisin, 2005] Vidal, P. Broisin, J. « Fédération des ressources pédagogiques. Vers la virtualisation des ressources pédagogiques : une architecture fédérée de systèmes de gestion de contenus d'apprentissage », actes de la conférence EIAH, ISBN 2-7342-0999-3, 2005.

[Wiley, 2002] Wiley, D. "The Instructional Use of Learning Objects", the Agency for Instructional Technology and the Association for Educational Communications and Technology, ISBN: 0-7842-0892-1, 2002.

[Xiang and al., 2003] Xiang, X. Shen, Z. Guo, L. and Shi, Y. "Introduction of the Core Elements Set in Localized LOM Model", ISO/IEC JTC1 SC36 WG4, 2003.

ANNEXES

Annexe A : Extrait de la base de connaissances

Extrait de la hiérarchie de concepts au sein du modèle de domaine :

```
ComputerScience :: Concept.

    Database :: ComputerScience.

        ObjectRelationalDatabase :: Database.

        ObjectOrientedDatabase :: Database.

        RelationalDatabase :: Database.

            RelationalAlgebra :: RelationalDatabase.

            RelationalCalculus :: RelationalDatabase.

...

```

Extrait de la définition des relations rhétoriques :

```
// Extent: relation between Object Relational DB and Relational DB
FORALL X,Y
X[extent -> Y] <- X : ObjectRelationalDatabase AND Y : RelationalDatabase.

// Contrast: symmetrical relation between R. Algebra and R. Calculus
FORALL X,Y
X[contrast -> Y] <- X : RelationalAlgebra AND Y : RelationalCalculus.
FORALL X,Y
Y[contrast -> X] <- X[contrast -> Y].
...

```

Pour chaque objet F-logique il faut définir une instance, dont voici un extrait :

```
computerScience : ComputerScience.
informatique : ComputerScience.

    database : Database.

```

```
baseDeDonnees : Database.

    relationalDatabase : RelationalDatabase.

baseDeDonneesRelationnelle : RelationalDatabase.

    relationalAlgebra : RelationalAlgebra.

    algebreRelationnelle : RelationalAlgebra.

    relationalCalculus : RelationalCalculus.

    calculRelationnel : RelationalCalculus.

objectRelationalDatabase : ObjectRelationalDatabase.

baseDeDonneesRelationnelleObjet : ObjectRelationalDatabase.

objectOrientedDatabase : ObjectOrientedDataBase.

baseDeDonneesOrienteeObjet : ObjectOrientedDataBase.

...
```

Ces différentes descriptions sont décrites dans des fichiers spécifiques sous le répertoire « Model » dont le chemin d'accès est communiqué au serveur OntoBroker. Le modèle de domaine est décrit au niveau du fichier « DomainModel.flo » alors que les instances sont décrites au niveau du fichier « InstancesDomainModel.flo ».

Par rapport à ce modèle, nous avons défini quelques rôles comme suit :

```
introduction : Role.

exercice : Role.

definition : Role.

example : Role.

chapter : Role.
```

Il faut également ajouter des instances pour la classe KnowledgeLevel :

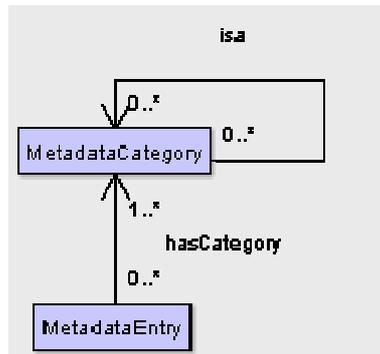
```
beginner : KnowledgeLevel.

medium : KnowledgeLevel.

expert : KnowledgeLevel.
```

D'autre part, il est faut introduire la notion de métadonnées classiques décrites par le standard LOM [IEEE 2002]. Les catégories et les attributs du standard sont décrits par le diagramme

de classe ci-dessous, sachant que nous avons mis le minimum de restrictions sur les cardinalités pour être compatible avec divers profils LOM :



Ce modèle a été traduit en F-logique comme suit :

```

MetadataCategory
[
    number => STRING;
    name => STRING;
    // indicate the allowed number of elements
    size => STRING;
    // indicate if the order of elements is significant
    order => STRING;
    // indicate the set of allowed values
    dataType => STRING
].

MetadataEntry
[
    hasCategory => MetadataCategory;
    hasValues ==>> STRING
]
  
```

Voici un extrait de la définition des instances de la classe MetadataCategory :

```

LOM_General :: MetadataCategory.
general : LOM_General.
  
```

```

general
[
    number -> 1;
    name -> "General";
    size -> 1;
    order -> unspecified
].

LOM_Identifier :: LOM_General.
identifier : LOM_General.
identifier
[
    number -> 1.1;
    name -> "Identifier";
    size -> 10;
    order -> unspecified
].

LOM_Category :: LOM_Identifier.
catalog : LOM_Identifier.
catalog
[
    number -> 1.1.1;
    name -> "Catalog";
    size -> 1;
    order -> unspecified;
    dataType -> CharacterString
].

LOM_Entry :: LOM_Identifier.
entry : LOM_Entry.
entry
[
    number -> 1.1.2;
    name -> "Entry";

```

```

        size -> 1;

        order -> unspecified;

        dataType -> CharacterString

    ].

LOM_Language :: LOM_General.

language : LOM_Language.

language

[

    number -> 1.3;

    name -> "Language";

    size -> 10;

    order -> unordered;

    dataType -> CharacterString

].

```

Dans cette instanciation la hiérarchie entre catégories LOM a été définie explicitement.

Voici un exemple de définition de métadonnées LOM pour un objet d'apprentissage :

```

lo_001_Category : MetadataEntry

[

    hasCategory -> catalog;

    hasValues -> "URI"

].

lo_001_Entry : MetadataEntry

[

    hasCategory -> entry;

    hasValues -> "http://www-inf.int-evry.fr/cours/BD/accueil-ext.html"

].

lo_001_Title : MetadataEntry

[

    hasCategory -> title;

```

```
    hasValues -> "Introduction aux bases de données"
].

lo_001_Language : MetadataEntry
[
    hasCategory -> language;
    hasValues -> "fr"
].
```

Annexe B : Prédicats de haut niveau

Voici un extrait des prédicats de haut niveau que nous avons utilisés pour implémenter les différentes phases de notre approche :

```
// Graph transitions
FORALL G,T
    p_g_transitions(G,T) <- G[hasTransition ->T].
// Graph vertices
FORALL G,V
    p_g_vertices(G,V) <- G.hasTransition[toVertex ->V] OR G[hasRoot ->V].
// LO vertices of a graph.
FORALL G,V
    p_g_lo(G,V) <- p_g_vertices(G,V) AND EXISTS X X:LearningObject AND unify(V,X).
// LO vertices of a graph.
FORALL G,O
    p_g_operator(G,O) <- p_g_vertices(G,O) AND EXISTS X X:Operator AND
unify(O,X).
// Number of vertices in a graph
FORALL G,V,N
    p_g_numberOfVertices(G,N) <- p_g_vertices(G,V) AND count(G,V,N).
// composed learning objects on the composition graph of a learning object
FORALL G,V
    p_g_clo(G,V) <- p_g_lo(G,V) AND EXISTS X X:ComposedLearningObject AND
unify(V,X).
// ROOT of a graph
FORALL G,R
    p_g_root(G,R) <- G.hasTransition[fromVertex -> R] AND NOT EXISTS T
(G[hasTransition ->> T] and T[toVertex ->R]).
// Number of ROOTs
FORALL G,R,N
    p_g_numberOfRoots(G,N) <- p_g_root(G,R) and count(G,R,N).
// ENDS of graph
```

```

FORALL G,V
    p_g_ends(G,V) <- G.hasTransition[toVertex -> V] AND NOT EXISTS T
(G[hasTransition ->> T] and T[fromVertex ->V]).
// Number of ALT operators in a graph
FORALL G,V,N
    p_g_numberOfALT(G,N) <- p_g_vertices(G,V) AND V:ALT AND count(G,V,N).
// Number of PAR operators in a graph
FORALL G,V,N
    p_g_numberOfPAR(G,N) <- p_g_vertices(G,V) AND V:PAR AND count(G,V,N).
// Number of LO in a graph
FORALL G,V,N
    p_g_numberOfLO(G,N) <- p_g_lo(G,V) AND count(G,V,N).
// Number of primitive LO in a graph
FORALL G,V,N
    p_g_numberOfPLO(G,N) <- p_g_lo(G,V) AND V:PrimitiveLearningObject AND
count(G,V,N).
// Number of composed LO in a graph
FORALL G,V,N
    p_g_numberOfCLO(G,N) <- p_g_CLO(G,V) AND count(G,V,N).
// List of bad ends
FORALL G,V,L
    p_g_badEnds(G,V) <- p_g_ends(G,V) and V:L and L::Operator.
// Number of bad ends
FORALL G,V,N
    p_g_numberOfBadEnds(G,N) <- p_g_badEnds(G,V) and count(G,V,N).
// List of next nodes
FORALL G,T,N,V
    p_g_nextNodes(G,N,V) <- G[hasTransition -> T] and T[fromVertex -> N] and
T[toVertex ->V].
// LO vertices
FORALL LO,V,G
    p_lo_vertices(LO,V) <- LO[hasGraph->G] AND p_g_vertices(G,V).
// LO transitions
FORALL LO,V,G
    p_lo_transitions(LO,V) <- LO[hasGraph->G] AND p_g_transitions(G,V).

```

// Predicate that associates to a learning objects its vertices that are not composed learning objects

FORALL LO,V

$p_lo_notCLO(LO,V) \leftarrow p_lo_vertices(LO,V) \text{ AND NOT EXISTS } X$
 $X:ComposedLearningObject \text{ AND } unify(X,V).$

// learning objects on the composition graph of a learning object

FORALL LO,V

$p_lo_lo(LO,V) \leftarrow p_lo_vertices(LO,V) \text{ AND } V:LearningObject.$

// composed learning objects on the composition graph of a learning object

FORALL LO,V

$p_lo_CLO(LO,V) \leftarrow p_lo_lo(LO,V) \text{ AND } V:ComposedLearningObject.$

// root of the composition graph of a composed learning object

FORALL LO,V,G

$p_lo_root(LO,V) \leftarrow LO[hasGraph-\>G] \text{ AND } p_g_root(G,V).$

// roles in content

FORALL LO,G,V,R,C

$p_lo_existsRole(LO,R) \leftarrow LO[hasGraph-\>G] \text{ AND } p_g_lo(G,V) \text{ AND EXISTS } V$
 $V[hasContent@(R)-\>C].$

// concepts in content

FORALL LO,G,V,R,C

$p_lo_concepts(LO,C) \leftarrow LO[hasGraph \ -\>G] \text{ AND } p_g_lo(G,V) \text{ AND } V[hasContent@(R)-\>>C].$

...

Annexe C : Système COLORS

COLORS (COoperative Learning Object Repository System) est une banque d'objet d'apprentissage. Ce logiciel libre est issu du code source de la banque d'objet d'apprentissage DOOR (Digital Open Object Repository). COLORS est développé au sein de l'unité de recherche UTIC (Unité de recherche en Technologies de l'Information et de la Communication) de l'ESSTT (Ecole Supérieure de Sciences et Techniques de Tunis). Elle est développée en PHP en utilisant un système de gestion de base de données MySQL.

Sa version actuelle est conforme à la spécification « IMS Content Package 1.1.3 », SCORM 1.2 et SCORM 2004. Elle permet d'importer, de retrouver et d'accéder à des objets d'apprentissage conformes à ces spécifications. Ce système est multilingue et il supporte déjà trois langues, à savoir le français, l'anglais et l'arabe.

COLORS offre la possibilité de faire coopérer plusieurs banques d'objets d'apprentissage. Toutefois, il se distingue par la facilité avec laquelle un ou plusieurs réseaux de banques d'objets d'apprentissage sont mis en place.

Voici la page d'accueil de l'administrateur :

COLORS

Accueil Recherche Explore Statistiques Préférences À propos Déconnexion

Accueil

Bonjour admin admin, tu es identifié.

Analyser un objet

Importer un objet

Gérer les paramètres

- Types d'interactivité
- Mots-clé
- Langues
- Types de ressources
- Destinataires
- Contextes d'utilisation

Colors Coordinator

Mettre à jour le profil

Accueil COLORS

Cette section permet de faire les actions suivantes:

Analyser un objet

Tu peux analyser ton objet pour comparer les aspect Langues, Taille, Types de ressources, Destinataires, Type d'objet et des Mots clé existants avec ceux des dix learning object les plus téléchargés de l'entred.

Importer un objet

Tu peux importer un learning object existant. La procédure est simple, tu dois simplement choisir le learning object et la position dans le catalogue.

Gérer les utilisateurs

Tu peux ajouter, modifier et éliminer des utilisateurs. Chaque utilisateur est pourvu des informations suivantes: nom d'utilisateur, mot de passe, nom, prénom, institut, adresse électronique, type (administrateur, auteur ou utilisateur normal) et état (actif ou non actif).

Gérer les paramètres

Tu peux ajouter, modifier et éliminer les paramètres qui seront utilisés durant la procédure de création des learning objects. Il y a cinq paramètres différents: mots-clé, langues, types de ressource, destinataire et contexte d'utilisation. Les paramètres correspondent à la définition dans les spécifications des metadonnées en IMS.

Colors Coordinator

Tu peux ajouter, modifier et éliminer des réseaux Colors Coordinators. Pour chaque réseau, il faut créer un compte utilisateur en ajoutant les informations suivantes: login, mot de passe. Il faut aussi donner l'adresse du réseau pour créer ton compte.

Mettre à jour le profil

Tu peux modifier ton profil en mettant à jour le mot de passe, nom, prénom, institut, et adresse électronique.

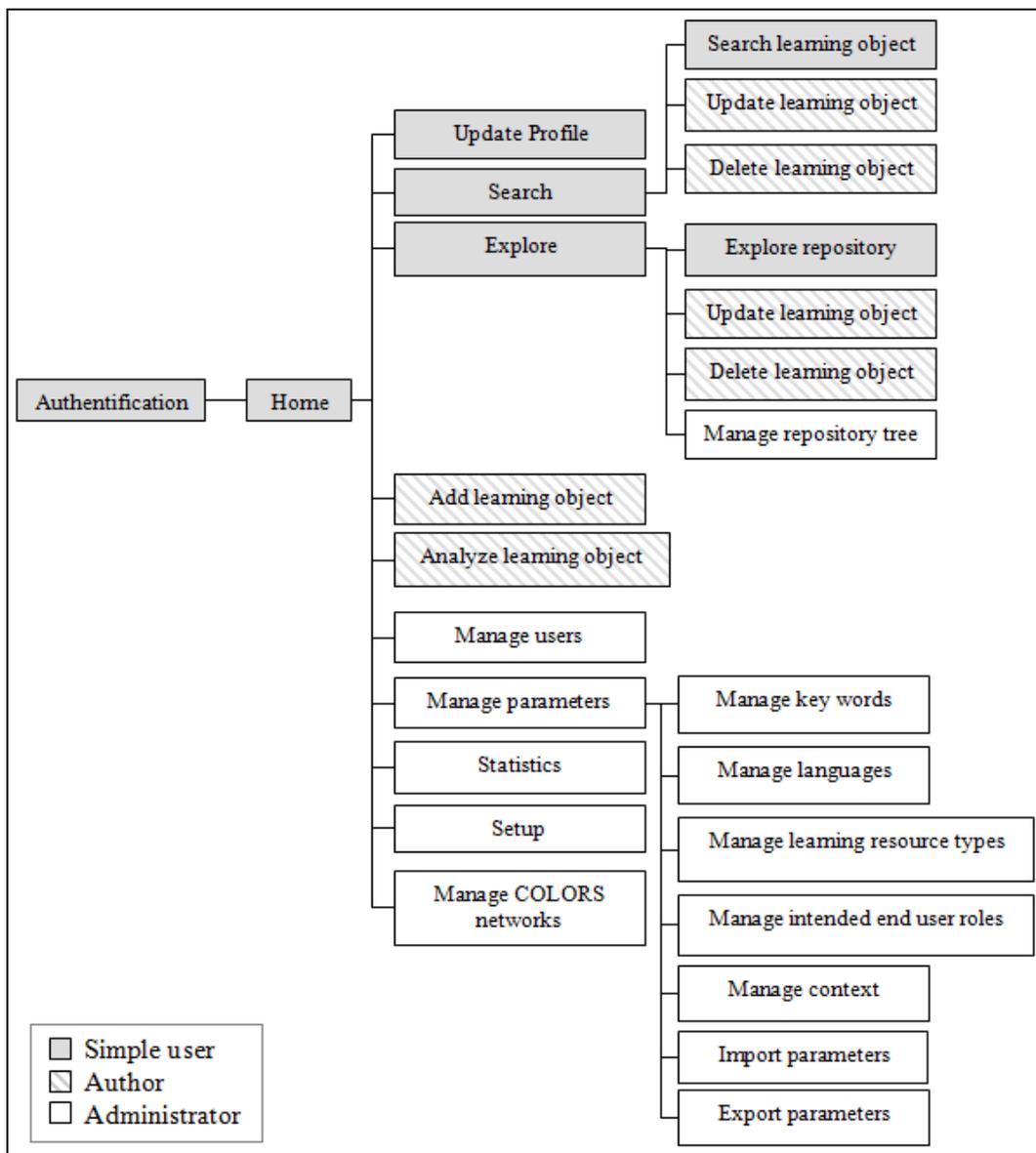
Développé par UTIC

Au niveau de la page d'accueil, nous trouvons une explication pour chaque fonctionnalité offerte par le système. Cette page se différencie pour chaque type d'utilisateur. Cela est dû

aux privilèges associés à chacun. Dans ce qui suit, nous allons expliquer ces différentes fonctionnalités.

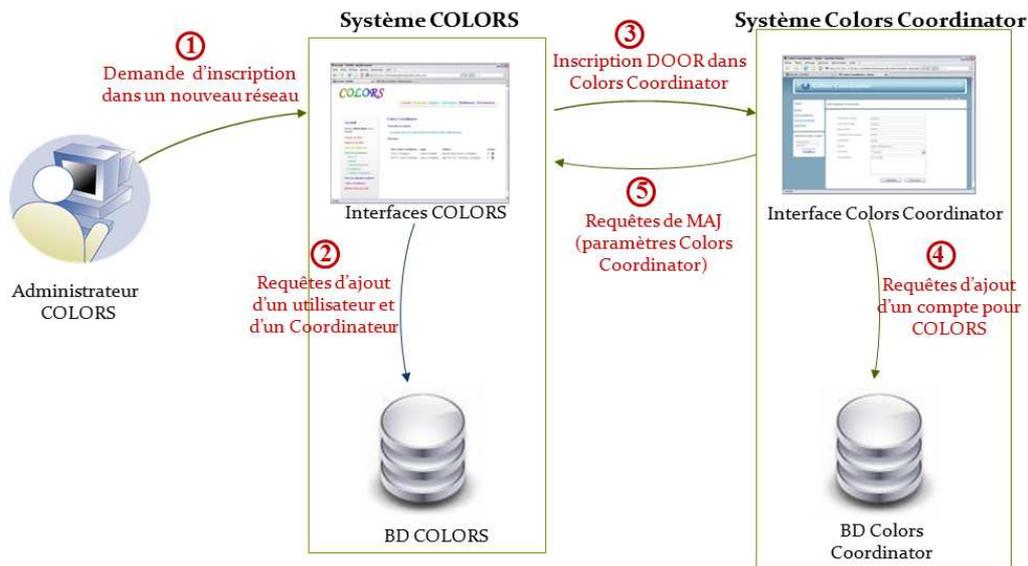
Fonctionnalités de COLORS

Nous pouvons classifier les fonctionnalités de COLORS suivant le type de l'utilisateur (administrateur, auteur, utilisateurs simple) comme l'indique la figure ci-dessous :



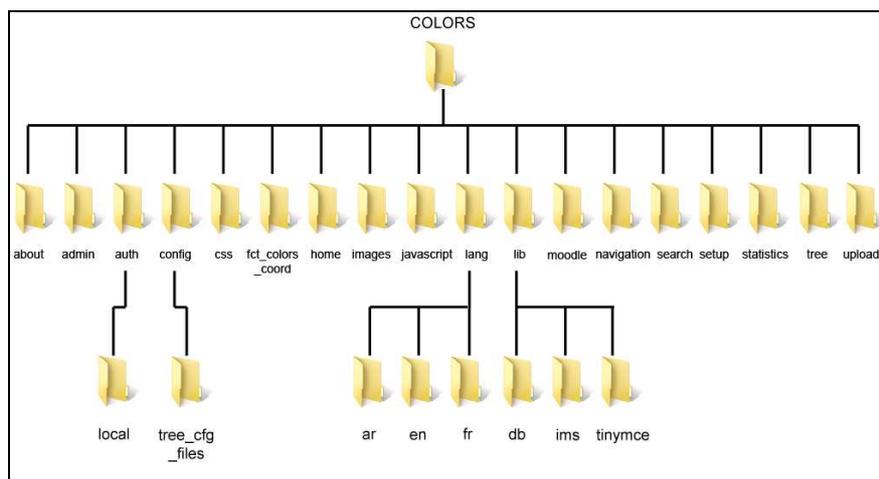
Un utilisateur simple (un LMS ou un apprenant) peut utiliser le système pour la recherche des objets d'apprentissage se trouvant dans la banque ou dans un ou plusieurs réseaux auxquels est intégrée cette banque, explorer le dépôt local et mettre à jour son profil. Cependant les auteurs ont en plus des fonctionnalités des utilisateurs simples la possibilité d'ajouter des objets d'apprentissage au système, de les analyser, de les mettre à jour et de les supprimer.

L'administrateur a en plus des fonctionnalités d'un auteur la possibilité de gérer les utilisateurs et les paramètres de la banque d'objets d'apprentissage, de consulter les statistiques d'utilisation, de modifier la configuration de l'installation et de gérer les inscriptions aux réseaux COLORS.



L'architecture des dossiers de COLORS

Les fichiers de COLORS sont organisés et classés dans des répertoires formant une architecture cohérente et significative. Cet aspect est très important pour faciliter la maintenance et l'amélioration du code source. En effet, les fichiers source sont classifiés dans des répertoires selon leurs fonctionnalités. Ainsi, il est plus facile de repérer le code à modifier ou à améliorer.



Voici une description détaillée du contenu des répertoires :

- « about » contient les fichiers nécessaires pour l’affichage des informations sur le système, telles que la version, la licence, le langage du développement, les éditeurs, etc.
- « admin » contient les pages disponibles aux utilisateurs de la catégorie ‘administrateur’.
- « auth » contient les systèmes d’authentification supportés par COLORS.
- « config » contient les fichiers de configuration.
- « css » comprend les fichiers de style.
- « home » comporte les pages liées à la section ‘home’.
- « images » contient les images utilisées dans le site.
- « javascript » contient les fichiers JavaScript.
- « lang » contient les fichiers de langues.
- « lib » contient les bibliothèques qui gèrent les erreurs, la communication avec la base de données et la gestion des fichiers importés.
- « moodle » contient les fichiers pour l’intégration avec Moodle.
- « navigation » contient les scripts définissant l’entête et le pied des pages.
- « search » comporte les pages liées à la section de recherche.
- « setup » contient les scripts d’installation et le script SQL de la base de données.
- « statistics » comprend les fichiers liés à la section statistiques.
- « tree » contient les scripts pour la fonctionnalité ‘Explore’
- « uploads » comporte les fichiers contenant les objets ajoutés par les auteurs et l’administrateur.

Module Analyse de métadonnées

L’analyse a pour objectif de positionner l’objet en question par rapport aux dix objets les plus téléchargés.

La raison de l’analyse est de donner à l’auteur une vue de son objet suivant les critères de réutilisabilité, simplicité et densité pédagogique. En effet parmi les métadonnées que nous avons prises en compte dans notre base, nous avons opté pour la langue, la taille, le format, le type des ressources contenu dans l’objet et le public cible.

Pour chaque métadonnée nous attribuons un degré de compatibilité avec ce qu’il y a dans la table des meilleurs objets. Si sa valeur correspond à celle du meilleur objet nous l’affichons avec une couleur verte, si elle est parmi les meilleurs objets nous l’affichons avec une couleur orangé et si elle est inexistante nous l’affichons avec une couleur rouge.

Accueil

Bonjour **admin admin**, tu es identifié.

[Analyser un objet](#)

[Importer un objet](#)

[Gérer les utilisateurs](#)

[Gérer les paramètres](#)

- Types d'interactivité

- Mots-clé

- Langues

- Types de ressources

- Destinataires

- Contextes d'utilisation

[Colors Coordinator](#)

[Mettre à jour le profil](#)

Résultat d'analyse

L'analyse effectuée a permis d'extraire les métadonnées suivantes et leurs situations par rapport aux dix meilleurs objets du system, ainsi que le diagramme associé pour chacune, qui peuvent être visualisés en cliquant sur l'icône correspondante au niveau de la colonne détail.

Critère	Valeurs	Détail
Langues	fr	
Taille	309.79 Ko	
Types de ressource	Experiment	
Type d'objet	IMS	
Mots-clé	avec 1 Mots-clé	

Légende :

- Correspond à la meilleure valeur par rapport au 10 meilleurs objets
- Correspond à valeur parmi celles des 10 meilleurs objets
- valeur manquantes ou n'est pas parmi celles des 10 meilleurs objets

Développé par UTIC

En effet, nous affichons pour l'auteur ces métadonnées importantes avec ces couleurs significatives en lui donnant la possibilité de visualiser un diagramme qui montre les statistiques des métadonnées des dix meilleurs objets

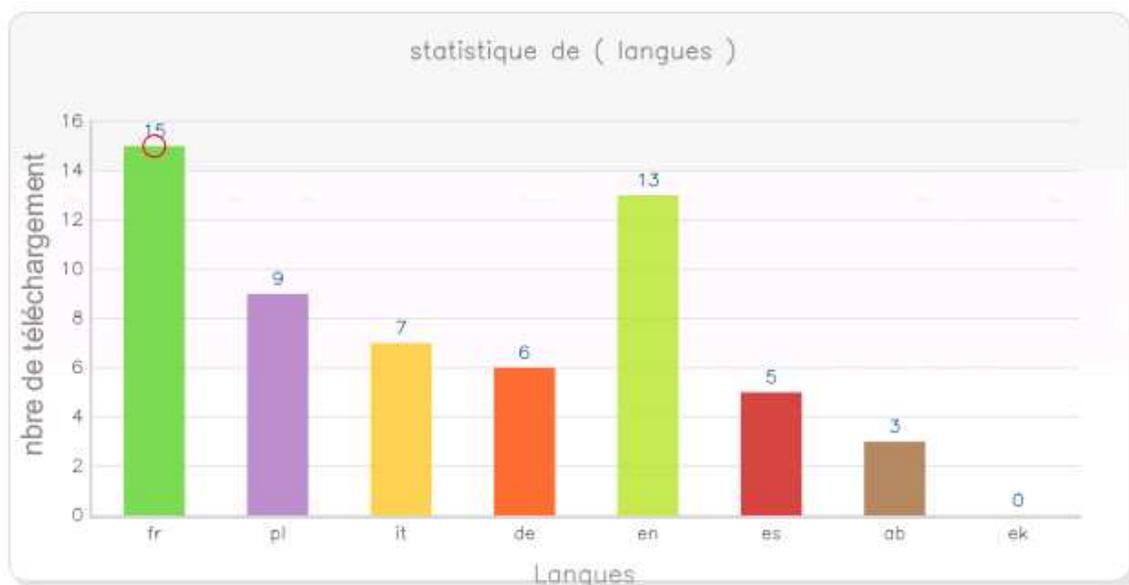
Afin de mieux comprendre la raison pour laquelle ces métadonnées ont pris une telle interprétation, nous avons préparé pour l'auteur un diagramme explicatif pour chaque métadonnée qui montre l'aspect de cette dernière par rapport aux meilleurs dix objets existants.

Pour ce faire, nous avons eu recours à la technologie « AJAX » qui assure une création dynamique de chaque diagramme. Ainsi, le nombre de téléchargements est détecté automatiquement, et donc les valeurs présentées au niveau de l'axe des ordonnées s'organisent suivant la valeur maximale existante. En outre, les métadonnées prises en compte et présentées sur l'axe des abscisses sont extraites des valeurs adoptées des dix meilleurs objets. Comme le nombre de téléchargements varie, ces métadonnées sont variables du point de vue nombre et valeur, ce qui assure une présentation proportionnelle et spécifique pour chaque diagramme et pour chaque état de la base.

En effet, nous avons voulu montrer à l'auteur à quel niveau se situe son objet au sein du diagramme, donc nous avons choisi de dessiner un petit cercle rouge en haut de la barre correspondante pour le distinguer des autres.

En plus, pour une raison esthétique, nous avons coloré les barres présentées dans les diagrammes par différentes couleurs prises pour aller en avant avec le thème coloré utilisé par le système COLORS.

Comme nous l'avons expliqué, ce diagramme se génère automatiquement lors de l'analyse. Ses axes sont les valeurs de la métadonnée en question en abscisses et le nombre de téléchargements de cette dernière en ordonnées.



Cette figure montre les diverses langues détectées dans les dix meilleurs objets telles que le français (fr), l'anglais (en), etc. Le nombre de téléchargements de chaque langue est indiqué sur sa barre associée dans le diagramme, et suivant ces nombres se présentent les valeurs indiquées sur l'axe des ordonnées. Dans cet exemple la langue adoptée par l'objet d'apprentissage analysé est bien le français, ce qui est présenté par le cercle rouge tout en haut de la barre associé à la langue française.