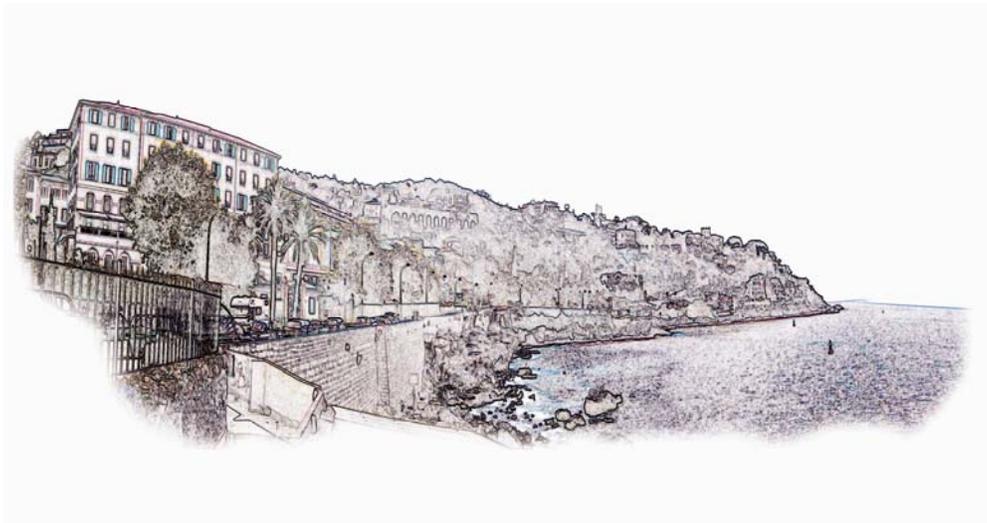


Plate-forme AFIA / Nice, du 30 mai au 3 juin 2005

JOURNEE THEMATIQUE:

Web sémantique pour le e-Learning



Rose Dieng-Kuntz
Monique Grandbastien
Danièle Hérin

Journée Thématique "Web Sémantique pour le E-Learning"

La Journée thématique « Web sémantique pour le E-learning » journée finale de l'Action Spécifique du CNRS & COLORS WebLearn, se tiendra à Nice le 31 mai dans le cadre de la plate-forme AFIA'2005.

Présentation

On trouve de plus en plus sur le Web toutes sortes de ressources pédagogiques : transparents (dans des formats tels que PowerPoint), cours électroniques (correspondant aux «polycopiés»), bibliographie de documents auxquels ces cours font référence, exercices, FAQs, ou toute autre ressource sur le Web qui présente un aspect pédagogique. Selon les cas, ces ressources sont disponibles publiquement ou réservées à une communauté (entreprise, organisme de formation, académie, etc.).

Cependant, il est actuellement impossible d'accéder de façon rationnelle, et avec de bonnes chances de trouver les documents pertinents, à cette masse grandissante de ressources. Il est donc nécessaire d'étudier les techniques qui permettent d'indexer, d'analyser, de composer, de mémoriser et de rechercher de tels documents.

Le Web sémantique vise à rendre les ressources du Web actuel non seulement compréhensibles par les humains (comme c'est le cas pour le Web actuel) mais aussi interprétables par des machines. Parmi les travaux visant à la construction du Web sémantique, l'approche la plus fréquente consiste à décrire ces ressources à l'aide de méta-données, ou annotations sémantiques, en utilisant le vocabulaire conceptuel fourni par une ontologie. De façon générale, ces annotations sémantiques peuvent être externes ou internes à la ressource, objectives ou subjectives, individuelles ou collectives.

L'étude des applications possibles des technologies émergentes du web sémantique et des web services pour le e-Learning constituait l'objet de l'action spécifique du CNRS & COLORS WebLearn, financée par le CNRS et par l'INRIA Sophia Antipolis. Dans ce cadre, nous avons étudié plusieurs scénarios d'e-learning selon le contexte (académique vs industriel), les acteurs concernés (enseignants vs apprenants), objectifs de ces acteurs (mise à disposition de ressources éducatives, annotation de ces ressources, composition de cours, interaction entre enseignants et/ou apprenants dans le cadre d'une formation à distance.

Les travaux du groupe WebLearn étaient organisés dans les workpackages suivants :

- Modèles de e-learning et personnalisation (**Liupp**, Crego, Utc, Enstb, Aida, Lirmm, Clips).
- Ontologies pour le Web sémantique et le e-Learning (**Acacia**, UTC)
- Annotations et métadonnées (**Clips**, Crego, Lirmm, I3S, Enstb, Acacia)
- Découpage - Granularité : Composition - Ingénierie pédagogique (I3S, Lirmm)
- Web services (**Lirmm**, ENSTB, Acacia, I3S)
- Langages du Web Sémantique (**UTC**, Acacia, Lirmm)
- Outils dédiés au Web sémantique ou au e-Learning (**Acacia**, I3S, Clips, Lirmm)

Dans les actes de la journée thématique seront publiés les chapitres correspondant à trois worpackages :

- Ontologies pour le Web sémantique et le e-Learning
- Langages du Web Sémantique
- Outils dédiés au Web sémantique ou au e-Learning

Ils seront ultérieurement complétés par les chapitres des autres workpackages dans le rapport final de l'AS-COLORS WebLearn.

Il nous a semblé intéressant d'accueillir également des présentations extérieures au groupe, (correspondant aux soumissions acceptées après un appel à communications). Voici quelques thèmes sur lesquels nous souhaitons discuter lors de la journée et montrer les contributions tant du groupe WebLearn que des membres extérieurs :

- Scénarios de e-learning & Web sémantique
- Modèles du e-learning & Web sémantique
- Ontologies dédiées au e-learning
- Langages du Web sémantique et e-Learning
- Recherche d'information & e-Learning
- Web sémantique pour la coopération entre acteurs du e-Learning
- Annotations sémantiques sur des ressources pédagogiques
- Composition de ressources pédagogiques
- Hétérogénéité des ressources pédagogiques & Web sémantique
- Evolution des ressources pédagogiques & Web sémantique
- Aspects cognitifs & exploitation du Web sémantique pour le e-Learning
- Aspects sociologiques & exploitation du Web sémantique pour le e-Learning
- Analyse d'usages dans le contexte du e-Learning & Web sémantique
- Retour d'expériences d'applications du Web sémantique au e-Learning

Comité d'organisation :

Rose Dieng-Kuntz, ACACIA, INRIA Sophia Antipolis
Monique Grandbastien, LORIA, Université Henri Poincaré
Danièle Hérim, LIRMM, Université Montpellier II

Comité de programme

Marie-Hélène Abel , HEUDIASYC, UTC, Compiègne
Marie-Noëlle Bessagnet, LIUPPA, Université de Pau et des Pays de l'Adour
Olivier Corby , ACACIA, INRIA Sophia Antipolis
Cyrille Desmoulins, IMAG-CLIPS, Université Joseph Fourier & CNRS & INPG, Grenoble
Rose Dieng-Kuntz , ACACIA, INRIA Sophia Antipolis
Bernard Fallery, CREGO, Université Montpellier II
Catherine Faron-Zucker , I3S, Université de Nice Sophia Antipolis
Fabien Gandon, ACACIA, INRIA Sophia Antipolis
Serge Garlatti, LIASC, ENST Bretagne
Alain Giboin, ACACIA, INRIA Sophia Antipolis
Monique Grandbastien, LORIA, université Henri Poincaré
Danièle Hérim, LIRMM, Université Montpellier II & CNRS
Emmanuel Houzé, CREGO, Université Montpellier II
Sébastien Iskal, LIUM, Université du Maine, Le Mans
Dominique Lenne , HEUDIASYC, UTC, Compiègne
Claude Moulin, HEUDIASYC, UTC, Compiègne
Pierre Pompidor, LIRMM, Université Montpellier II & CNRS
Michel Sala, LIRMM, Université Montpellier II & CNRS

Remerciements

Nous remercions tous les membres du groupe WebLearn pour leur dynamisme et les discussions enrichissantes menées au sein de ce groupe, ainsi que les membres du comité de programme pour leurs relectures attentives et constructives des papiers soumis. Nous remercions vivement Fabien Gandon pour sa remarquable organisation de la plate-forme AFIA'2005, Monique Simonetti pour son professionnalisme exemplaire et Patricia Maleyran pour son aide logistique très utile pour la gestion des articles soumis et des participants. Enfin, nous sommes très reconnaissantes au CNRS et à l'INRIA Sophia Antipolis pour avoir financé cette collaboration pluridisciplinaire.

Sophia Antipolis, le 12 mai 2005
Rose Dieng-Kuntz, Monique Grandbastien, Danièle Hérim
Coordinatrices de l'AS-COLORS WebLearn

TABLE DES MATIERES

PREMIERE PARTIE

Vers une approche Web Sémantique dans la conception d'un système d'apprentissage. Revue du projet TRIAL SOLUTION	1
<i>Michel Buffa, Sylvain Dehors, Catherine Faron-Zucker, Peter Sander</i>	
Des Annotations Sémantiques pour Apprendre. QBLS : Modélisation et Expérimentation	15
<i>Sylvain Dehors, Catherine Faron-Zucker, Jean Paul Stromboni, Alain Giboin</i>	
Approche multidimensionnelle du texte pour le balisage des ressources pédagogique	31
<i>Isabelle Rondeau, Sylvie Ranwez, Michel Crampes</i>	
Un outil pour la qualification de ressources à base d'ontologies (<i>papier court</i>)	39
<i>Baruk Tolédano, Monique Baron, Hélène Giroire</i>	

DEUXIEME PARTIE : Les travaux du groupe WebLearn

Ontologies pour le Web Sémantique et pour le E-Learning	44
<i>Marie-Hélène Abel, Thanh-Le Bach, Sylvain Dehors, Rose Dieng-Kuntz, Fabien Gandon, Phuc-Hiep Luong, Claude Moulin</i>	
Langages pour le Web Sémantique et pour le E-Learning	97
<i>Marie-Hélène Abel, Rose Dieng-Kuntz, Danièle Hérin, Dominique Lenne, Claude Moulin, Pierre Pompidor, Anfana Traoré</i>	
Les Outils du Web Sémantique et du E-Learning	123
<i>Faiçal Azouaou, Tuan-Dung Cao, Sylvain Dehors, Cyrille Desmoulins, Rose Dieng-Kuntz, Catherine Faron-Zucker</i>	

Vers une approche Web Sémantique dans la conception d'un système d'apprentissage. Revue du projet TRIAL SOLUTION

Michel Buffa¹, Sylvain Dehors², Catherine Faron-Zucker¹, Peter Sander¹

¹ MAINLINE, I3S, UNSA
930 route des Colles, bât ESSI, BP 145, 06930 Sophia Antipolis cedex
{buffa, faron, sander}@essi.fr

² ACACIA, INRIA
2004 route des Lucioles, BP 93, 06902 Sophia Antipolis cedex
sdehors@sophia.inria.fr

Résumé : Dans le cadre du projet européen TRIAL SOLUTION, nous avons travaillé sur la publication de documents électroniques personnalisés à partir de livres scientifiques existants. L'approche générale adoptée dans ce projet a consisté à découper des livres électroniques en ressources pédagogiques élémentaires et à annoter celles-ci avec des méta-données décrivant leurs contenus sémantiques, leurs caractéristiques pédagogiques et leurs interrelations. Ces ressources annotées sont disponibles sur un serveur dédié à la production, par des étudiants et des enseignants, de matériel d'enseignement ou d'apprentissage personnalisé, à l'aide d'outils dédiés. Les annotations sont le support d'une recherche *sémantique* des ressources pédagogiques lors la production d'un document personnalisé. Nous décrivons dans cet article l'ensemble du projet et nous nous attachons plus particulièrement à la construction de la mémoire des ressources pédagogiques annotées, pour laquelle nous avons développé un outil d'annotation. Nous dressons ensuite un bilan de ce projet dans la perspective du Web sémantique, qui n'était pas présente lors de la définition de ce projet.

Mots-clés : Web sémantique, Système d'apprentissage, Entrepôt de ressources pédagogiques (LOR), Annotation de ressources pédagogiques, Extraction de ressources à partir de texte.

1 Introduction

TRIAL SOLUTION est un mot-valise pour Tools for Reusable, Integrated, Adaptable Learning - Systems/Standards for Open Learning Using Tested, Interoperable Objects and Networking. Ce projet européen s'est intéressé à la publication de documents personnalisés à partir de livres scientifiques existants (essentiellement dans le domaine des mathématiques).

La conception de nouveaux documents électroniques « from scratch » est très coûteuse et, pour cette tâche spécialisée, les auteurs de livres imprimés possèdent rarement les compétences nécessaires. D'autre part, la plupart des livres imprimés ont maintenant des formats digitaux, e.g. en Microsoft Word ou en Latex, très répandu dans la communauté des mathématiques ; ces formats peuvent faciliter l'automatisation de la conception de documents électroniques à partir de livres imprimés. Les bénéfices de cette approche basée sur le re-engineering de matériel existant a été décrit dans (Dahn, 2001) et le processus lui-même détaillé dans (Dahn *et al.*, 2001). Ces considérations ont présidé à la conception du projet TRIAL SOLUTION.

L'approche générale consiste à découper des livres électroniques en ressources pédagogiques élémentaires puis à raffiner ce découpage et à annoter les ressources avec des méta-données sur leurs contenus sémantiques, sur leurs caractéristiques pédagogiques et leurs interrelations, dans le but de permettre une recherche « intelligente ». L'ensemble des ressources annotées est disponible sur un serveur Web, dédié aux enseignants et aux étudiants. Des outils de recherche de ressources et de publication de documents ont été développés pour permettre à ces utilisateurs finaux de rechercher sur le serveur le matériel pédagogique qui les intéresse et de produire des documents personnalisés. Notre contribution, l'une des pierres angulaires du projet, est l'outil d'annotation des ressources pédagogiques, qui permet le raffinement de leur découpage et leur annotation.

Dans la section suivante, nous décrivons les processus d'exploitation des livres et d'extraction de ressources pédagogiques ; nous présentons les différents outils qu'intègre la plate-forme TRIAL SOLUTION, en détaillant plus particulièrement l'outil d'annotation que nous avons développé. Dans la section 2 nous dressons ensuite un bilan du projet, dans la perspective du Web sémantique ; nous mettons en évidence la similarité de l'approche adoptée dans le projet avec celles du Web sémantique qui visent à annoter automatiquement des ressources pédagogiques à partir d'une ontologie. Nous concluons en décrivant notre travail actuel de conception d'un système d'apprentissage qui tire partie de notre expérience du projet TRIAL SOLUTION et des technologies du Web sémantique.

2 Présentation du projet TRIAL SOLUTION

La plateforme TRIAL SOLUTION intègre trois principaux services : l'extraction et l'annotation automatiques de ressources pédagogiques à partir de livres électroniques, le re-engineering des ressources pédagogiques extraites et la recherche dans l'entrepôt de ressources ainsi constitué, basée sur les annotations sémantiques des ressources. La figure 1 ci après présente une vue générale du processus.

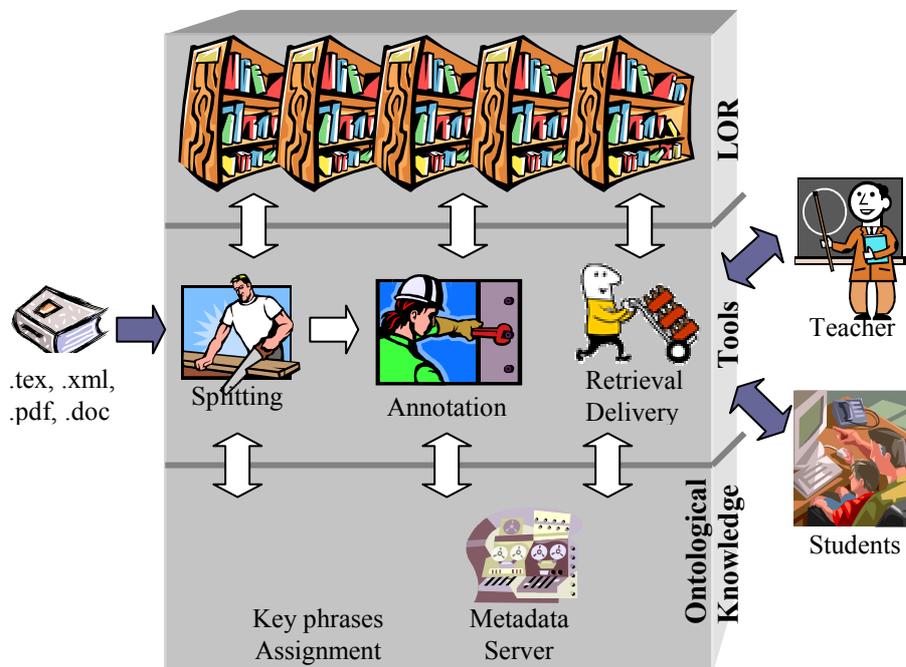


Fig. 1 – La plateforme TRIAL SOLUTION

2.1 Extraction de ressources pédagogiques à partir de textes

La première étape du processus consiste à désagréger automatiquement un document textuel en un ensemble de ressources, à l'aide d'un outil dédié, développé par la société Slicing Information Technology (SIT)¹. Cet outil permet de traiter des documents dans des formats structurés tels que Word ou LaTeX. Il extrait de tels documents la structure hiérarchique sous la forme d'un arbre XML, dans lequel chaque noeud correspond à une ressource (une « tranche » découpée dans le document). La DTD qu'il utilise est détaillée sur le site du projet (TRIAL SOLUTION, 2001). Les sections, chapitres, tableaux, figures, exemples sont parmi les ressources les plus faciles à identifier automatiquement par cet outil et plus un document utilise de styles standards, meilleur sera son découpage. Le résultat de ce découpage est un arbre de ressources pédagogiques stocké dans un répertoire de fichiers ; ce répertoire est logé par un serveur Internet accessible par le protocole HTTP.

Cet outil annote automatiquement les ressources pédagogiques extraites avec des méta-données sur le contenu sémantique des ressources, leurs interrelations dans le livre d'origine et l'auteur de celui-ci. L'identification des relations entre les ressources est basée sur les hyperliens, les occurrences de « voir aussi », « cf. », etc. présents

¹ Infotech Slicing Technology GmbH, <http://www.slicing-infotech.de/de/index.php>

dans le document original. Les mots-clés, spécifiés parfois par l'auteur du document, aident à la classification du contenu des ressources, e.g. si le terme « sinus » apparaît dans le contenu d'une ressource, celle-ci est identifiée comme traitant de trigonométrie. A partir d'un thésaurus construit pour le domaine dont traite le document, il s'agit de rechercher dans celui-ci quels mots ou phrases clés du thésaurus apparaissent. Deux outils ont été conçus à cet effet : un outil de gestion du thésaurus, appelé serveur de meta-données, et un outil d'attribution de phrases clés qui gère une collection de phrases clés du domaine des mathématiques issues d'un manuel standard de mathématique (Bronstein et al., 2000).

Cependant, ce processus automatique est presque toujours insuffisant pour annoter les ressources pédagogiques extraites, voire susceptible de produire des annotations incorrectes, comme cela est décrit dans (Dahn *et al.*, 2001). Les ressources pédagogiques extraites et leurs annotations sont affinées à l'aide d'un outil d'annotation.

2.2 Annotation de ressources pédagogiques

Nous avons développé l'outil d'annotation de la plateforme TRIAL SOLUTION qui permet à des experts du domaine traité de corriger et d'améliorer la base de ressources pédagogiques annotées et interconnectées produite automatiquement. Des expériences de mise en valeur d'un livre à l'aide de cet outil sont décrites dans (DAHN *et al.*, 2001). Les utilisateurs de cet outil doivent avoir une bonne connaissance du contenu du livre original ; il peut s'agir des auteurs eux-mêmes ou plus souvent d'enseignants utilisant ce livre comme matériel pédagogique.

Notre outil d'annotation est un client du serveur de ressources pédagogiques ; il communique avec celui-ci à l'aide d'un protocole où à la fois les requêtes (et leurs paramètres) et les valeurs retournées par le serveur sont basés sur un encodage XML défini dans la DTD de TRIAL SOLUTION.

Lorsqu'il est connecté au serveur de ressources pédagogiques extraites automatiquement à partir de livres électroniques, cet outil permet (1) d'attribuer aux ressources un titre ; (2) d'éditer leur contenu ; (3) d'éditer la structure arborescente de l'ensemble de l'entrepôt des ressources : les ressources peuvent être réorganisées, découpées, fusionnées, détruites et de nouvelles ressources peuvent être créées ; et (4) d'éditer les méta-données associées aux ressources. Trois types de méta-données sont distingués. Les « types » représentent le rôle pédagogique du contenu des ressources : définition, théorème, explication, etc. Les « mots-clés » représentent les sujets abordés dans le contenu des ressources. Les « relations » avec d'autres ressources constituent un réseau sémantique de ressources : référence, prérequis, etc.

Notre outil d'annotation permet l'édition non seulement des méta-données associées aux ressources pédagogiques mais aussi du thésaurus sur lequel reposent ces annotations : l'édition des termes du thésaurus, de la description associée à un terme, des relations hiérarchiques entre les termes, des relations de synonymie, de parenté.

La figure 2 ci-après présente l'interface de l'outil d'annotation pour l'édition des mots-clés. Une ressource intitulée « Zinsen und Zinsperiode » (intérêts et période d'intérêt) est sélectionnée ; elle a précédemment été automatiquement annotée avec

Web sémantique pour le E-learning

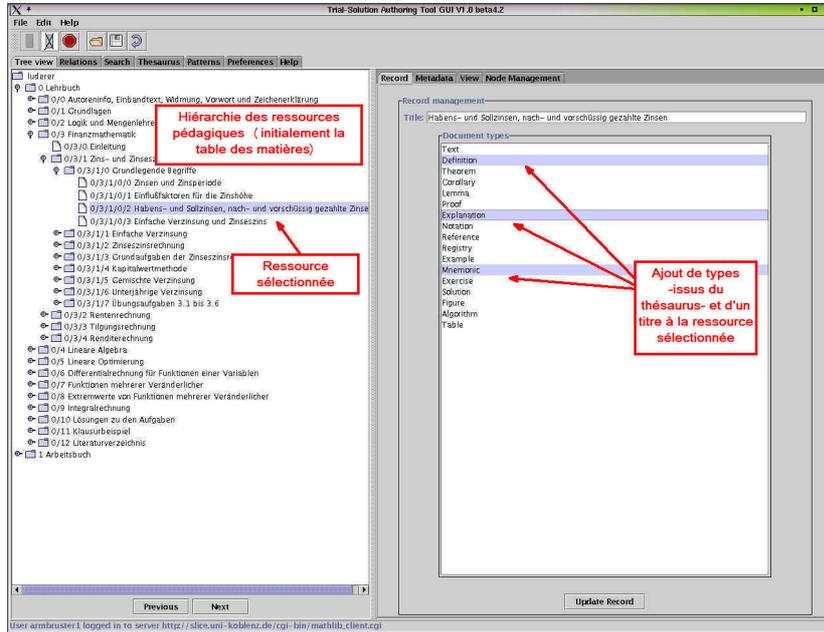


Fig. 3 – Edition de méta-données : les types pédagogiques

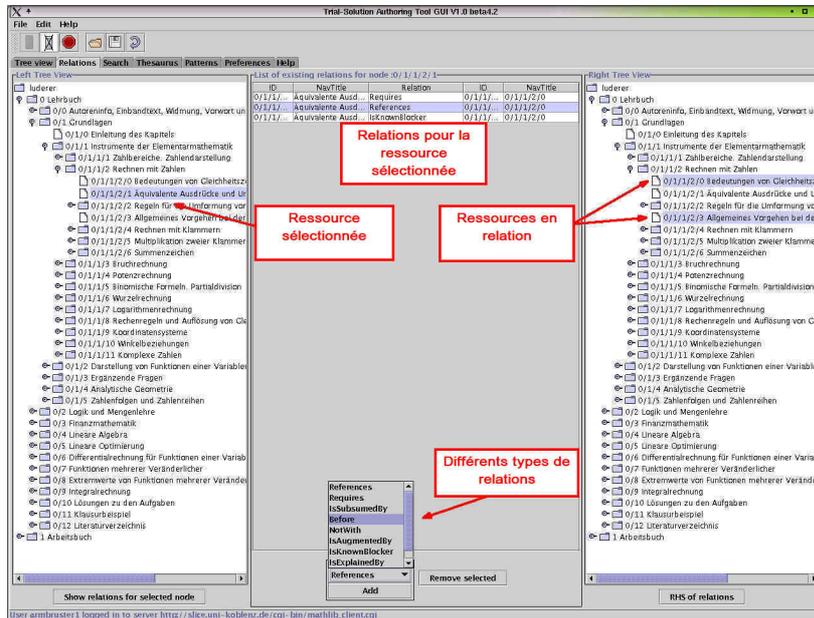


Fig. 4 – Edition de méta-données : les relations entre ressources pédagogiques

trois mots-clés : « Finanzmathematik » (mathématique financière), « Zinsen » et « Zinsperiode ». Un quatrième mot-clé lui est ici ajouté : « Nachschüssige Zinsen » (intérêts cumulés) ; ce mot-clé est sélectionné parmi une liste fournie par l’outil de gestion du thesaurus avec lequel communique l’outil d’annotation.

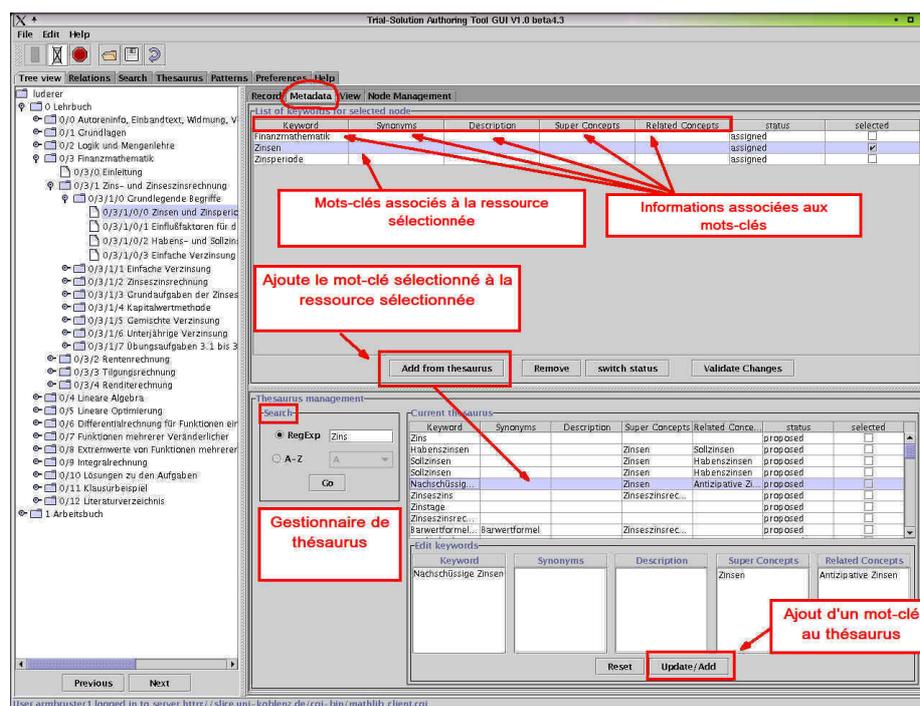


Fig. 2 – Edition de méta-données : les mots-clés du domaine

La figure 3 ci-après présente l’interface de l’outil d’annotation pour l’édition des types des ressources pédagogiques – types relatifs à leurs rôles pédagogiques. Sur cette figure, la ressource sélectionnée est annotée par les types « Definition », « Explanation » et « Mnemonic » sélectionnés dans une liste prédéfinie.

Les relations structurelles, les types et les mots-clés sont pour partie automatiquement extraits du document d’origine. En revanche les relations sémantiques entre les ressources sont plus délicates à extraire ; la plupart du temps elles sont acquises manuellement en utilisant l’outil d’annotation qui permet l’édition de relations, à choisir parmi une liste de relations prédéfinies, comme le montre la figure 4 ci-après.

La figure 5 ci-après montre un extrait du fichier XML dans lequel figurent les méta-données associées aux ressources. Dans cet extrait, la ressource intitulée « Chapter 2 » (1) est annotée par le type « Chapter » (2), par une relation de « Reference » deux autres ressources (3) et par deux mots-clés (4).

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE manifest SYSTEM "TS_IMS_CONTENT.dtd">
<manifest identifier="TSBook">
...
  <title> Complete Book</title>
  <item identifier="TSBook_all_i"
        identifierref="TSBook_r">
    <title>Title of Book</title>
...
  <item identifier="TSBook_all_i_2"
        identifierref="TSBook_r_2">
(1)   <title> Chapter 2</title>
      <metadata>
        <record>
          <sourcereference>2</sourcereference>
          <navtitle>Title of Chapter 2</navtitle>
(2)   <types>
        <type>Chapter</type>
      </types>
      <start/>
      <main href="slib_2_main.tex"/>
      <end/>
(3)   <relation kind="References">
        <target href="TSBook_i_1_1"/>
        <target href="TSBook_i_1_2"/>
      </relation>
      <relation kind="IsAugmentedBy">
        <external href="http://www.trial-
solution.de">The Trial-Solution Project</external>
      </relation>
(4)   <keyphs>
        <keyph
          href= "slib_keywords_keys.db_i_260_s"
          inherits="down" author="ts_s"
          status="proposed"/>
        <keyph
          href="slib_keywords_keys.db_s_41861_s"/>
      </keyphs>
      </record>
    </metadata>
  </item> ... </item>
</manifest>
```

Fig. 5 – Les méta-données associées à la ressource intitulée « Chapter 2 »

2.3 Recherche de ressources pédagogiques

Une fois la version électronique d'un livre découpée et annotée, elle est disponible sous la forme d'un entrepôt de ressources pédagogiques annotées ; les utilisateurs finaux peuvent alors assembler sous la forme de documents personnalisés des ressources recherchées sur la base des méta-données qui leur sont associées. L'outil de recherche est une application Web destinée à des étudiants et des enseignants dans un environnement d'éducation.

Un utilisateur peut demander la recherche des ressources pédagogiques indexées par des mots-clés particuliers, e.g. « trigonométrie ». Il peut raffiner sa recherche en spécifiant le type des ressources à rechercher : définition, théorème, démonstration, etc. Le matériel « sur mesure » qui résulte d'une recherche peut être délivré en HTML ou PDF ; toutes les sections, figures et pages sont numérotées de sorte que le document final est autonome et cohérent.

La figure 6 ci-après présente l'interface de recherche de la plateforme TRIAL SOLUTION. Un utilisateur recherche les ressources de tous les types qui traitent d'« alphabet ». Le serveur propose deux mots-clés : « alphabet » et « alphabet d'un langage ». L'utilisateur choisit le second et demande également toutes les ressources qui sont des pré-requis aux ressources annotées par « alphabet d'un langage ». Il sauvegarde le document résultat de sa recherche dans un répertoire personnel.

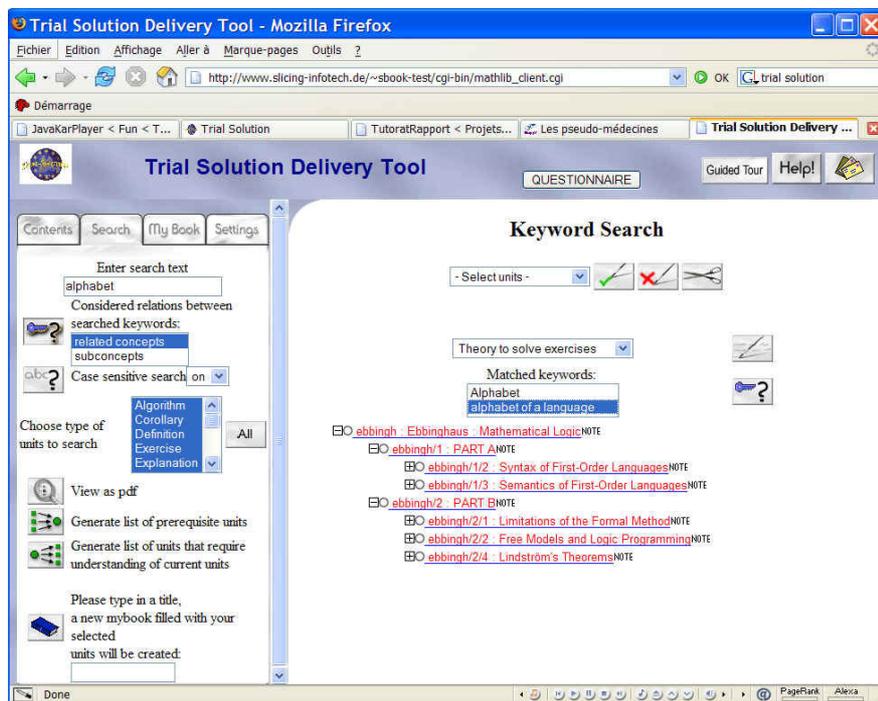


Fig. 6 – Recherche de ressources pédagogiques basée sur les méta-données

A l'issue de TRIAL SOLUTION, différents serveurs développés dans le cadre du projet sont maintenant en ligne, disponibles gratuitement sur le site Web du projet. D'autre part, des livres sont exploités commercialement par l'entreprise SIT.

3 Bilan du projet dans la perspective du Web Sémantique

Nous dressons dans cette partie un bilan du projet Trial Solution dans la perspective du Web sémantique. Celle-ci ne présidait pas à la définition du projet, mais la description du projet met en évidence une démarche et des objectifs comparables.

3.1 Comparaison de TRIAL SOLUTION avec d'autres travaux

Une approche similaire de découpage automatique de document électronique a été mise en œuvre dans le projet SeLeNe (Rigaux *et al.*, 2003). Les documents traités respectent la DTD DocBook qui est utilisée pour leur découpage. Dans le projet IMAT (Desmoulins & Grandbastien, 2002) les documents sont issus de manuels techniques exportés en XML, puis découpés selon leur structure. Ils sont indexés selon leur contenu en identifiant la présence de termes issus d'une ontologie du domaine. L'indexation de l'aspect pédagogique est effectuée manuellement de même que la composition des documents finaux.

Dans les domaines des tutoriels intelligents (ITS) et des systèmes hypermédias adaptatifs (AHS) où l'utilisation de ressources pédagogiques élémentaires est très répandue, l'approche la plus commune pour construire des entrepôts de ressources est plutôt de concevoir le contenu de ces ressources à l'aide de systèmes auteurs dédiés (Brusilovsky, 2003).

D'autres projets tels que Ariadne (Duval *et al.*, 2001) ou Memorae (Abel *et al.*, 2004) ont également proposé des entrepôts d'objets pédagogiques – ou Learning Object Repositories (LOR) - où les ressources sont stockées et annotées, avec des méta-données standard dans le premier cas, avec des ontologies dans le deuxième. Mais aucune hypothèse n'est faite sur l'origine des ressources, par exemple si elles ont été conçues spécialement pour constituer ces entrepôts ou bien si elles ont été extraites de documents existants. Dans l'entrepôt OLR (Brase & Nejdil, 2003), les ressources pédagogiques sont annotées le langage RDF et sont organisées hiérarchiquement en tirant partie de la structure en chapitres du document dont elles sont tirées. Les ressources héritent ainsi les annotations de leurs « ancêtres » dans la structure. D'une manière générale tous ces LORs sont munis d'outils d'annotation qui permettent d'associer des méta-données aux ressources.

Dans TRIAL SOLUTION un document « sur mesure » est construit à partir des ressources « découpées », cette technique est également celle proposée par (Crampes & Ranwez, 2000) où le parcours de lecture individuel est calculé à partir des annotations sur les documents. Ces annotations sont issues d'ontologies et exprimées dans un langage s'inspirant des graphes conceptuels.

L'outil d'annotation de TRIAL SOLUTION est comparable aux outils d'annotation des LORs; la spécificité de l'approche adoptée dans le projet est la combinaison d'un processus d'annotation manuelle des ressources avec une phase préliminaire d'extraction et d'annotation automatique des ressources à partir de documents électroniques.

3.2 TRIAL SOLUTION et les standards du Web

La plateforme TRIAL SOLUTION est basée sur différents standards publics du web. Elle utilise le standard IMS Content Packaging (IMS, 2001) pour la modélisation des objets qui doivent être échangés entre les différents outils qui la composent. La spécification de ce standard est dédiée à la conception de formats d'échange, permettant en particulier l'agrégation et la désagrégation de ressources.

Les méta-données associées aux ressources pédagogiques sont conformes aux standards Dublin Core Metadata (DCMI, 1999), IMS Learning Resource Metadata (IMS, 2002) et LOM Learning Object Metadata (IEEE LTSC, 2002). Des adaptations ont cependant été nécessaires pour représenter de nouveaux matériels (Lenski & Wette-Roch, 2001). Il a été nécessaire de spécifier un format de méta-données propre au projet, pour pallier certains défauts des spécifications des standards disponibles à l'époque de la conception du projet.

Les technologies ont depuis rapidement évolué. L'émergence des « Web services » (W3C, 2004c) répondrait aujourd'hui aux besoins d'échanges d'objets complexes entre client et serveur, aussi bien pour l'outil d'annotation que pour celui de recherche de ressources. De même, les langages du Web sémantique - RDF (W3C, 1999), RDFS (W3C, 2004a) et OWL (W3C, 2004b) – conviendraient davantage à la représentation des méta-données ; ils faciliteraient en outre leur échange et leur réutilisation.

3.3 Vers une approche « Web sémantique organisationnel »

Le projet TRIAL SOLUTION au regard de l'approche adoptée dans la construction d'un entrepôt de ressources pédagogiques, s'accorde avec les enjeux du Web sémantique. Nous avons construit un entrepôt de ressources pédagogiques destinées à être partagées et réutilisées par une communauté d'utilisateurs. Ces ressources peuvent être efficacement recherchées grâce à leurs annotations sémantiques qui portent sur leurs contenus, leurs rôles pédagogiques, leurs interrelations structurelles et sémantiques. En somme, l'approche générale n'est pas différente de celle qui préside à la construction d'une mémoire organisationnelle pour une communauté donnée.

3.3.1 Utilisation des standards du Web sémantique

A l'époque de la conception du projet, les participants étaient étrangers à la communauté du Web sémantique et n'étaient pas conscients de son essor et des perspectives que cela ouvrait. A l'issue du projet, nous préconisons une représentation homogène de l'ensemble des méta-données, sous la forme d'annotations RDF basées

sur des ontologies RDFS ou OWL. Outre l'homogénéité de représentation qui facilite le partage et la réutilisation des annotations, ces langages offrent une puissance de représentation bien supérieure à celle de simples mots-clés pour décrire les ressources.

3.3.2 Extraction automatique d'annotations à partir de texte

La seconde leçon que nous tirons de TRIAL SOLUTION concerne la démarche adoptée pour la population et l'annotation d'un entrepôt de ressources pédagogiques. L'expérience nous a montré que l'activité d'annotation manuelle des ressources est accablante pour des enseignants ou d'autres scientifiques, face à une importante quantité de ressources. Nous pensons qu'il est important d'automatiser autant que faire se peut ce processus ; notre expérience d'extraction automatique de ressources à partir de documents structurés est tout à fait concluante et les techniques mises en œuvre pour l'annotation automatique des ressources extraites pourraient être améliorées.

Cette approche n'exclue cependant pas l'utilisation d'un éditeur d'annotations. L'expérience montre que la mise au point des annotations ne se passe pas d'un expert du domaine et/ou d'un ingénieur de la connaissance et un outil d'édition des annotations reste nécessaire.

3.3.3 Explicitation collaborative du rôle pédagogique des ressources

Corollairement, il apparaît que le succès de l'étape d'extraction automatique de méta-données à partir du contenu textuel des ressources pédagogiques dépend largement de la collaboration des auteurs lors de la conception des documents originaux. Nous avons déjà souligné l'importance de l'utilisation de styles standards dans la conception des documents : ils permettent d'extraire facilement le type des ressources et le réseau sémantique selon lequel elles s'organisent. Une phase d'acquisition de connaissances auprès de l'auteur des documents permettrait de mieux expliciter la correspondance entre l'organisation pédagogique du document et sa mise en forme et d'améliorer ainsi l'interprétation de sa mise en forme, e.g. la sémantique des différents styles, du code des couleurs, etc. Les annotations sur les caractéristiques pédagogiques des ressources seraient ainsi enrichies.

Des méthodes de traitement automatique du langage naturel (TALN) pourraient quant à elles améliorer l'extraction du contenu sémantique des ressources pédagogiques. Nous n'avons pas davantage exploré cette direction de recherche ; cela dépasse le cadre de cet article. Nous préconisons plutôt de minimiser l'utilisation des techniques de TALN et l'extraction du contenu sémantique des ressources pour se focaliser sur l'extraction de l'organisation pédagogique des ressources. Nous pensons que cette approche est possible pour la construction d'entrepôts de ressources pédagogiques concernant un domaine spécifique (par opposition à des entrepôts généraux) : dans ce cas, la connaissance du domaine devient implicite ; une annotation minimale sur le domaine suffit et c'est l'annotation des ressources sur leurs caractéristiques pédagogiques qui va permettre de les rechercher efficacement et de les réorganiser dans des documents personnalisés.

3.3.4 Définition de scénarios d'apprentissage précis

Enfin, dans le projet TRIAL SOLUTION, les buts des utilisateurs finaux étaient très généraux ce qui a rendu la tâche d'annotation difficile. Pour composer un document personnalisé, l'utilisateur est susceptible de rechercher n'importe quelle ressource, dans n'importe quel contexte. Nous préconisons la définition de scénarios d'apprentissage plus spécifiques et de profils d'apprentissage pour améliorer l'adéquation du contenu des annotations aux requêtes des utilisateurs.

En somme, l'expérience du projet nous conduit vers une approche Web sémantique dans la conception d'un entrepôt de ressources pédagogiques, cet entrepôt étant dépendant d'un domaine, d'une communauté, tout comme une mémoire organisationnelle. L'utilisation des standards du Web sémantique est la première conséquence de ce bilan. Nous préconisons l'implication des auteurs des documents originaux en amont du processus d'extraction automatique d'annotations portant sur le rôle pédagogique des ressources à partir de la mise en forme du document. La prise en compte des buts précis des utilisateurs devrait permettre de mieux définir la nature des annotations intéressantes pour la recherche ultérieure des ressources annotées.

4 Conclusion et perspectives

Nous avons présenté le projet TRIAL SOLUTION dédié à la publication de documents personnalisés, basée sur des livres scientifiques existants. L'approche générale consiste à construire un entrepôt de ressources pédagogiques et pour cela à extraire et annoter des ressources à partir de documents textuels électroniques. Les annotations sont le support d'une recherche sémantique de ressources dans l'entrepôt, par les utilisateurs finaux, étudiants ou enseignants, dont l'activité est de composer des documents personnalisés. L'automatisation du processus d'annotation exploite la mise en forme des documents et l'outil dit de re-engineering que nous avons développé permet d'éditer le contenu des ressources et leurs annotations afin d'enrichir manuellement les annotations et le thesaurus à partir duquel ces dernières sont construites.

Nous avons dressé un bilan du projet dans la perspective du Web sémantique. Les standards du Web sémantiques émergeaient à peine au moment de la conception du projet, si bien que les annotations des ressources suivent LOM mais ni OWL ni RDF(S). Cependant, l'approche générale adoptée dans le projet s'accorde tout à fait avec les enjeux du Web sémantique. Nous avons construit un entrepôt de ressources pédagogiques annotées comme une mémoire organisationnelle. Ces ressources sont destinées à être partagées et réutilisées, leurs annotations sont basées sur une pseudo-ontologie et portent sur leur contenu sémantique, leurs rôles pédagogiques et leurs interrelations.

Les leçons que nous avons pu tirer de ce projet nous amènent à préconiser une approche « mémoire organisationnelle » dans la construction d'un entrepôt de ressources pédagogiques dédié à un domaine particulier ; à utiliser les standards du

Web sémantique pour annoter ces ressources ; à accentuer l'automatisation du processus d'extraction de ressources et d'annotation à partir de documents textuels ; à impliquer l'auteur du document dans ce processus d'annotation pour expliciter la correspondance entre sa stratégie pédagogique et la mise en forme de son document ; à définir des scénarios pédagogiques qui seront pris en compte dans le contenu des annotations.

Dans la continuité de ce projet, nos travaux actuels s'inscrivent dans le cadre de l'action scientifique Web-Learn dont le but général est d'étudier les bénéfices méthodologiques et techniques d'une approche Web sémantique pour le E-Learning. Nous tirons partie de notre expérience de TRIAL SOLUTION et travaillons à une méthode collaborative de construction d'une mémoire organisée de ressources pédagogiques à partir de documents existants (Dehors *et al.*, 2005). Elle consiste dans un premier temps à acquérir auprès de l'auteur d'un document pédagogique la correspondance entre la mise en forme de son document et sa stratégie pédagogique. Nous construisons ensuite une ontologie permettant de représenter les caractéristiques pédagogiques explicitées. Les ressources pédagogiques et leurs annotations sont alors extraites du document en tirant partie de cette correspondance. Ces annotations sémantiques sont ensuite utilisées pour offrir différentes stratégies de navigation dans la mémoire des ressources pédagogiques d'un cours donné. Les ressources sont retrouvées en utilisant le moteur de recherche sémantique (CORBY *et al.*, 2004) ; la recherche est basée sur les annotations RDF des ressources et l'ontologie RDFS sur laquelle elles sont construites.

Références

- M.-H. ABEL, C. BARRY, A. BENAYACHE, B. CHAPUT, D. LENNE, C. MOULIN (2004), Using an Organizational Memory for e-learning, in Proceedings of Workshop of Knowledge Management and Organizational Memories (ECAI'2004), Valencia, Spain, 2004
- J. BRASE & W. NEJDL (2003), Ontologies and Metadata for eLearning, Handbook on Ontologies, pp. 579-598, Springer Verlag, 2003.
- BRONSTEIN, SEMENDJAJEW, MUSIOL & MÜHLIG (2000), *Taschenbuch der Mathematik*, Verlag Harri Deutsch, 2000.
- P. BRUSILOVSKY (2003), Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools. In T. Murray, S. Blessing and S. Ainsworth Eds.: *Authoring Tools for Advanced Technology Learning Environment*. Dordrecht: Kluwer Academic Publishers, pp. 377-409, 2003.
- O. CORBY, R. DIENG-KUNTZ & C. FARON-ZUCKER (2004), Querying the Semantic Web with the Corese Search Engine, in Proceedings of the European Conference on Artificial Intelligence (ECAI'2004), subconference PAIS, pp. 705-709, Valencia, Spain, 2004.
- M. CRAMPES & S. RANWEZ (2004) Ontology-Supported and Ontology-Driven Conceptual Navigation on the World Wide Web, in Proceedings of 11th ACM Conference on Hypertext, San Antonio, Texas, 2000.
- I. DAHN (2001), Slicing Book Technology – Providing Online Support for Textbooks, in Proceedings of the International Conference on Distant Education, ICDE'2001, Dusseldorf, Germany, 2001.

Web sémantique pour le E-learning

I. DAHN, M. ARMBRUSTER, U. FURBACH & G. SCHWABE (2001). Slicing Books – The Authors' Perspective, in R. Bromme, E. Stahl Eds. : *Writing Hypertext and Learning: Conceptual and Empirical Approaches*, Pergamon, 2001.

DCMI (1999), Dublin Core Metadata Element Set, Version 1.1: <http://dublincore.org/documents/dces/>

S. DEHORS, C. FARON-ZUCKER, A. GIBOIN & J.P. STROMBONI (2005), Un web sémantique de formation par questionnement, in Actes de la journée EIAH, EGC2005, Paris, 2005.

C. DESMOULINS, M. GRANDBASTIEN (2002) Des ontologies pour la conception de manuels de formation à partir de documents techniques. In *Sciences et Techniques Educatives*, vol.9, n°3-4, Hermès.

E. DUVAL, E. FORTE, K. CARDINAELS, B. VERHOEVEN, R. VAN DURM, K. HENDRIKX, M. WENTLAND-FORTE, N. EBEL, M. MACOWICZ, K. WARKENTYNE & F. HAENNI (2001), The ARIADNE Knowledge Pool System, in *Communications of the ACM*, 44 (5), pp. 73-78.

IEEE LTSC (2002), LOM, Learning Object Metadata: <http://ltsc.ieee.org/wg12/>

IMS (2001), IMS Content Packaging Specification: <http://www.imsproject.org/content/packaging/index.html>

IMS (2002), Learning Resource Metadata, <http://www.imslobal.org/metadata/index.html>

W. LENSKI, E. WETTE-ROCH (2001). The Trial-Solution Approach to Document Re-use, in *Proceedings of Workshop on Electronic Media in Mathematics*, Coimbra, Portugal.

P. RIGAUX & N. SPYRATOS (2003), Metadata Management and Learning Object Composition in a Self e-Learning Network, in *Proceedings of Workshop on Information Search, Integration and Personalization*, Japan, 2003.

B. SIMON & J. QUEMADA (2002), A Reflection of Metadata Standards based on Reference Scenarios, 2002, <http://www.wu-wien.ac.at/usr/wi/bsimon/publikationen/SimonQuemada-ReflectionOnMetadataStandards.pdf>

TRIAL SOLUTION (2001), EU project, Information Society Technologies Program (IST): <http://www.trial-solution.de/>

W3C (1999), RDF, Resource Description Framework: <http://www.w3.org/RDF/>

W3C (2004a), RDFS, RDF Vocabulary Description Framework: <http://www.w3.org/TR/rdf-schema/>

W3C (2004b), OWL, Web Ontology Language: <http://www.w3.org/TR/owl-features/>

W3C (2004c), Web Services: <http://www.w3.org/2002/ws/>

WEB-LEARN (2004), Action Scientifique CNRS et Action Colors INRIA: <http://www.lirmm.fr/~toutou/as-weblearn/>

Des Annotations Sémantiques pour Apprendre. QBLS : Modélisation et Expérimentation

Sylvain Dehors¹, Catherine Faron-Zucker², Jean Paul Stromboni², Alain Giboin¹

¹ACACIA, INRIA

2004 route des Lucioles, BP 93, 06902 Sophia Antipolis cedex
{sdehors, giboin}@sophia.inria.fr

²MAINLINE, I3S, UNSA

930 route des Colles, bât ESSI, BP 145, 06930 Sophia Antipolis cedex
{faron, strombon}@essi.fr

Résumé : Mettre un cours à la disposition des apprenants par le biais d'un EIAH impose de la part de l'auteur un effort de formalisation, à la fois des ressources disponibles et de la stratégie pédagogique envisagée. Dans ce contexte nous étudions l'apport de techniques inspirées du web sémantique et leurs implications tant pour l'enseignant que pour l'apprenant. Partant d'un exemple de « mise en ligne » d'un cours, motivé par une volonté pédagogique particulière, nous démontrons la faisabilité de cette approche ainsi que ses avantages et inconvénients. Cette analyse est étayée par une expérience d'utilisation réelle dont nous présentons ici les premiers résultats

Mots-clés : Web Sémantique, e-learning, EIAH, ontologie, annotation sémantique

1 Introduction

Dans le contexte actuel d'introduction de l'informatique dans les environnements d'apprentissage (cartables électronique, supports numériques, etc..), nous avons étudié un scénario d'évolution d'un enseignement classique vers un enseignement assisté par un outil d'apprentissage informatique : le système QBLS (pour *Question Based Learning System*), que nous avons développé à cet effet.

De manière générale notre travail s'inscrit dans la perspective actuelle d'utilisation de plus en plus large d'outils informatiques dédiés à l'apprentissage au sein du campus. Proposer ainsi de nouveaux supports pédagogiques, comme les cours en ligne ou des outils de simulation, bouleverse non seulement les pratiques et les outils classiques mais impose d'envisager une nouvelle démarche pédagogique, sur laquelle des réflexions ont déjà été menées par l'un d'entre nous (Stromboni, 2002).

Dans ce contexte, notre but est d'expérimenter des techniques de gestion des connaissances envisagées dans le cadre du web sémantique (Dieng *et al.*, 2002). Nous avons cherché à évaluer l'apport de ces techniques, notamment l'introduction d'annotations sémantiques (Azouaou *et al.*, 2003) pour décrire des ressources

pédagogiques. Cette introduction est mise en parallèle avec la poursuite d'un objectif d'apprentissage prédéterminé.

Adoptant une démarche résolument pragmatique, notre travail jette un pont entre les aspects de modélisation, de sémantique, et la pratique pédagogique telle qu'elle se présente aujourd'hui dans le domaine de l'enseignement supérieur. Dans le scénario que nous avons adopté, le support de cours initial constitue le point de départ incontournable. Les annotations sémantiques ajoutées sur ce cours, permettent une mise en ligne automatisée et sont exploitées pour favoriser la compréhension et l'utilisation du contenu pédagogique. L'application développée s'intègre dans le cadre d'un enseignement où l'outil informatique est utilisé comme vecteur principal d'accès à l'information.

Dans la section suivante nous positionnons cette contribution par rapport aux travaux existants. Une description du dispositif QBLS est ensuite présentée détaillant les aspects de modélisation, d'architecture et d'interface. La quatrième partie s'attache à décrire les expériences que nous avons menées avec cet outil ainsi que les premiers résultats obtenus.

2 Etat de l'art

Dans les différents travaux concernant l'accès à un « cours en ligne » on distingue ceux où le contenu est simplement présenté, généralement sous la forme d'un hypertexte ; ceux où le contenu est adapté et personnalisé ; et ceux où le support est créé sur mesure à partir d'éléments existants. Concernant cette distinction, on notera que (Brusilovsky & Vassileva, 2003) préfèrent séparer les systèmes selon le moment où se déroule l'opération d'adaptation, soit avant l'entrée de l'utilisateur, soit pendant son parcours, soit les deux. Ces critères se recoupent fortement. Les exemples cités ci-dessous n'ont qu'une valeur illustrative, il en existe bien d'autres.

Les systèmes de « présentation » insistent sur la navigation, la visualisation et l'organisation des informations pour faciliter leur compréhension par le plus grand nombre. Le système QBLS se place dans cette perspective.

Les travaux visant la personnalisation du contenu relèvent du domaine des hypermédias adaptatifs. L'utilisateur est guidé dans sa navigation hypertextuelle par des techniques adaptatives (De Bra *et al.*, 2003) (Murray, 2003). La plupart de ces techniques font appel à des modèles du domaine abordé par le cours couplés à des modèles de l'utilisateur exprimant l'état des connaissances par rapport au domaine (*overlay model*). L'adaptation consiste généralement à guider la navigation dans l'ensemble des ressources.

Lors d'une personnalisation du support préalablement à son utilisation le système construit un document de cours sur mesure à partir d'un ensemble de ressources annotées. Par exemple (Crampes & Ranwez, 2000) propose de construire le cours à l'aide d'algorithmes organisant les différentes ressources annotées avec des ontologies. On retrouve la même démarche dans (Rigaux *et al.*, 2003) où les ressources sont formées par des paragraphes au format XML DocBook et sont indexées par les termes de la classification ACM, et dans (Dahn *et al.*, 2001) où les

ressources sont des paragraphes issus de livres préalablement "découpés" et annotés. Dans ce cas le cours résultant est ciblé et logiquement ordonné, l'apprenant y dispose d'une liberté de navigation réduite.

Un autre aspect de l'accès à l'information pédagogique est celui abordé par les travaux sur les LCMS (Learning Content Management Systems) où l'intérêt porte plus sur la recherche et l'accès aux données pédagogiques que sur leur présentation. Le terme consacré de LOR (Learning Object Repository) est d'ailleurs généralement utilisé pour désigner les dispositifs contenant l'ensemble des ressources accessibles. Les éléments d'information utilisés pour la recherche sont des méta-données souvent standardisées (Duval *et al.*, 2001). L'utilisation d'ontologies pour indexer les ressources (Desmoulins & Grandbastien, 2000) permet d'envisager des recherches plus riches. Les ressources sont alors organisées en "mémoire de formation" (Abel *et al.*, 2004) s'inspirant de l'idée du Web Sémantique (Berners-Lee *et al.*, 2001) (Stoanovic *et al.*, 2001) appliquée à un ensemble restreint de ressources pédagogiques.

Le problème de la création des ressources est abordé selon divers angles, soit par le biais d'outils auteur (Brusilovsky, 2003), soit par la réutilisation de ressources existantes (Rigaux *et al.*, 2003) (Dahn *et al.*, 2001). Ce dernier point est souvent cité comme un avantage du fait de l'existence d'un grand nombre de ressources pédagogiques accessibles aujourd'hui sur le web. La création des annotations va de pair avec le problème de création des ressources, et les mêmes solutions sont proposées : soit l'annotation est manuelle, soit elle est générée automatiquement à partir des documents existants. Les outils d'extraction d'annotations reposent généralement sur le traitement automatique de la langue et la structure des documents.

Le système QBLS, décrit dans la partie suivante, s'inspire des approches précédemment citées. Il s'agit d'un système de visualisation de ressources sous la forme d'un hypertexte se concentrant sur l'aspect de présentation de l'information. Par contre les techniques mises en œuvre pour cela sont plus généralement destinées à la recherche de documents dans les LCMS, notamment l'approche « web sémantique de formation » (voir section suivante). Pour la génération des ressources nous avons utilisé un document préexistant que nous avons mis en forme puis traité automatiquement afin d'en extraire le contenu et les annotations associées.

3 QBLS

Pour l'étudiant, QBLS est un outil d'aide à la résolution de questions de Travaux Dirigés (TD) à partir d'éléments de cours annotés sémantiquement. Pour l'enseignant, QBLS est vu non seulement comme un outil permettant de diffuser son support électronique sous une forme facilitant la navigation, mais également comme un moyen d'inciter les étudiants à aller chercher activement les connaissances.

3.1 Modélisation

Un des principes pédagogiques à la base de la conception de QBLS est d'aborder l'apprentissage par une stratégie pédagogique de questionnement. L'idée est d'utiliser le cycle ou rythme ternaire décrit dans (Beau, 2005) qui organise l'assimilation en trois étapes. La première, dite *heuristique*, cherche à mettre en évidence le besoin des connaissances du cours. Survient alors la seconde étape, ou *démonstration*, qui donne accès aux connaissances recherchées. Enfin la troisième étape, ou *application*, consiste en une mise en œuvre des connaissances par l'apprenant.

Dans cette optique, QBLS propose d'entrée un ensemble de questions reliées aux connaissances clés du cours, et l'objectif de l'apprenant est de trouver la réponse à ces questions, c'est l'étape heuristique. Des accès aux connaissances utiles sont suggérés pour répondre aux questions, ce qui matérialise l'étape de démonstration. La résolution des questions posées à l'apprenant constitue l'application, l'enseignant compte d'ailleurs tirer un bénéfice pédagogique de l'exploitation des erreurs de l'apprenant.

Il faut souligner que toutes ces particularités de QBLS, qu'il s'agisse de questionnement, de recherche d'information, de résolution, et d'analyse d'erreur contribuent à rendre l'élève apprenant actif et acteur pour faciliter l'assimilation.

Pour assurer l'étape de démonstration, on ne se contente pas d'un document de cours classique parsemé de points d'entrée matérialisés par exemple par des ancres html. Le document de cours a été analysé en détail et structuré en *fiches* d'informations afin de faire ressortir les savoirs clés. Ceux-ci sont organisés en *thèmes* - les objectifs explicites du cours – et en *notions* - les concepts qui concourent à les expliquer. Les références à des *notions* ou des *thèmes* dans le contenu des *fiches* forme alors un réseau de relations entre les *fiches* et les savoirs.

La navigation dans le contenu des fiches permet donc soit d'approfondir le sujet en suivant les liens à l'intérieur du contenu soit de suivre l'ordre des thèmes qui définit une progression pédagogique.

Pour terminer, il faut distinguer les fiches selon la nature de l'information qu'elles contiennent : général, particulier, théorique, appliqué, quand elles contribuent à décrire un thème ou une notion. Nous avons donc défini un ensemble de termes pour exprimer cette information : *Définition, Exemple, Formalisation, Précision*.

Cette analyse préliminaire du contenu, à la fois structurelle et qualitative, confère au document de cours une valeur ajoutée à la fois pour l'auteur et pour l'apprenant, puisqu'elle permet d'explicitier et de justifier la présence et l'importance des informations placées dans le cours.

En s'inspirant de l'idée de Web Sémantique, le cours est alors vu comme un ensemble de ressources (les fiches) interconnectées. L'activité de lecture du cours prend la forme d'une navigation au sein de cet ensemble. Chaque ressource peut être annotée sémantiquement par rapport à une ou plusieurs ontologies, et cette information sémantique est utilisée pour faciliter la compréhension et les choix de navigation dans l'ensemble des ressources.

Deux familles d'ontologies interviennent classiquement pour indexer les ressources pédagogiques (Mizogushi *et al.*, 1997). Le premier type regroupe les ontologies du domaine, elles fournissent le vocabulaire conceptuel permettant de décrire le contenu de chaque ressource vis-à-vis du domaine étudié. Le second type concerne les ontologies dites « pédagogiques », cela regroupe les modèles utilisés pour décrire l'approche pédagogique mise en œuvre, elles décrivent notamment le rôle pédagogique que peut occuper une ressource. Par exemple on distingue classiquement les exemples, les preuves, les explications, etc. (Ullrich, 2004)

À l'issue de l'interview de l'enseignant auteur du document de cours initial nous avons identifié les différents rôles joués par les ressources dans la démarche pédagogique. Ceci a permis de définir l'ensemble des concepts nécessaires pour exprimer le rôle pédagogique joué par une *fiche* (voir Figure 1). La granularité étant faible il ne nous a pas semblé possible qu'une fiche puisse jouer plusieurs de ces rôles à la fois, par contre il est clairement apparu qu'une fiche pouvait jouer son rôle pour plusieurs notions différentes, en particulier les exemples de courbes illustrent plusieurs notions à la fois (ex : « signal bloqué » et « bloqueur »)

Les types de savoirs clés sont identifiés à partir de l'observation du document initial : le *cours* (le sujet du cours en lui-même), les *thèmes* (problématiques clés), les *questions* (exercices d'application) et les *notions* (concepts classiques du domaine). Ils sont regroupés sous le terme de *Ressource Pédagogique Abstraite*. L'ontologie résultant de cette modélisation est présentée en Figure 1, elle ne concerne que les aspects pédagogiques.

L'ontologie du domaine se limite ici à un thésaurus. Les concepts du domaine sont des instances des classes modélisant les types de savoirs clés du cours. L'ontologie unique de QBLS traite donc seulement de la pédagogie et se place à un niveau « méta » par rapport au domaine.

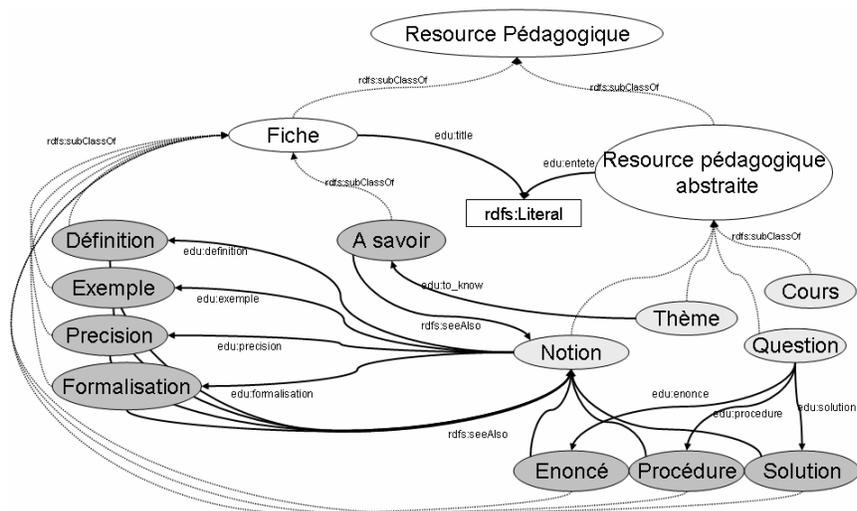


Fig. 1– L'ontologie de QBLS

On remarque que la majorité des concepts de cette ontologie sont présents dans l'ontologie proposée par (Ullrich, 2004) et que le modèle global du cours auquel nous avons abouti est proche de celui proposé par (Hewy, 2003). Cependant même avec ces similarités, nous défendons ici l'idée qu'utiliser un modèle prédéfini par ailleurs pour annoter le document existant aurait nécessité un effort très rebutant pour l'enseignant/annotateur.

Une fois l'ontologie établie, les annotations sémantiques des différentes parties du document de cours initial sont déduites de la mise en forme. Ces annotations expriment le rôle pédagogique de chaque paragraphe (ex: telle partie est une définition, telle autre est un exemple), et fixent la représentation du domaine enseigné en désignant l'ensemble des notions et des thèmes clés en accord avec l'ontologie décrite ci-dessus. La technique employée pour extraire les annotations est décrite dans (Dehors *et al.*, 2005). La figure 2 montre un exemple d'annotation RDF. Celle-ci décrit la notion de « signal discret » et indique que deux fiches (un exemple et une définition) sont reliées à cette notion.

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:edu="http://www.inria.fr/acacia/exp_weblearn#">
<edu:Notion rdf:about="http://www.inria.fr/acacia/exp_weblearn#d0e971">
  <edu:definition rdf:resource="#d0e980"/>
  <edu:exemple rdf:resource="#d0e992"/>
  <edu:entete>signal discret</edu:entete>
  <edu:entete>discret</edu:entete>
</edu:Notion>
<edu:Definition rdf:about="http://www.inria.fr/acacia/exp_weblearn#d0e980">
  <edu:titre>Définir : signal discret, ou signal à temps discret.</edu:titre>
  <edu:contenu rdf:resource="/fiches/d0e980.xml"/>
  <rdfs:seeAlso rdf:resource="#d0e940"/>
  <rdfs:seeAlso rdf:resource="#d0e950"/>
</edu:Definition>
<edu:Exemple rdf:about="http://www.inria.fr/acacia/exp_weblearn#d0e992">
  <edu:titre>E: Illustration de signal discret.</edu:titre>
  <edu:contenu rdf:resource="/fiches/d0e992.xhtml"/>
</edu:Exemple>
</rdf:RDF>
```

Fig. 2 - Annotation de la notion de signal discret

3.2 Architecture

Les ressources issues du processus d'extraction à partir du document initial sont stockées avec leurs annotations respectives dans un « entrepôt de connaissances pédagogiques ». Le système QBLS exploite alors les annotations sémantiques formalisées dans le langage RDF à l'aide de requêtes au moteur Corese (Corby *et al.*,

2004). Cet ensemble forme ce que nous appelons un "web sémantique de formation" dont l'architecture est illustrée sur la Figure 3. Cette architecture est basée sur l'utilisation des logiciels à notre disposition à savoir un serveur tomcat, et le moteur de recherche sémantique Corese. Ainsi l'effort de développement a uniquement porté sur la partie « engin de visualisation », et en combinant les technologies JSP et XSL pour traiter les résultats RDF du moteur de recherche, le coût de développement est minimal.

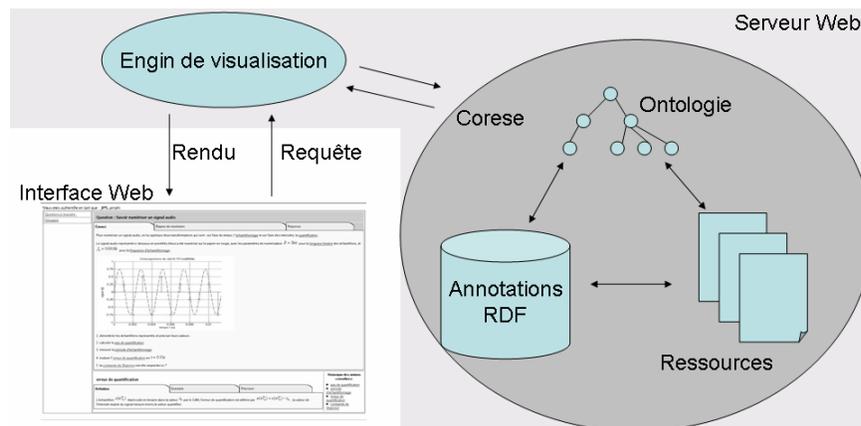


Fig. 3 – L'architecture du système QBLS

Outre la simplification de la tâche du développeur, l'utilisation de Corese permet d'exploiter la sémantique des relations reliant notions et fiches. Par une requête unique on peut en effet retourner la référence de toutes les fiches qui sont reliées à une notion ou un thème particulier et obtenir uniquement le contenu de la fiche « prioritaire » (celle qui doit s'afficher par défaut). Ainsi l'engin de visualisation n'effectue pas ce choix il ne fait que traduire le résultat du moteur. Pour déterminer la fiche « prioritaire » nous avons introduit une règle dans le moteur indiquant que les liens de définitions, énoncés, et « à savoir » étaient de type «edu:lienPrioritaire». Ainsi la requête demande de renvoyer les fiches reliées par un lien prioritaire avec leur contenu et s'il n'y en a pas alors la première non prioritaire est renvoyée.

On notera qu'une fiche d'Exemple peut servir à illustrer plusieurs concepts du domaine, il suffit qu'elle soit reliée par une relation d'exemple à ces concepts. Grâce à l'utilisation d'un moteur de recherche sémantique la question de l'aspect prioritaire de cette fiche, qui peut varier d'un cas à l'autre, est résolue par le moteur qui prend en compte cette information dans sa réponse.

3.3 Interface

Un travail important a été réalisé au niveau de l'interface pour qu'elle rende compte de l'organisation sémantique des ressources pédagogiques (et à travers celle-ci l'approche pédagogique de l'enseignant auteur). Nous avons réalisé un travail

itératif en collaboration avec un ergonome pour limiter les problèmes liés directement à l'interface. L'interface a donc fait l'objet de tests d'utilisabilité (par la méthode d'évaluation heuristique). Ces tests ont permis de régler les problèmes d'usage les plus criants. Ensuite un pré-test a été conduit auprès d'un échantillon d'étudiants afin de valider la faisabilité de l'expérience grandeur nature envisagée et de prévoir plus finement son déroulement. L'interface qui résulte de ce travail de co-conception informatique et ergonomique est présenté ci-dessous.

Vous êtes authentifié en tant que : JPS_amphi

Questions à résoudre : Glossaire

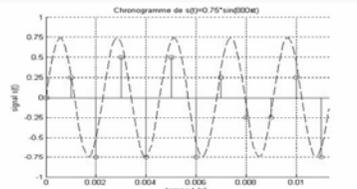
Question : Savoir numériser un signal audio

Étapes de résolution Réponse

Énoncé

Pour numériser un signal audio, on lui applique deux transformations qui sont : sur l'axe du temps, l'échantillonnage et sur l'axe des intensités, la quantification.

Le signal audio représenté ci-dessous en pointillés bleus a été numérisé sur le papier en rouge, avec les paramètres de numérisation $B = 384\text{it}$ pour la longueur binaire des échantillons, et $f_s = 1000\text{Hz}$ pour la fréquence d'échantillonnage.



1. dénombrer les échantillons représentés et préciser leurs valeurs
 2. calculer le pas de quantification
 3. mesurer la période d'échantillonnage
 4. évaluer l'erreur de quantification en $T = 0.01\text{s}$
 5. la contrainte de Shannon est-elle respectée ici ?

erreur de quantification

Définition Exemple Précision

L'échantillon $x(nT_s)$ étant codé en binaire dans la valeur \hat{x}_n par le CAN, l'erreur de quantification est définie par $e(nT_s) = x(nT_s) - \hat{x}_n$, la valeur de l'intensité exacte du signal mesuré moins la valeur quantifiée.

Historique des notions consultées :

- pas de quantification
- erreur d'échantillonnage
- erreur de quantification
- contrainte de Shannon

Fig. 4 – L'interface de QBLS

Dans cette interface chaque « ressource abstraite » peut être décrite sous plusieurs aspects (les fiches) présentés dans un ensemble d'onglets répartis horizontalement. Chaque onglet représente graphiquement une relation sémantique entre un concept abstrait et une fiche (définition, exemple, précision...). Le rôle de la fiche est exprimé par l'intitulé de l'onglet. La partie supérieure de la fenêtre présente le cours, les thèmes ou les questions, la partie inférieure permet de visualiser les notions. Lorsqu'une notion est citée dans le contenu d'une fiche, un lien hypertexte permet de dévoiler la notion correspondante qui remplace alors dans la zone inférieure la précédente. Ces liens hypertextes sont la représentation graphique des relations *seeAlso* entre une fiche et une notion citée par cette fiche.

Ainsi la figure 4 montre, dans la partie supérieure de la fenêtre, une question liée à trois fiches symbolisées par les trois onglets : l'énoncé (la fiche visible), les étapes de résolution et la réponse (les deux onglets « en retrait »). Lorsque l'utilisateur clique sur un lien hypertexte dans le contenu de l'énoncé la partie inférieure affiche la notion sélectionnée, ici l'« erreur de quantification » avec ses trois fiches : définition, exemple, précision. La fiche ouverte initialement par défaut est la définition. Au

moment de la capture d'écran, l'utilisateur avait déjà visualisé d'autres notions qui sont rappelées dans la zone « historique des notions consultées » en bas à droite. On laisse à tout instant la possibilité d'accéder à la liste des questions, en haut à gauche de l'écran. Un glossaire complet donnant toutes les notions peut également être déplié ou replié à tout instant à gauche de l'écran.

4 Expérimentation

4.1 Dispositif expérimental

QBLS a servi de support pour une séance de TD et un cours magistral. Cet enseignement s'inscrit dans le cursus de formation de première année de l'Ecole Supérieure en Sciences Informatiques (ESSI) de l'université de Nice Sophia Antipolis.

En termes d'évaluation, nous avons orienté notre expérimentation selon les trois dimensions : utilité, utilisabilité et acceptabilité proposées par (Tricot *et al.*, 2003) en employant les outils empiriques préconisés. Nous avons tout d'abord cherché à évaluer la faisabilité d'un enseignement assisté par QBLS. Au delà d'une évaluation du système en lui-même nous souhaitons qualifier l'apport de l'annotation sémantique dans cette démarche globale d'enseignement assisté par ordinateur. Ceci aussi bien du point de vue de l'enseignant, auteur de l'annotation, que des élèves utilisateurs du système.

Lors du cours magistral, l'interface était aménagée pour permettre une projection sur grand écran. Elle a servi donc de support à la manière de transparents Power-Point. L'enseignant, l'un des chercheurs de notre groupe, a été observé pendant son cours par deux d'entre nous.

Lors de la séance de Travaux Dirigés, les étudiants étaient face au système et devaient résoudre une liste de questions proposées par le système en utilisant les possibilités de navigation dans le cours. Les étudiants devaient écrire leurs réponses sur des formulaires de réponse papier. Nous avons tenté de mesurer l'apport de l'annotation sémantique de la façon suivante : 1/ au travers d'un questionnaire rempli par les étudiants à la fin du test, 2/ au travers de nos observations durant la séance, 3/ par l'analyse des logs d'interaction avec le système.

4.2 Faisabilité d'un enseignement assisté par QBLS

4.2.1 Le point de vue de l'apprenant

Dans l'ensemble, les étudiants ont correctement utilisé le système et n'ont pas semblé désorientés. L'utilisation de QBLS par l'enseignant pour le cours magistral a permis aux étudiants de se familiariser avec le fonctionnement de l'interface. Certaines utilisations atypiques ont été observées, nous les détaillons dans la suite.

Un temps d'adaptation à cette nouvelle forme d'enseignement a été nécessaire pour certains étudiants. Il est arrivé que le formulaire de réponse papier introduise des

incompréhensions. Certains étudiants ont notamment cherché les questions sur le formulaire papier et ont interprété le titre des thèmes comme des questions ouvertes; d'autres n'ont pas immédiatement trouvé les questions sur QBLS.

Les étudiants n'ont dans l'ensemble pas été curieux d'explorer QBLS davantage que nécessaire pour répondre aux questions posées. Les étudiants entrent dans une démarche de recherche d'information suscitée par la question, trouvent cette information en naviguant et l'appliquent immédiatement. Le système n'est pas perçu comme permettant d'approfondir ses connaissances. Ceci est apparu très clairement à la fois dans les réponses au questionnaire de satisfaction et quand les élèves ont été confrontés à une erreur dans les liens. L'erreur a été introduite fortuitement par le professeur et le parcours des notions suggérées dans l'énoncé ne permettait plus de répondre à la question. Un élève a alors interpellé le professeur non pour lui demander où poursuivre sa recherche mais pour lui signaler que l'erreur, réaction relayée par d'autres étudiants au même moment. Les étudiants ont donc perçu le système comme permettant de trouver directement l'information et non comme un support plus riche, ce qui explique que les parties du cours non couvertes par les questions n'aient pas été visitées.

Il semble que la lecture des fiches associées aux notions explorées a été effective et complète. Les élèves en difficulté n'essayaient pas de comprendre en lisant plus de fiches mais faisaient directement appel à l'enseignant. Comme le note (Paquette *et al.*, 2003) dans un contexte de téléapprentissage, la présence du tuteur est primordiale, cette expérience montre que ceci est aussi valable en présentiel, même lors d'une tâche pourtant prévue pour être menée seul.

Concernant les comportements marginaux on peut éventuellement les interpréter par une réticence vis-à-vis de ce nouveau mode d'apprentissage. En effet le principal comportement marginal a été observé chez un étudiant qui, pour répondre à chaque question, explorait méthodiquement le cours en ayant recours à une utilisation intensive du glossaire ce qui reproduit le comportement classique face à un document papier.

Le dernier point sur l'observation de l'expérience de TD aborde le problème posé par la démarche de questionnement. L'efficacité en terme de motivation pour se déplacer dans le cours semble tout à fait démontrée car les étudiants ont bien parcouru les fiches contenant les réponses. En revanche la phase qui a posé problème est celle de la mise en ligne des solutions. En effet il faut à un moment fournir la solution aux questions, pour que les étudiants puissent vérifier la validité de leur démarche.

Il apparaît que la mise en ligne des solutions ne rend cependant pas inutile la présence de l'enseignant. En effet les étudiants ont exprimé le besoin de confirmation par l'enseignant de leur bonne compréhension et de leur réussite à répondre aux questions. De plus les étudiants en difficulté sur une question n'attendent pas la mise en ligne des réponses et font appel à l'enseignant pour des explications personnalisées. Le système ne présente pourtant pas de problème d'utilité à ce niveau puisqu'il remplit bien la tâche de fournir les réponses, mais il est clair que cela ne suffit pas.

Il manque un signalement aux étudiants des moments de mise en ligne des solutions (Les étudiants demandent où ils peuvent trouver les solutions).

Pour pouvoir évaluer QBLS le formulaire de réponse comportait une consigne leur demandant d'analyser leurs erreurs éventuelles. La vérification par les étudiants de leurs réponses et l'analyse de leurs erreurs sont, de prime abord, déroutantes pour les étudiants. Ils ont tendance à corriger leurs réponses plutôt qu'à expliquer leurs erreurs.

Au-delà de l'expérimentation de QBLS, cette auto-analyse pourrait faire partie intégrante d'une stratégie pédagogique centrée autour de QBLS. Des formulaires de réponses en ligne pourraient aider à cette activité (Dans le second et surtout le troisième groupe, l'enseignant a insisté sur cet aspect et a demandé explicitement d'indiquer toujours quelque chose dans cette partie d'analyse).

On remarquera que les étudiants ont par ailleurs peu utilisé le système pour apprendre par eux-mêmes, même si ce n'était pas le but de QBLS. La stratégie basée sur les questions montre donc ses limites en terme de motivation des étudiants, par contre elle semble remplir tout à fait son rôle lorsque la question cible une connaissance précise.

4.2.2 Le point de vue de l'enseignant

L'expérience montre que la présence d'annotations sémantiques impose une certaine rationalisation à la fois du document de cours et dans la démarche de l'auteur. Ceci sans toutefois l'entraver au point de générer chez lui un refus de l'outil. Grâce à la méthode d'annotation que nous proposons (Dehors *et al.*, 2005) l'effort consenti ne semble pas constituer un obstacle majeur. L'expérience nous amène ensuite à évaluer la pertinence de ces annotations collectées de manière systématique. Grâce à l'utilisation du système en amphi, nous savons que l'enseignant a dû effectuer les tâches suivantes, pour lesquelles les annotations ne l'aidaient pas :

- prévoir un parcours prévisionnel des thèmes et des notions visitées durant le cours, selon une progression chronologique et logique, préparée sur un exemplaire du polycopié distribué (polycopié reprenant uniquement les thèmes du cours, et devant servir aux élèves comme support de notes).
- vérifier que les liens hypertextes implémentés permettent de décrire le parcours prévu, en ajouter ou les corriger sinon.
- renforcer la cohérence de l'exposé et éviter la lecture du texte des fiches d'information, en plaçant des figures pour schématiser les connaissances clés, au moins pour introduire les objectifs clés de la leçon, ce qui a été fait dans quatre thèmes sur six.

Pour faciliter la navigation entre les notions, il apparaît que les liens hypertextes vers les notions liées devraient être placés ou répétés dans une zone extérieure au texte, pour faciliter la navigation en particulier lors du cours.

S'il semble tout à fait souhaitable d'automatiser le plus possible l'acquisition et l'organisation des ressources pédagogiques dans QBLS, l'éditeur d'annotation de QBLS s'est révélé nécessaire pour la mise en ligne des solutions aux questions par

l'enseignant au cours du TD ainsi que pour apporter des mises à jour au contenu des ressources pédagogiques à l'issue d'une séance.

Les questions posées par les étudiants révèlent des précisions utiles à ajouter et collaborent ainsi à la conception de QBLS (par ex. signalement de la mauvaise visibilité d'un graphique, demande de définition d'un terme qui sera remplacé par un autre).

4.3 La valeur ajoutée du web sémantique

4.3.1 Apport de la démarche ontologique

Les apports de l'ontologie constatés dans cette expérience peuvent être abordés à la lumière de l'analyse de (Psyché *et al.*, 2003). Ainsi la définition d'une ontologie a permis le/la:

- Repérage des objets pédagogiques : le contenu sémantique de chaque ressource ou paragraphe a été décrit par rapport à l'ontologie.
- Acquisition des connaissances : il est clair que la définition de l'ontologie a permis de fiabiliser et d'automatiser le processus d'extraction des annotations à partir du document existant.
- Fiabilité : grâce à la formalisation introduite par l'ontologie, nous avons pu déterminer les notions du cours qui n'étaient jamais introduites ou référencées par les autres ressources avec une requête unique au moteur de recherche sémantique Corese. Cette vérification permet ensuite de garantir la cohérence de la navigation, en excluant toute possibilité de ne pas pouvoir atteindre une fiche.
- Maintenance des objets pédagogiques : L'ontologie a autorisé la mise en œuvre d'un outil d'annotation générique pour assurer la maintenance.
- Spécification d'une conceptualisation : la vision du domaine enseigné apparaît dans l'ensemble des instances des notions et des thèmes même si nous n'avons pas développé d'ontologie spécifique du domaine. La vision pédagogique quant à elle est clairement exprimée par l'ontologie.

La réutilisation et le partage des objets pédagogiques invoqués par (Psyché *et al.*, 2003) et caractéristiques de l'approche Web Sémantique doit se comprendre dans le cadre restreint de notre « web sémantique de formation », et se limite aux personnes participant au consensus ontologique, c'est-à-dire uniquement les enseignants chercheurs associés à cette expérience. On peut donc considérer que cet aspect est également présent à très faible échelle. L'utilisation du standard RDF permet d'envisager dans le futur d'étendre le web sémantique de formation à d'autres intervenants ou systèmes.

4.3.2 Apport de l'annotation sémantique des ressources pédagogiques

Un des principaux problèmes rencontrés et qui constitue un des points majeurs de ce bilan est le problème de la mise à profit par l'apprenant des aspects sémantiques introduits dans le cours.

L'interface a certes été facilement acceptée par les étudiants. Mais alors qu'elle reprend en partie la conception introduite dans l'ontologie (voir figure 1) il semble que les étudiants n'aient pas vraiment exploité les apports du modèle. Cette réflexion est basée sur l'observation du peu de clics sur les onglets (voir figure 4).

Lors du cours nous avons cependant constaté que les indices sémantiques rendus visibles (Définition, Exemples, etc.) permettent de structurer le discours, tout comme les liens hypertextes permettent de rationaliser la progression dans les concepts. Ainsi on ne se trouve pas en présence d'une juxtaposition de diapositives où le lien doit être fait oralement. Dans le cas présent les indices sémantiques semblent soutenir la logique de la progression (ex: "Nous allons maintenant voir un exemple" suivi d'un clic sur l'onglet exemple).

L'enseignant, bien que auteur des ressources pédagogiques de QBLS, a parfois été surpris par la fiche à laquelle il accédait en cliquant sur certains onglets. Cela met en évidence une possible inadéquation de certains termes visibles sur les onglets pour qualifier le contenu sémantique des ressources auxquelles on accède.

Bien que faiblement utilisés, le questionnaire montre que les étudiants ont cependant bien perçu le rôle des onglets et n'expriment pas de gêne vis-à-vis de leurs titres. Leur faible utilisation n'est donc a priori pas un problème lié à la sémantique des termes mais montrerait plutôt que les étudiants vont « au plus court » dans leurs recherches et finalement exploitent parfaitement l'outil pour cela.

5 Conclusion

Dans cette expérience nous avons mis en pratique des techniques du web sémantique pour l'accès à un contenu pédagogique. L'expérience montre la faisabilité de cette approche en se basant sur l'extraction des ressources à partir d'un document original. Les techniques d'extraction et de gestion des ressources à l'aide d'annotations sémantiques ont également permis de mettre en place une architecture légère et néanmoins capable de générer des interfaces dynamiques exprimant une certaine conceptualisation du domaine et impliquant une démarche pédagogique. L'ingénierie ontologique a permis de formaliser et facilement mettre en place cette démarche.

La maîtrise rapide du système par les étudiant montre les possibilités et les avantages d'introduire les aspects sémantiques dans l'interface et dans la navigation. La tâche demandée aux étudiants a dans l'ensemble été correctement réalisée ce qui nous conforte dans cette idée.

La faible utilisation du système en dehors des tâches de réponses et le peu de curiosité des étudiants montre l'importance de la démarche pédagogique dans de tels systèmes, qui sont très spécifiques et ne sont pas des systèmes à tout faire.

Pour que QBLS permette une compréhension de l'ensemble du cours, il faut donc que les questions de TD abordent tous les aspects du cours. Cette remarque souligne bien le changement de paradigme éducatif qui s'opère lorsqu'on utilise un mode d'accès à l'information aussi direct ; c'est ce que constatent également (Giardina & Oubenaïssa, 2003). Il n'y a plus d'apprentissage global par lecture complète d'un document puis application des connaissances emmagasinées. L'apprenant entre dans une démarche de recherche d'information, trouve cette information et l'applique immédiatement.

L'expérience a également mis en évidence le besoin de personnalisation invoqué par de nombreux travaux. Les parcours atypiques, et les différences de rythme entre les étudiants justifient maintenant d'envisager une adaptation à l'utilisateur.

Remerciements. A Leonid Synyukov, ergonomiste, pour sa collaboration dans la préparation et l'analyse des résultats de l'expérimentation de QBLS.

Références

- ABEL M.-H., BARRY C., BENAYACHE A., CHAPUT B., LENNE D., MOULIN C. (2004) Using an Organizational Memory for e-Learning, in Proceedings of *Workshop of Knowledge Management and Organizational Memories* (ECAI2004), Valencia, Spain.
- AZOUAOU F., DESMOULINS C. & MILLE D. (2003) Formalismes pour une mémoire de formation à base d'annotations: articuler sémantique implicite et explicite, in Acte du Colloque *Environnements Informatiques pour l'Apprentissage Humain* (EIAH2003), Strasbourg, France.
- BEAU D. (2005) La boîte à outils du formateur – 100 fiches pour animer vos formations, éditions d'Organisation, 1976 1996 2000 2002 2005.
- BERNERS-LEE T., HENDLER J., LASSILA O. (2001) The Semantic Web, *Scientific American*, May.
- BRUSILOVSKY P. (2003), Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools. In T. Murray, S. Blessing and S. Ainsworth Eds. : *Authoring Tools for Advanced Technology Learning Environment*. Dordrecht: Kluwer Academic Publishers, pp. 377-409.
- BRUSILOVSKY P. & VASSILEVA J. (2003) Course Sequencing Techniques for Large-Scale Web-Based Education, *International Journal of Continuing Engineering Education and Lifelong Learning*, 2003, vol 13, pp 75-94.
- CORBY O., DIENG R. ET FARON C. (2004) Querying the Semantic Web with the Corese Search Engine, in Proceedings of the *European Conference on Artificial Intelligence* (ECAI2004), pp 705-709.
- CRAMPES M. & RANWEZ S. (2000) Ontology-Supported and Ontology-Driven Conceptual Navigation on the World Wide Web, in Proceedings of the *ACM Conference on Hypertext and Hypermedia* (HT00), San Antonio, Texas, USA.
- DAHNI I., ARMBRUSTER M., FURBACH U. & SCHWABE G. (2001) Slicing Books – The Authors' Perspective, in R. Bromme, E. Stahl Eds. : *Writing Hypertext and Learning: Conceptual and Empirical Approaches*, Pergamon.
- DE BRA P., AERTS A., BERDEN B., LANGE B., ROUSSEAU B. (2003) AHA! The Adaptive Hypermedia Architecture, in Proceedings of the *ACM Conference on Hypertext and Hypermedia* (HT2003), Nottingham, UK.
- DEHORS S., FARON-ZUCKER C., GIBOIN A., STROMBONI J.P. (2005), Un web sémantique de formation par questionnement, In Actes de l'*Atelier Extraction et Gestion des Connaissances dans les EIAH* (EGC2005), Paris.

- DESMOULINS C., GRANDBASTIEN M. (2000). Des ontologies pour indexer des documents techniques pour la formation professionnelle, In Actes de la conférence *Journées francophones d'Ingénierie des Connaissances (IC2000)*, Toulouse.
- DIENG-KUNTZ R., CORBY O., GANDON F., GIBOIN A., GOLEBIOWSKA J., MATTA N. ET RIBIERE, M. (2001). Méthodes et outils pour la gestion des connaissances : Une approche pluridisciplinaire du Knowledge Management, Paris : Dunod.
- DUVAL E., FORTE E., CARDINAELS K., VERHOEVEN B., VAN DURM R., HENDRIKX K., WENTLAND-FORTE M., EBEL N., MACOWICZ M., WARKENTYNE K. & HAENNI F. (2001) The ARIADNE Knowledge Pool System, in *Communications of the ACM*, 44 (5), pp. 73-78.
- HEIWIY V., DUCATEAU C-F. (2003) Un modèle de ressources pédagogiques pour la FOAD, in Actes du colloque *Environnements Informatiques pour l'Apprentissage Humain (EIAH2003)*, Strasbourg, France.
- GIADINA M., OUBENAÏSSA L. (2003) Projet d'apprentissage/enseignement en ligne, in *Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF)*, Numéro spécial : Technologies et Formation à distance, Vol 10.
- MICHAU, F., PLOIX, S. (2003) Proposition de carte organisationnelle : décrire le rôle d'un environnement informatique au sein d'un dispositif pédagogique, in Actes du colloque *Environnements Informatiques pour l'Apprentissage Humain (EIAH2003)*, Strasbourg France.
- MIZOGUCHI R., IKEDA M. AND SINITSU K. (1997) Roles of Shared Ontology in AI-ED Research, Intelligence, Conceptualization, Standardization and Reusability, in *Proceeding of the International Conference on Artificial Intelligence in Education (AIED97)*, Kobe, Japan.
- MURRAY, T. (2003) MetaLinks: Authoring and Affordances for Conceptual and Narrative Flow in Adaptive Hyperbooks, in *Journal of Artificial Intelligence and Education*, Vol. 13 (Special Issue on Adaptive and Intelligent Web-Based Systems).
- PAQUETTE G., BOURDEAU J., HENRI F., BASQUE J, LEONARD M, MAINA M. (2003) Construction d'une base de connaissances et d'une banque de ressources pour le domaine du téléapprentissage, in *Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF)*, Numéro spécial : Technologies et Formation à distance, Vol 10.
- PSYCHE V., MENDES O., BOURDEAU J. (2003) Apport de l'ingénierie ontologique aux environnements de formation à distance, in *Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF)*, Numéro spécial : Technologies et Formation à distance, Vol 10.
- RIGAUX P. AND SPYRATOS N. (2003) Metada Management and Learning Object Composition in a Self e-Learning Network, in *Proceedings of Workshop on Information Search, Integration and Personalization*, Japan.
- STOJANOVIC L., STAAB S AND STUDER R. (2001) ELearning Based on the Semantic Web, in *Proceedings of the World Conference on the WWW and Internet (WebNet2001)*, Orlando, Florida, USA.
- STROMBONI J.P. (2002) Un cours introductif au traitement du signal à travers les applications audio de l'ordinateur multimédia, in *Proceedings of Technologies de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie (TICE'2002)*.
- TRICOT, A., PLEGAT-SOUTJIS, F., CAMPS, J.-F., AMIEL, A., LUTZ, G., MORCILLO, A. (2003) Utilité, utilisabilité, acceptabilité : interpréter les relations entre trois dimensions de l'évaluation des EIAH. In Actes du colloque *Environnements Informatiques pour l'Apprentissage Humain (EIAH2003)*, Strasbourg, France.
- ULLRICH C. (2004) Description of an Instructional Ontology and its Application in Web Services for Education, in *Proceedings of Workshop on Applications of Semantic Web Technologies for E-learning (SW-EL'04)*, Hiroshima, Japan.

Approche multidimensionnelle du texte pour le balisage des ressources pédagogiques

Isabelle Rondeau¹, Sylvie Ranwez² et Michel Crampes²

¹ CERIC, Université de Montpellier III – Paul Valéry,
3, Allée Doyen Nerson – BP 310, F-34 506 Béziers cedex
Isabelle.Rondeau@univ-montp3.fr
<http://www.ceric-mpl.org>

² Centre de Recherche LGI2P, Ecole des Mines d'Alès,
Site EERIE, Parc scientifique G. Besse F-30 035 Nîmes cedex 1
{Sylvie.Ranwez, Michel.Crampes}@ema.fr
<http://www.lgi2p.ema.fr/>

Résumé : Cet article propose d'appréhender le balisage des ressources pédagogiques selon les quatre dimensions du texte : la structure, la rhétorique, le contenu et la pragmatique. Nous posons ici que cette démarche, prémisses pour une méthodologie de balisage, facilite la conception d'interfaces pédagogiques multimodales tout en préservant l'intégrité du document. Nous appuyons notre réflexion sur un exemple d'application.

Mots-clés : Ressources pédagogiques, balisage, recomposition de documents, EIAH multimodal.

1 Introduction

Le Web sémantique tel qu'il a été défini par (Berners-Lee et al.2001) constitue une source d'inspiration pour bon nombre de chercheurs. De l'application commerciale à travers les Web services au e-learning, en passant par l'amélioration de la recherche d'information, cette multidisciplinarité élargit considérablement l'acception de "Web sémantique" et nécessite en permanence une redéfinition. Adoptant le point de vue des sciences de l'information et de la communication, le "Web sémantique" interroge la notion de document numérique. Les différentes approches du document numérique, issues majoritairement du domaine de l'informatique amènent certains chercheurs à œuvrer pour une théorie du document (Pédaque, 2005), afin de préserver sens et cohérence. Loin de nous l'intention d'entrer ici dans le débat engagé par Pédaque, nous reconnaissons néanmoins que le document numérique est en mutation et que sous cette poussée, il est nécessaire de poser les fondements et les rôles des cyber-institutions qui se feront le relais de ces documents : les bibliothèques numériques (Rondeau & Verlaet, 2005). L'article que nous proposons ici part d'un constat : l'enseignant en tant que médiateur du savoir, est politiquement encouragé à recourir au document numérique, à la fois comme ressource, mais aussi comme média pour dispenser son savoir. Cette utilisation passe en général par une phase d'adaptation du document au contexte d'enseignement. Par

adaptation nous entendons sélection de fragments pertinents et réorganisation selon une intention pédagogique et par "contexte d'enseignement", nous désignons le dispositif de formation dans lequel s'insère le document (par exemple dispositif de formation ouverte et à distance, dispositif e-learning, dispositif présentiel incluant des ressources numériques etc.), ainsi que le niveau d'étude. Or, si en situation présenteielle, l'adaptation est intrinsèquement possible, celle-ci est autrement plus complexe dans un univers pédagogique numérique. Le recours à des procédés techniques de sélection et recomposition plus ou moins supervisés rend nécessaire de penser le document de référence sous l'angle de la pragmatique pour conserver cohérence et sens. Dans la lignée des travaux menés sur les Documents Virtuels Personnalisables (DVP), le Web Pragmatique (Munindar, 2002) , et les hypermédias adaptatifs, nos recherches portent sur la conception d'EIAH flexibles à l'aide des deux outils du Web sémantique que sont XML et les ontologies. Dans cet article, nous proposons d'aborder le texte selon une approche multidimensionnelle pour baliser et composer un cours multimodal c'est-à-dire un cours dont le contenu peut être visualisé selon différents axes.

Nous présentons tout d'abord l'enseignement à partir duquel nous menons cette recherche, puis nous exposons les principes de l'approche multidimensionnelle nécessaire au balisage pour, enfin, justifier l'adoption d'une telle démarche.

2 La problématique de l'enseignement de l'analyse situationnelle

L'enseignement qui fait l'objet de cette recherche porte sur les sciences de l'information et de la communication ; le support de cet enseignement est intitulé "Le sens dans la communication en situation" et destiné à des étudiants de "Mastère 2". Ce corpus est composé de 41 articles chacun définissant un concept de l'Analyse Situationnelle (AS) des communications. Il constitue un document de référence dans la discipline ; il définit chaque concept, propose d'approfondir la notion dans un "développement", l'illustre par un ou plusieurs exemples et propose de le situer par rapport aux autres concepts dans un paragraphe dédié "liens sémantiques avec d'autres concepts". Très théorique, ce cours pose la question de sa mise en ligne. Comment penser la numérisation d'un document essentiellement textuel et iconique de sorte que ce passage au digital présente une valeur ajoutée comparativement à sa version papier ? Comment faciliter l'appropriation de ce contenu dans sa version numérique ?

De notre point de vue, le document doit proposer un mode de lecture distinct de la lecture sur support papier. La première raison est physiologique : l'écran est un support de lecture difficile. On lit plus lentement à l'écran que sur papier. Par ailleurs, rechercher une spécificité au document textuel numérique c'est rechercher l'essence même de ce support. Nombre de documents scientifiques sont publiés en ligne dans des formats PDF et RTF qui ne se justifient que par la diffusion mondiale qu'ils permettent.

Pour répondre à cette attente et justifier le recours au numérique, nous proposons donc d'utiliser l'annotation XML sur des documents textuels numériques existants pour guider le lecteur dans sa consultation. Notre démarche consiste à baliser le document puis à en produire différents rendus via la mise en œuvre de feuilles de style XSL-T. La spécificité de notre approche est d'aborder le balisage et la recomposition de documents d'après les quatre dimensions du texte que nous présentons ci-dessous.

3 Approche multidimensionnelle du balisage

Qu'il s'agisse d'indexation conceptuelle (Prié, 2000), de balisage ciblé (Rey & Zaoui, 2004) ou d'annotation sémantique (Desmontils & Jacquin, 2002), la littérature ne préconise, à notre connaissance, aucune méthodologie particulière pour le balisage d'un document. Or en supposant que l'échange et la réutilisation des ressources pédagogiques trouvent un terrain d'application, cette réutilisation ne peut reposer que sur une forme d'annotation sémantique partagée, prenant appui sur des formalismes communs. Pour avancer vers une méthode généralisée d'annotation sémantique, nous proposons une approche ontologique multidimensionnelle.

3.1 Les quatre dimensions d'un texte

Il existe au moins autant de façons d'annoter un document qu'il existe de lecteurs de ce document et d'intentions de lecture. Marc et Jocelyne Nanard soulignent l'importance de l'*intention* dans tout processus d'annotation sémantique (Nanard & Nanard, 2004), de sorte qu'on pourrait parler "d'annotation pragmatique". Cette intention décide souvent de la nature de la ressource qui va entrer dans la composition d'un document, et également dirige fortement le rôle qu'elle va jouer dans ce document. Dans (Ranwez & Crampes, 2002) ce rôle est appréhendé en termes de *rôle pédagogique*. Une même citation peut servir à la fois d'exemple et de contre exemple selon le contexte dans lequel elle est utilisée. L'*usage*, ou pragmatique en linguistique, est donc de notre point de vue une dimension intrinsèque du document. Mais il n'est pas seul. Au fil de l'histoire, le document s'est entouré de normes facilitant sa compréhension. Ainsi, le document écrit sur support papier répond à une norme que l'on peut qualifier de *structurelle*. L'article scientifique, par exemple, comporte un titre, un résumé, des sous-titres, une introduction, des sections et sous-sections, une conclusion, une bibliographie... Le poème, quant à lui, répond à d'autres critères : il est composé de vers et de strophes composant ou non un sonnet mais le genre est identifiable par ses caractéristiques éprouvées au fil des siècles.

Par ailleurs, la *rhétorique*, se retrouve dans la plupart des textes (Perelman & Olbrechts-Tyteca, 1983). Selon la nature du texte, elle est utilisée à différents niveaux de sophistication et de complexité. Les formes rhétoriques les plus souvent utilisées en pédagogie sont l'analogie, l'énoncé de faits ou de vérités, l'exemple, la

démonstration logique (mathématiques), la preuve par l'expérience (sciences physiques), et l'autorité (philosophie). La rhétorique est la troisième dimension d'un document écrit. Enfin, la quatrième dimension du document, et non la moindre, concerne **le contenu**. Le contenu renvoie aux différents aspects du sujet dont traite le document, c'est-à-dire à un ensemble de concepts et de relations entre concepts spécifiques à un domaine et formalisables dans une ontologie

Ces quatre dimensions – pragmatique, structurelle, rhétorique et contenu, sont concomitantes. Qu'on se place dans une position de lecture cursive ou de lecture analytique, chacune de ces dimensions est interrogée à des degrés variables. Un lecteur face à un livre va regarder le titre, l'auteur, la quatrième de couverture, le sommaire, l'index pour voir si l'ouvrage répond à ces attentes : c'est donc par la dimension *structurelle* qu'il interroge le document. Si cette première manipulation du livre lui laisse entendre qu'il peut répondre à ce qu'il cherche (*dimension pragmatique*), c'est que cette structure est déjà porteuse d'un sens (*dimension contenu*). Ensuite, il le parcourt, à la recherche de précisions, d'arguments, d'exemples : il interroge ici la *dimension rhétorique* de l'ouvrage.

Partant de ce constat, on peut s'essayer à une analyse de document selon les quatre axes cités précédemment, pour ensuite dégager une méthode d'annotation sémantique.

3.2 Application

Nous proposons ici, dans la table 1, d'illustrer cette approche par son application à l'analyse situationnelle des communications que nous souhaitons enseigner aux étudiants de Master 2.

Table 1. Grille d'analyse d'un document en fonction des quatre dimensions : structure, contenu, pragmatique et rhétorique

<i>Structure</i>	<i>Contenu</i>	<i>Pragmatique</i>	<i>Rhétorique</i>
Titre de l'ouvrage, auteur, éditeur, date de publication, sommaire, chapitres, titre de l'article, section, sous-section, bibliographie, schéma, tableau.	Mots-clés issus de l'ontologie du domaine (<i>par exemple "élément significatif de la situation", "situation pour un acteur", "sens" etc. mais aussi les auteurs qui s'inscrivent dans le domaine</i>),	prendre connaissance (exposition), pour pratiquer (exercice), pour situer, pour comprendre, pour discuter.	Argument; argument d'autorité; exemple ; contre exemple ; comparaison ;

À partir de là, plusieurs possibilités de balisage en xml s'offrent à nous. La première consiste à désigner les éléments xml par le nom du domaine duquel il relève ; il n'y a alors que quatre éléments (<structure>, <rhétorique>, <contenu>,

<pragmatique>). Pour un article de l'analyse situationnelle, nous obtenons le balisage suivant :

```
<contenu concept="communication paradoxale">
  <structure niveau="titre"> COMMUNICATION PARADOXALE </structure>
  <structure niveau="sous-section">
    <structure niveau="sous-titre"> Définition </structure>
    <rhétorique type="définition">
      texte de la définition
    </rhétorique>
  </structure>
  ...
  <structure niveau="sous-section">
    <structure niveau="sous-titre"> sous-titre Exercice </structure>
    <pragmatique type="exercice">
      énoncé de l'exercice
    </pragmatique>
    <pragmatique type="corrigé">
      corrigé de l'exercice
    </pragmatique>
  </structure>
</contenu>
```

Mais cette approche est limitée. En effet, on constate ici par exemple, qu'il n'y a pas de moyen de typer plus avant l'exercice. Il serait incorrect d'ajouter un deuxième attribut à l'élément pour qualifier cet exercice car il serait non pas un attribut de l'élément, mais un attribut d'attribut. Pour pouvoir aller plus avant dans l'annotation sémantique, la seconde solution consiste à créer autant d'éléments que de concepts se rapportant à la dimension, et d'attributs permettant de renforcer l'annotation. Mais alors il faut donner une indication d'appartenance à la dimension.

Nous disposons de quatre dimensions qui peuvent être décrites par des ontologies. Dès lors la démarche la plus appropriée nous semble être la suivante :

```
<cont:concept nom="communication paradoxale">
  <struc:titre> COMMUNICATION PARADOXALE </struc:titre>
  <struc:sous-section>
    <struc:sous-titre> Définition </struc:sous-titre>
    <rh:definition type="référence" auteur="Paul Watzlawick">
      texte de la définition
    </rh:definition>
  </struc:sous-section>
  ...
  <struc:sous-section>
    <struc:sous-titre> Exercice </struc:sous-titre>
```

```
<prag:exercice niveau="débutant">
    énoncé de l'exercice
</prag:exercice>
<prag:corrige>
    corrigé de l'exercice
</prag:corrige>
</struc:sous-section>
</cont:concept>
```

Par l'intermédiaire d'espaces de noms référant aux quatre domaines, l'élément est situé dans sa dimension, puis nommé précisément. Cette stratégie de balisage nous semble plus légitime car on voit bien ici qu'elle permet de décrire le fragment plus précisément par le biais d'une annotation plus cohérente. Les attributs se rapportent bien à l'élément, et chaque élément est renvoyé à son espace de nom¹ (son ontologie) via son préfixe.

Dans l'exemple ci-dessus, on notera que les balises apparaissent dans un certain ordre hiérarchique. La première raison tient au fait que la structure d'un document XML est un arbre, et le document ainsi balisé présente nécessairement une telle structure. L'ordre apparent est donc d'abord un ordre formel. Mais il est légitime de se poser la question de l'existence d'une hiérarchie d'organisation des classes de balises qui ferait sens au-delà du formel. Ceci revient à donner une signification à la stratégie d'organisation du schéma (ou de la DTD) de balisage. Comme le balisage correspond à une fragmentation potentielle du texte en vue de sa recombinaison pour un lecteur dans un contexte de lecture particulier, le schéma de balisage doit être construit en pensant aux angles de lecture possibles. Il correspond à un tamis qui définit les vues du texte attendues par le lecteur. Intuitivement, la première vue qui surgit est celle du contenu. Un lecteur lit un texte pour en saisir le contenu. C'est ce qui nous fait placer le contenu en premier, les autres balises relevant d'une forme d'expression du contenu, en particulier dans sa composante 'rhétorique'. Mais ce point reste encore à discuter à partir d'observations multiples de différents textes et de leur recombinaison personnalisée dans un contexte pédagogique.

3.3 Exploitation pédagogique du balisage multidimensionnel

Les usages d'un document se traduisent dans différentes combinaisons de fragments. On passe ici de l'annotation à la composition. L'apport de l'approche multidimensionnelle est d'ordre méthodologique. La décomposition et l'étude selon les quatre dimensions vont aussi servir à la composition multimodale du document. La présentation va inciter le lecteur à lire d'une certaine manière et en cela relève de l'acte de langage. L'étape suivante va alors consister à établir **des combinaisons entre les différents fragments et leurs correspondances avec des usages**. Par exemple, dans notre interface, nous proposons de ne comprendre un concept que par

¹ <http://www.w3.org/TR/REC-xml-names/>

la lecture de sa définition assortie d'un exemple. Ce mode de lecture répond au besoin du débutant qui cherche à comprendre et de l'averti qui veut simplement se remémorer le concept dans les termes mêmes définis par l'enseignant auteur. Nous avons développé 5 modes de lecture qui reposent tous sur la recombinaison : un mode de lecture "définition", un deuxième "définition et exemple(s)", un troisième "article complet", un quatrième "liens sémantiques entre les concepts" et enfin, un cinquième "concepts et auteurs de référence". Ces modes de lecture du document sont associés à un dispositif d'entraînement à l'analyse situationnelle, constitué de différents exercices ; de nombreuses autres combinaisons multidimensionnelles sont bien sûr possibles. Elles seront explorées ultérieurement.

L'axe pragmatique se traduit donc ici par des modes d'exploitation (sélection, recombinaison) des autres axes selon les situations d'usage pédagogique ce qui signifie qu'au-delà du document d'origine, la transformation est une pragmatique dont la présentation (au sens de mise en forme) est le vecteur.

4 Discussion et perspectives

L'initiative de l'approche multidimensionnelle demande à être complétée par des modèles ontologiques de chacune des quatre dimensions pour permettre un meilleur partage de l'indexation. Bien qu'elle soit un processus coûteux, l'indexation présente de nombreux avantages que nous avons présentés dans "qualité d'une indexation" (Crampes et al., 2003). Nous insistons ici sur le principe d'économie. L'enseignant, en repérant et en qualifiant les fragments par l'intermédiaire du balisage, rend le document malléable, prêt à différents usages, pour différents utilisateurs. Il faut entendre "économie" ici comme une optimisation du contenu pour l'apprentissage. À travers la recombinaison de document, techniquement traduite par des feuilles de style XSL-T, l'enseignant fait un acte de langage. Extraire d'un texte et recomposer est une incitation pour l'apprenant à lire ce qui lui est proposé, non pas dans sa structure, ni dans sa mise en scène ou dans son contenu mais dans une autre position structure-rhétorique-contenu voulue par le "recompositeur". En réponse aux opposants de l'approche fragmentaire du document numérique recomposé (Pédauque, 2005), nous posons que l'approche multidimensionnelle permet à la fois de préserver l'intégrité des documents d'origine et de servir de support à l'édification d'une cohérence du document recomposé, à la construction d'un sens en situation.

Le balisage multidimensionnel n'est pas le seul porteur de cette construction. Le moteur de recombinaison en est aussi un élément fondateur. Sa traduction technique est la feuille de style XSL-T. Le chantier est donc double. D'une part clairement poser les ontologies de balisage pour expliciter les dimensions du document d'origine et en conserver l'intégrité tout en suggérant ses usages multiples. D'autre part traduire l'acte de déplacement de situations de lecture (de déplacement pragmatique) par des formes d'écriture de transformations, en l'occurrence des formes d'écriture de feuilles de style. Nous situons l'enjeu du document numérique, qui par essence est transformable à l'inverse du document papier, dans une 'stylistique' de la

transformation selon des intentions. Transformer un ou des documents est un 'acte de langage' puisqu'il déplace les documents selon l'axe pragmatique.

En corollaire à la transformation, l'annotation sous la forme du balisage permet de fournir les éléments qui guident la transformation. En ce sens, baliser est déjà un acte de langage qui, paradoxalement, n'est pas encore énoncé et dont on ne connaît pas encore les modes d'énonciation. C'est là toute sa difficulté.

Références

BERNERS-LEE T., HENDLER J. & LASSILA O. (2001), The Semantic Web, in The Scientific American.

http://sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

CRAMPES M., RANWEZ S., PLANTIE M. & VAUDRY C. (2003). Qualités d'une indexation portée par XML et une ontologie au regard d'un standard, in *Sciences et techniques éducatives*, Hors série 2003, "Ressources numériques, XML et éducation", Eric Bruillard et Brigitte de La Passardière eds., Hermès/Lavoisier, p.105-134

DESMONTILS E. & JACQUIN C. (2002). Annotation sur le web: notes de lecture. Journées de l'AS-CNRS Web sémantique.

<http://www.lalic.paris4.sorbonne.fr/stic/octobre/programme0209.html>

MUNINDAR P. SINGH, (2002), The Pragmatic Web: Preliminary Thoughts. NSF-OntoWeb Workshop on Database and Information Systems Research for Semantic Web and Enterprises. Stohr, E. T. and J. V. Nickerson.

<http://lsdis.cs.uga.edu/SemNSF/Singh-Position.pdf>

NANARD M. & NANARD J. (2004). Vers une spécification par intention de la production de documents, actes de la journée d'étude Intelligence collective, Partage et redistribution des savoirs, CERIC-LGI2P, Nîmes, 29 et 30 sept.2004.

PERELMAN CH., OLBRECHTS-TYTECA L. (1983) *Traité de l'Argumentation : la Nouvelle Rhétorique*, Editions de l'Université de Bruxelles, 4ième édition.

PRIE Y. (2000). Sur la piste de l'indexation conceptuelle de documents – une approche par l'annotation. In *L'indexation, Document Numérique Vol 4. N°1-2*, J.M. Jolion eds., Hermès.

RANWEZ S. & CRAMPES M. (2002). Instanciation d'Ontologies Pondérées et Calcul de Rôles Pédagogiques - Principe et mise en œuvre. In *Sciences et Techniques Educatives*, Volume 9, N°3, Hermès/Lavoisier, p.341-370.

REY C. & ZAOUÏ C. (2004), Le balisage XML ciblé : une nouvelle approche dans l'informatisation des corpus, actes CIFT 2004, p.121-123.

http://www.up.univ-mrs.fr/delic/perso/rey/cift_2004.pdf

R.T.PEDAQUE, (2005), Le texte en jeu - Permanence et transformations du document, Document de travail - version provisoire,

<http://rtp-doc.enssib.fr/IMG/doc/Erasme-V2-2.doc>

RONDEAU & VERLAET, (2005), The New Digital Documents: Which Perspectives For Digital Libraries? Actes de la conférence LIDA 2005, Dubrovnik, Croatie, juin 2005 (à paraître).

Un outil pour la qualification de ressources à base d'ontologies

Baruk Tolédano – Monique Baron – Hélène Giroire

Laboratoire d'Informatique de Paris 6,
Université Pierre et Marie Curie
4 place Jussieu, F-75252 Paris Cedex 05,
{prenom.nom}@lip6.fr

Résumé : La recherche et la réutilisation de ressources pédagogiques posent des problèmes d'indexation spécifiques. Nous allons présenter dans cet article les principes de conception d'un outil permettant de qualifier des ressources pédagogiques. Cette qualification, dont certains champs peuvent prendre leur valeur dans des ontologies, doit respecter un schéma de métadonnées. Cet outil aide l'utilisateur à respecter un schéma de métadonnées fourni en entrée et à choisir des valeurs provenant d'ontologies de façon transparente, lors de l'édition d'une qualification d'une ressource pédagogique.

Mots clés : Métadonnées, qualification, ressources et objets pédagogiques, ontologie, schéma de métadonnées, éditeur XML, DTD, LOM, gestion de contenus numériques.

1 Introduction

Les pratiques dans le monde éducatif ont beaucoup changé ces dernières années, notamment par l'utilisation des TIC¹ qui permettent le développement de ressources électroniques et leur utilisation lors d'activités d'apprentissage en présence ou à distance. Ce nouvel apport a permis de compléter, voire de remplacer, certains dispositifs et outils liés à l'enseignement, que ce soit au niveau collectif (classes en réseau et/ou connectées à Internet) ou au niveau individuel avec entre autres l'utilisation de « cartables électroniques » physiques (tablettes graphiques, e-manuels) ou virtuels (sur Internet).

On a donc des pratiques d'enseignement et d'apprentissage de plus en plus instrumentées par des outils informatiques et des ressources de plus en plus nombreuses. Mais l'usage de ressources pose des problèmes d'accès, de gestion, de réutilisation, d'intégration (aussi bien au niveau technique qu'au niveau pédagogique), ce qui nécessite la conception d'outils adaptés.

L'indexation des ressources via des informations qui les décrivent est une solution à une partie de ces problèmes, visant à faciliter le partage et la réutilisation de celles-ci. Cette approche nous a amenés à concevoir un outil dédié à des auteurs ou à des enseignants, pour qualifier des ressources (ou objets) pédagogiques selon un schéma de métadonnées quelconque. Afin d'améliorer la qualité d'indexation des ressources, nous proposons à l'utilisateur, pour certains descripteurs, la possibilité de choisir des valeurs provenant d'ontologies.

Nous présenterons d'abord succinctement les notions d'objet pédagogique et de schémas de métadonnées (section 2). Après un bref aperçu d'autres travaux d'indexation à base d'ontologies (section 3), nous présenterons notre approche (section 4) en développant la fonctionnalité spécifique de l'éditeur existant à savoir la gestion de valeurs provenant d'ontologies. Nous terminerons cette section par une brève comparaison avec d'autres éditeurs, avant de conclure.

¹ TIC : Technologies de l'Information et de la Communication.

2 Qualification d'Objets Pédagogiques

2.1 Notion d'objet ou de ressource pédagogique

D'après (LTSC-IEEE, 2001), un « learning object », traduit habituellement par « objet pédagogique » ou « ressource pédagogique » est une « entité, numérique ou non, qui peut être utilisée pour l'éducation, la formation ou l'apprentissage » (traduction AILF²).

Cette définition, assez floue, permet différentes interprétations suivant les auteurs, les contextes d'utilisation (Bourda, 2001) et les standards utilisés. Une idée transversale est la notion de granularité : les ressources sont souvent vues comme construites par agrégations successives allant du simple document atomique (un fichier audio, par exemple) à des modules de formation. Plusieurs classes d'objets pédagogiques sont distinguées dans (Pernin, 2003) : les unités pédagogiques, les activités pédagogiques et les ressources pédagogiques.

2.2 Schémas de métadonnées pour des objets pédagogiques

Le partage et la réutilisation de ressources ou d'objets pédagogiques, qui sont des objectifs à atteindre, nécessitent des moyens d'accès à des ressources pertinentes, dans un contexte et pour des besoins particuliers. Différentes approches pour la recherche de ressources ont été mises en œuvre. L'approche basée sur la recherche d'informations dans la ressource même est limitée d'une part, par l'absence de certaines informations qui ne sont généralement pas contenues dans la ressource (par exemple des informations techniques, pédagogiques ou liées aux coûts d'utilisation) et d'autre part, par la nature multimédia des ressources : la plupart des ressources de type image ou son ne contiennent pas d'information textuelle. Une approche par métadonnées semble alors plus indiquée dans notre contexte. De nombreux travaux dans le domaine de l'éducation ont conduit à l'élaboration de schémas conceptuels de métadonnées, comme le Dublin Core Education et le standard IEEE LOM. Ces schémas ont été largement décrits et comparés dans la littérature (CREPUQ, 2002), (De La Passardière & Grandbastien, 2003), (Pernin, 2003). Parmi ces schémas, le LOM semble être le plus abouti ; il est constitué de plus de 70 éléments répartis en neuf catégories. Des travaux en cours au sein de l'AFNOR portent sur la création d'un profil d'application français du LOM.

Cependant, certains champs (du LOM) peuvent être renseignés librement, par exemple, les mots-clés (LOM 1.5), la couverture (LOM 1.6) ou la description pédagogique (LOM 5.10). Ceci entraîne l'utilisation de vocabulaires divers qui risquent de ne pas être largement partagés et de poser problème pour la recherche ultérieure des ressources. L'approche que nous préconisons est d'utiliser des vocabulaires conceptuels issus d'ontologies partagées par des communautés de pratique. Nous donnerons des exemples d'approches similaires dans le point suivant.

3 Travaux liés à l'indexation à base d'ontologies

L'utilisation d'ontologies dans un contexte d'indexation s'est développée ces dernières années, dans différents domaines : dans le domaine médical (projet CISMéF (Thirion & Darmani, 1999)), dans le domaine audiovisuel (Bachimont & Dechilly, 2003), (Crampes, 2002), ou encore dans le domaine de la formation. Dans ce dernier, plusieurs travaux sont basés sur l'idée générale d'indexer des fragments de documents sur la base d'ontologies de natures différentes (ontologies de structure documentaire, ontologies de domaine, ontologies pédagogiques) en vue de les réutiliser pour composer plus ou moins automatiquement des nouvelles ressources de formation : le projet IMAT³ (Desmoulins & Grandbastien, 2002), le projet SIMBAD (Duitama et Al., 2003) ou dernièrement la conception d'un EIAH par questionnement (Dehors et Al., 2005). Des travaux plus généraux visent l'auto-composition de documents ou l'élaboration de documents virtuels personnalisables (Ranwez & Crampes, 1999), (Iksal, 2002).

² AILF : Association des Informaticiens de Langue Française (www.ailf.net)

³ IMAT : Integrating Manual And Training (<http://imat.swi.psy.uva.nl/>)

4 Conception d'un outil de qualification de ressources à base d'ontologies

4.1 Principes généraux

Nous avons voulu créer un outil ouvert permettant à des concepteurs ou à des enseignants de qualifier des ressources pédagogiques par des métadonnées, éventuellement de manière partielle. L'absence de norme et la diversité des standards encore en évolution nous ont incités à concevoir un outil qui ne soit pas dépendant d'un schéma de métadonnées particulier. Par ailleurs, cet outil doit permettre de choisir de façon simple des valeurs provenant d'ontologies diverses (générales, régionales et pédagogiques).

En effet, l'utilisation d'ontologies, comme celle de thésaurus, de dictionnaires et de classifications permet de :

- fixer un vocabulaire commun permettant une meilleure communication entre les différents utilisateurs des ressources,
- créer des qualifications des ressources précises grâce à un vocabulaire riche dont les termes sont ordonnés sémantiquement,
- proposer à l'utilisateur de choisir des valeurs dans une liste donnée, ce qui évite la saisie au clavier (et le risque de fautes de frappe) lors des phases d'édition des métadonnées et de recherche de ressources.

En outre, l'utilisation d'ontologies permet lors d'une recherche de trouver des ressources en « élargissant » la requête de l'utilisateur, grâce à un mécanisme d'inférence qui exploite les relations sémantiques entre les concepts de l'ontologie afin de permettre le remplacement des termes de la requête par des concepts sémantiquement liés à ces derniers (voir aussi l'approche « médiateur » dans les bases de données comme le projet PICSEL (Reynaud, 2004)).

C'est sur cette base générale qu'a été développé OBGeXE (Ontology Based Graphical XML Editor), un éditeur XML graphique à base d'ontologies. Il permet à un utilisateur de saisir ou de compléter les métadonnées (au format XML) d'une ressource en respectant un schéma de métadonnées donné en entrée (au format DTD). Certains champs peuvent prendre des valeurs dans des ontologies, également données en entrée (**Fig. I**).

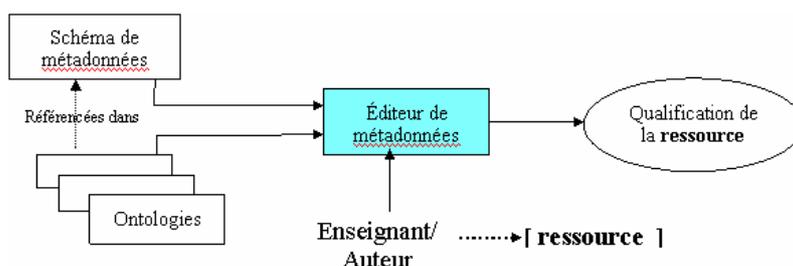


Fig. I – Entrées / Sorties de l'éditeur de métadonnées de ressources pédagogiques.

4.2 Un aperçu de l'interface

L'interface graphique de l'éditeur (**Fig. II**) est composée de quatre panneaux ; celui de droite est dédié à l'édition des métadonnées, celui du milieu haut à la visualisation du schéma de métadonnées, celui de gauche à la visualisation d'ontologies, celui du milieu bas, entre autres, à la visualisation de l'aide.

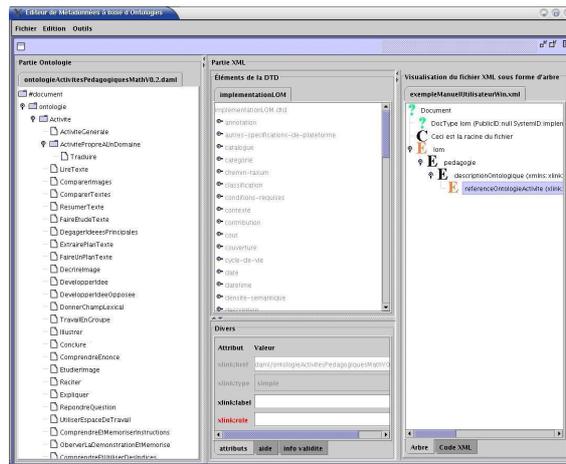


Fig. II – Aperçu de l'interface graphique de l'éditeur.

Une ontologie peut se représenter sous la forme d'un graphe dont les sommets sont les concepts de l'ontologie, reliés par des arcs représentant les relations entre ces concepts. Il existe pour certaines relations, comme la relation de spécialisation « Est-un », un sous-graphe associé qui est un arbre couvrant du graphe⁴.

C'est ce type de sous-graphe que nous utilisons car il permet de représenter facilement à l'écran, sous la forme d'un arbre, l'ensemble des concepts de l'ontologie ordonnés par cette relation (Fig. II). L'utilisateur peut alors facilement choisir le concept qui lui semble le plus adapté.

4.3 Le lien entre ontologies et schémas de métadonnées

Un problème que nous avons rencontré pour la réalisation de l'éditeur est le suivant : il n'est pas prévu dans la spécification XML du W3C, i.e. le langage de DTD, d'utiliser des ontologies comme domaine de valeur d'élément ou d'attribut XML, valeurs qui pourraient être vérifiées lors de la validation du fichier XML.

Comment spécifier alors qu'un élément XML doit prendre une valeur provenant d'une ontologie ? Une idée serait d'étendre le langage des DTD pour créer un nouveau type de données, par exemple :

```
<!ATTLIST elem valeur ONTOLOGY "http://www.monsite.com/monOntologie.owl">
```

Le mot clé 'ONTOLOGY' préciserait que l'attribut *valeur* de l'élément *elem* doit prendre sa valeur dans l'ontologie qui se trouve à l'URL <http://www.monsite.com/monOntologie.owl>. Mais ceci nécessiterait une modification des standards existants et le développement de nouveaux outils (un nouvel analyseur syntaxique).

La solution que nous proposons, inspirée de (Hunter & Lagoze, 2001), est une solution pratique qui respecte les standards actuels : nous fixons un couple d'attributs qui jouent le rôle de l'extension évoquée ci-dessus. Nous l'illustrons par un exemple dans le point suivant.

4.4 Exemple de lien entre DTD et ontologie.

L'exemple qui suit (Fig. III) concerne un extrait d'une implémentation du LOM sous forme de DTD. Nous nous intéressons à l'élément <description> de la catégorie 5 *Pédagogie*. Le rôle de cet élément est « explications relatives à l'emploi de l'objet pédagogique » (traduction AILF).

```
<!ELEMENT lom (general?, ..., pedagogie?, droits?, ...)>
```

⁴ Les concepts les plus abstraits de chaque composante connexe sont reliés à un super concept, considéré comme le concept le plus abstrait du graphe, ce qui nous permet de nous ramener à un graphe à une seule composante connexe.

```

<!ELEMENT pedagogie (type-d-interactivite?, ..., temps-d-apprentissage-type?,
description?, langue*)>
<!ELEMENT description (#PCDATA)>

```

Fig. III - Définition de l'élément <description> de la catégorie 5 *Pédagogie* du LOM.

Cette définition de l'élément <description> permet de renseigner librement les objectifs pédagogiques (d'où le type de valeur #PCDATA).

Afin de fixer le vocabulaire en utilisant une ontologie d'activités pédagogiques, nous allons utiliser à la place de l'élément <description>, l'élément <descriptionOntologique>, défini ci-dessous (**Fig. IV**)

```

<!ELEMENT pedagogie (type-d-interactivite?,..., temps-d-apprentissage-type?,
descriptionOntologique?, langue*)>
<!ELEMENT descriptionOntologique (referenceOntologieActivite*)>
<!ELEMENT referenceOntologieActivite EMPTY>
<!ATTLIST referenceOntologieActivite
xlink:href CDATA #FIXED
"http://www.monsite.com/ontologieActivitesPedagogiques.daml"
xlink:role CDATA #REQUIRED ...>

```

Fig. IV - Nouvelle définition de l'élément <description>, qui fait référence à une ontologie.

L'élément <descriptionOntologique> n'est plus du type #PCDATA (c'est-à-dire une chaîne de caractères) mais contient des éléments <referenceOntologieActivite> dont l'un des attributs, `xlink:href`⁵, contient l'adresse de l'ontologie d'activités pédagogiques et dont l'autre l'attribut, `xlink:role`, permet de mémoriser le concept qui sera choisi comme valeur (**Fig. V**).

```

...<descriptionOntologique>
  <referenceOntologieActivite xlink:role="#ResumerTexte"/>
  <referenceOntologieActivite xlink:role="#CommenterTexte"/>
</descriptionOntologique>...

```

Fig. V - Qualification modifiée contenant des références vers l'ontologie d'activités pédagogiques.

Nous pouvons ainsi constituer un schéma de métadonnées dont certains descripteurs sont liés à des ontologies comme domaine de valeur.

Lors de l'édition d'une qualification, OBGEXE détecte dynamiquement les éléments qui prennent leur valeur dans une ontologie. Pour un tel élément, il présente alors l'ontologie correspondante sous la forme d'un arbre, dans lequel l'utilisateur choisit, de façon transparente, un concept.

4.5 OBGEXE et les autres éditeurs XML

Il existe de nombreux éditeurs XML, pourquoi en avoir conçu un nouveau ? A notre connaissance, aucun autre éditeur ne propose une interface unique pour renseigner des métadonnées en respectant un schéma de métadonnées quelconque et choisir des valeurs provenant d'ontologies.

Il existe des éditeurs XML « ouverts » comme XMLSPY⁶ ou <oxygen/>⁷ qui permettent de saisir des métadonnées simples respectant un schéma de métadonnées quelconque mais sans utiliser d'ontologies, et des outils spécifiques comme Metasaur (Kay & Lum, 2003) ou SHAME⁸.

Metasaur permet d'indexer une ressource sous la forme d'une liste plate de valeurs qui sont des concepts provenant d'une (unique) ontologie. Il n'y a aucun schéma de métadonnées. A l'inverse, SHAME permet de saisir des métadonnées en RDF suivant un schéma de métadonnées choisi parmi une liste fermée (entre autre Dublin Core, LOM) et dont la structure ne peut pas être modifiée. Par ailleurs, la gestion des ontologies dans SHAME est quasi inexistante, puisque

⁵ Recommandation XLink du W3C (<http://www.w3c.org/TR/xlink/>, spécification des liens standards sur le Web.

⁶ XMLSPY (http://www.altova.com/products_ide.html)

⁷ <oxygen/> : <http://www.oxygenxml.com/>

⁸ SHAME: Standardized Hyper Adaptable Metadata Editor (<http://kmr.nada.kth.se/shame/index.html>)

l'éditeur ne propose que le choix d'une ontologie dans une liste, sans en présenter les concepts à l'utilisateur.

OBGeXE combine les possibilités de ces différentes classes d'éditeur permettant de rester à la fois adaptable à n'importe quel schéma de métadonnées. Il est donc adaptable à n'importe quel domaine d'application.

5 Conclusion

Notre objectif était de montrer la faisabilité d'un outil convivial destiné à instrumenter l'activité de qualification de ressources selon une approche à base d'ontologies. Nous avons conçu et réalisé un éditeur graphique XML ouvert et général, permettant d'utiliser des ontologies comme vocabulaire conceptuel pour des métadonnées, ontologies qui doivent être largement partagées par des communautés de pratique.

Nous devons à présent tester le prototype sur le public visé afin d'avoir un retour d'expérience qui nous permette de tester l'utilisabilité et d'améliorer l'ergonomie de notre outil. Par ailleurs, nous avons commencé à spécifier un moteur de recherche sémantique basé sur les mêmes principes d'utilisation d'ontologies pour indexer des ressources.

Références

- BACHIMONT B. & DECHILLY T. (2003). Une ontologie pour éditer des schémas de description audiovisuels, extension pour l'inférence sur les descriptions. In *Ingénierie des Connaissances*, (Tome 2), à paraître
- BOURDA Y. (2001). Objets pédagogiques, vous avez dit objets pédagogiques ? *Cahier de Gutenberg* p. 39-40 - *Actes de GUT2001*, Metz mai 2001.
- CRAMPES M. (2002). Auto-Composition Active et émergence du sens dans l'interaction Homme-Machine sous contrainte, *Habilitation à diriger des recherches, Université de Montpellier II*, mai 2002.
- CREPUQ (2002). Les normes et standards de la formation en ligne : État des lieux et enjeux, étude réalisée par le Groupe de travail sur les normes et standards de la formation en ligne pour la Conférence des recteurs et des principaux des universités du Québec, <http://profetic.org/file/norm-0210-d-RAPPORT.pdf>.
- DE LA PASSARDIERE B. & GRANDBASTIEN M. (2003). Présentation du LOM v1.0, standard IEEE. In BRUILLARD E. & DE LA PASSARDIERE B. Eds, *Revue Ressources numériques XML et Éducation*, Hors série 2003, Sciences et techniques éducatives, Hermès-Lavoisier, p. 211-218.
- DEHORS S., FARON-ZUCKER C., GIBOIN A., STROMBONI J.-P. (2005). Un web sémantique de formation par questionnement. *Acte des journées Extraction et Gestion des Connaissances 2005 (EGC 2005)*, à paraître.
- DESMOULINS C. & GRANDBASTIEN M. (2002). Des ontologies pour la conception de manuels de formation à partir de documents techniques. *Revue Sciences et techniques éducatives*, 2002.
- DUITAMA J.-F., DEFUDE B., BOUZEGHOUB, A., CARPENTIER C. (2003). A Framework for the the Generation Of Adaptive Courses based on Semantic Metadata, *Proc. MAWIS Workshop, with OOIS'03 Conference, Geneva*, september 2003.
- IKSAL S. (2002). Spécification Déclarative et Composition Sémantique pour des Documents Virtuels Personnalisables, *Thèse de doctorat, EHESS, Brest, France*, 2002.
- KAY, J. & LUM, A. (2003). An ontologically enhanced metadata editor, TR 541, University of Sydney, Australia.2003.
- HUNTER J. & LAGOZE C. (2001). Combining RDF and XML schemas to Enhance Interoperability Between Metadata Application Profiles. In *The Tenth International World Wide Web Conference Hong Kong*, pp. 457-466, ACM Press, May 1-5, 2001 (<http://www10.org/cdrom/papers/572/>)
- LTSC-IEEE (2001) Draft Standard for Learning Object Metadata. *IEEE P1484.12/D6.1*. 18. April 2001
- PERNIN J.-P. (2003). Objets pédagogiques: Unités d'Apprentissage, Activités ou Ressources ? ». In BRUILLARD E. & DE LA PASSARDIERE B. Eds, *Revue Ressources numériques XML et Éducation*, Hors série 2003, Sciences et techniques éducatives, Hermès-Lavoisier, p.179-210,
- RANWEZ S. & CRAMPES M. (1999). Conceptual Documents and Hypertext Documents are two Different Forms of Virtual Document", *Workshop on "Virtual Documents Hypertext Functionality and the Web" of the 8th Intl World-Wide Web Conf.*, Toronto, 1999.
- REYNAUD C. (2004), Building Scalable Mediator Systems, Topical Day in *Semantic Integration of Heterogeneous Data, IFIP World Computer Congress*, Toulouse, August 23, 2004.
- THIRION B. & DARMONI S.-J. (1999), Simplified access to MeSH Tree Structures on CISMef. *Bulletin of the Medical Library Association* 1999, Oct. 87,480-1

Ontologies pour le Web Sémantique et le e-Learning

Marie-Hélène Abel¹, Thanh-Le Bach², Sylvain Dehors²,
Rose Dieng-Kuntz², Fabien Gandon², Phuc-Hiep Luong², Claude Moulin¹

¹Laboratoire Heudiasyc - UMR CNRS 6599
Université de Technologie de Compiègne
BP 20529, 60205 Compiègne Cedex - FRANCE
{Marie-Helene.Abel, Claude.Moulin}@utc.fr

² INRIA, Projet ACACIA,
2004 route des Lucioles, BP 93, 06902 Sophia Antipolis cedex
{Thanh-Le.Bach, Sylvain.Dehors, Rose.Dieng}@sophia.inria.fr

Au cours des années 1990 les ontologies ont acquis une place prépondérante dans la modélisation des connaissances et sont donc susceptibles de trouver une application dans l'activité de transfert de connaissances qu'est l'enseignement. Les ontologies partagent avec les langages cette dualité d'être à la fois des objets conçus et maintenus par ingénierie des connaissances et des primitives utilisées pour construire d'autres modèles. Ce chapitre issu du travail du WP3 de l'action WebLearn présente une synthèse sur les ontologies puis se focalise sur celles dédiées au e-Learning.

1 Méthodologies de conception d'ontologies : historique

Dans cette première partie, nous recenserons de façon synthétique les contributions apportées à la définition et aux méthodologies de conception des ontologies.

1.1 De l'Ontologie à l'ontologie

Le mot "Ontologie" se compose des racines Grecques ontos (ce que existe, l'existent) et logos (le discours, l'étude). En philosophie, on peut voir l'Ontologie comme une branche fondamentale de la Métaphysique qui s'intéresse à la qualité d'être, à la notion d'existence et aux catégories fondamentales de l'existent (Wikipedia). C'est une partie de la philosophie qui a pour objet l'étude des propriétés les plus générales de l'être, telles que l'existence, la possibilité, la durée, le devenir. (Académie) C'est l'étude de l'être en tant qu'être, c'est-à-dire l'étude des propriétés générales de ce qui existe. L'Ontologie a donc des implications directes sur notre conception de la réalité (Wikipedia).

En informatique une "ontologie" est un objet et non une science. C'est le résultat d'une tentative de formulation exhaustive et rigoureuse de la conceptualisation d'un domaine. Cette formulation inclut classiquement, une organisation hiérarchique des concepts pertinents, des relations qui existent entre ces concepts, des règles et axiomes qui les contraignent. (Wikipedia) C'est de cet aspect de description de l'existant et de ses catégories que les ontologies informatiques ont emprunté leur nom à l'Ontologie philosophique. De ce rapprochement vient aussi la possibilité d'adapter des méthodes de la philosophie pour proposer des méthodes de l'ingénierie ontologique comme nous le verrons dans les contributions recensées ci-après.

Une ontologie informatique permet, en particulier grâce aux travaux de l'intelligence artificielle symbolique, d'implanter des mécanismes de raisonnement déductif, de classification automatique, de recherche d'information et d'assurer l'interopérabilité. (Wikipedia)

Dans ce document, nous utiliserons conceptualisation au sens de (Guarino & Giaretta, 1995) i.e. une structure sémantique intensionnelle qui encode les règles implicites contraignant la structure d'un morceau de réalité. Nous emploierons aussi le terme pour dénoter l'action de construire une telle structure. Nous utiliserons le terme ontologie pour le compte rendu explicite et possiblement partiel d'une conceptualisation, ayant pour but de fournir des primitives pour la représentation de connaissances.

(Misoguchi et al., 1997) expliquaient en 1997, que la majeure partie des logiciels conventionnels sont construits sur une conceptualisation implicite et qu'un défi pour les prochaines générations seraient de construire sur des conceptualisations explicites. Dans cette optique, l'Ontologie en tant que science philosophique n'est pas suffisante et que l'on doit développer une ingénierie des ontologies. C'est le début de ce développement méthodologique que nous vous proposons de synthétiser dans la section suivante résumant un certain nombre de contributions depuis l'adoption du terme ontologie par l'informatique au début des années 90.

1.2 Brève histoire de l'ingénierie d'ontologies

Nous résumons et étendons ici la revue donnée par (Gómez-Pérez et al., 2003) sur les contributions aux méthodologies d'ingénierie d'ontologies.

Le projet Cyc (Lenat & Guha, 1990) bien qu'étend pionnier dans la formalisation massive de ce qu'ils appelaient des connaissances de sens commun, n'en avait pas moins déjà intégré l'évolution des outils de traitement de la langue naturelle. Le cycle de conception des bases de connaissances de Cyc comptait en effet trois étapes : (1) l'initialisation des bases par un codage manuel des connaissances de sens commun (2) l'utilisation d'outils exploitant les connaissances formalisées pour assister la saisie de nouvelles connaissances (3) l'application d'outil d'analyse de textes extrayant automatiquement de nouvelles connaissances en s'aidant des bases déjà disponibles.

Les concepteurs de Cyc ont aussi très tôt fait la différence entre deux étapes et deux niveaux de modélisation : le choix des primitives de représentation de

l'ontologie et des concepts de haut niveau ; l'utilisation des ces primitives dans la représentation des connaissances.

La conception de Cyc est aussi un exemple de modularisation des conceptualisations grâce à sa division en micro théories.

(Uschold & King, 1995) ont été les premiers à proposer une méthode d'ingénierie d'une ontologie, cette méthode a, en quelque sorte, fusionné avec la méthode de (Gruninger et Fox, 1995) pour donner une version longue dans (Uschold et Gruninger, 1996). Initialement, cette méthode avait quatre étapes : (1) identifier le but et la portée de l'ontologie ; (2) construire l'ontologie : capturer les connaissances, coder, réutiliser et intégrer des ontologies existantes ; (3) évaluer l'ontologie ; (4) documenter l'ontologie. L'extension de 1996 identifie et compare trois stratégies pour la construction de l'ontologie : la stratégie ascendante commence par les concepts les plus spécifiques et les généralise pour former des concepts de plus en plus abstraits ; la stratégie descendante commence par les concepts les plus abstraits et les spécialise pour former des concepts de plus en plus précis ; la stratégie centrifuge commence par des concepts centraux puis les généralise et les spécialise pour former les couches hautes et les couches basses de l'ontologie.

La contribution initiale de (Gruninger & Fox, 1995) mentionnait déjà les notions de scénarios motivants (scénarios d'usages justifiant le développement de l'application) et de questions de compétences (questions rencontrées dans les scénarios et devant être répondues par l'application) pour fixer la portée de l'ontologie. Les auteurs prônaient même la formalisation de ces questions de compétence pour la vérification formelle de la complétude de la modélisation.

(Uschold & Gruninger, 1996) donne aussi un certain nombre de conseils pour générer les définitions des concepts.

L'approche proposée dans KACTUS (Bernaras et al, 1996) est aussi conditionnée au développement d'une application : (1) spécifier l'application basée sur l'ontologie en particulier les termes à collecter et les tâches à effectuer en utilisant cette ontologie; (2) organiser les termes en utilisant les méta-catégories (concept, relation, attribut, etc.) et importer, adapter et étendre les ontologies existantes et pertinentes ; (3) affiner l'ontologie et la structurer selon des principes de modularisation et organisation hiérarchiques.

Methontology (Gómez-Pérez et al., 2003) est l'une des contributions les plus complètes car pour chaque étape du cycle de développement elle s'inspire du cycle de développement standard de l'IEEE (IEEE, 1996). De plus, cette approche ne considère pas simplement le cycle de développement mais plus globalement le cycle de vie de l'ontologie. Methontology adopte un cycle de vie par prototypes et propose un certain nombre de techniques pour chaque étape de la gestion de ce cycle (prévision, contrôle, assurance qualité), du développement (spécification, conceptualisation, formalisation, implémentation, maintenance) et du support acquisition de connaissance, intégration, évaluation, documentation, gestion de la configuration). La méthodologie considère aussi les interdépendances entre les cycles de vie de plusieurs ontologies gérées en parallèle. La conceptualisation prône une conversion progressive de l'informel au formel en utilisant un ensemble de représentations intermédiaires essentiellement sous forme de tableaux et graphes. Le

principe est de graduellement combler le fossé entre les moyens d'expressions des intéressés et les langages d'implantation des ontologies. Séquentiellement, les représentations intermédiaires sont : les glossaires de termes, les taxonomies de concepts, les diagrammes des relations binaires, le dictionnaire des concepts, le tableau des relations binaires, les descriptions des attributs d'instances, les descriptions des attributs de classes, le tableau des constantes, les axiomes et les règles, les informations supplémentaires sur les instances.

Dans (Gandon, 2002) nous soulignons l'importance de conserver ces traces de conception. La formalisation est un processus d'enrichissement durant lequel les lexiques informels initiaux, sont augmentés avec des structures logiques additionnelles formalisant les aspects pertinents à nos scénarios d'application. Toute représentation intermédiaire devrait toujours être disponible comme une vue sur l'ontologie, même après la complétion de sa formalisation.

La méthode SENSUS (Swartout et al., 1997) est complètement différente des méthodes précédentes : elle commence par la réutilisation d'une énorme ontologie commune dans laquelle les concepts pertinents sont repérés afin d'extraire le squelette initial de la future ontologie. L'ontologie initiale se comporte comme une charnière entre les différentes ontologies développées et cette méthode promeut ainsi l'interopérabilité et la réutilisation.

On-To-Knowledge (Staab et al., 2001) a proposé une méthodologie en cinq étapes: étude de faisabilité (identifier le problème et les opportunités, se focaliser sur les options les plus prometteuses), coup d'envoi (spécification des besoins, analyse des entrées, taxonomie de base), raffinage (éliciter les concepts avec les experts, conceptualisation et formalisation, relations et axiomes) évaluation, (idem étude de faisabilité), maintenance (gestion du processus organisationnel de maintenance).

Le goulet d'étranglement de l'acquisition de connaissances pour les ontologies a aussi donné lieu à des travaux dits d'apprentissage d'ontologies. Les tâches assistés par les outils issus de ces contributions inclues : l'apprentissage d'ontologies à partir de textes (Aussenac-Gilles et al, 2000) (Kietz et al., 2000) , l'apprentissage d'ontologies à partir d'instances (Morik & Kietz, 1989) et l'analyse formelle de concepts pour la génération de treillis minimums (Sowa, 2000b).

Un aspect assez peu développé dans les autres méthodologies, et pourtant vital à la notion de consensus, c'est l'assistance à sa conception collaborative. Co4 est un protocole pour la construction collaborative de bases de connaissances (Rechenmann, 1993 ; Euzenat, 1995 ; 1996). Il spécifie une organisation hiérarchique des bases de connaissances sujettes à la recherche d'un consensus et un ensemble de protocoles pour gérer et implanter le résultat des accords et désaccords entre les différentes communautés.

Afin de systématiser la vérification du squelette ontologique, (Guarino & Welty, 2000) ont formalisé et illustré un ensemble de conditions avec leur définitions, fournissant ainsi aux ontologies des éléments méthodologiques et un système formel extrêmement rigoureux pour nettoyer et valider une taxonomie de concepts.

De même, repartant d'une approche linguistique pour la modélisation des connaissances, (Bachimont, 2000) propose de déterminer le sens d'un nœud / concept dans l'arbre ontologique / taxonomie, en vérifiant systématiquement quatre principes :

le principe de communauté avec le père ; le principe de différence avec le père ; le principe de différence avec les frères ; le principe de communauté avec les frères.

Toujours, pour aider à la construction de la taxonomie des concepts, (Kassel, 2002) repose sur la notion d'axe sémantique groupant les enfants d'un nœud selon les caractéristiques impliquées dans la définition de leur différentia. En fusionnant ce principe avec les travaux de (Guarino & Welty, 2000), la méthode OntoSpec (Kassel, 2002) assiste la spécification semi-informelle des ontologies en contraignant l'écriture et les aspects considérés lors de la documentation des concepts retenus.

2 Les ontologies : genèse et cycle de vie

Les ontologies sont à la fois un modèle de notre monde en terme de catégories d'objets et de relations possibles entre ces objets (ex : il existe des livres de cours, des étudiants, et un étudiant peut avoir à lire un livre de cours) et un ensemble de primitives permettant de décrire des situations particulières de ce monde (ex : l'étudiant "Jérôme Thomas" doit lire le livre de cours "Bescherelle"). Les ontologies capturent un vocabulaire conceptuel fournissant une base cohérente non ambiguë et une référence commune utilisable pour décrire et échanger des modèles et implanter les inférences qui s'en nourrissent. Pour être utilisables comme références communes, les ontologies doivent résulter d'un consensus ; elles réclament donc une construction et une maintenance collectives et sont dépendantes d'un point de vue socioculturel.

De par leur nature d'objet de modélisation de connaissances, les ontologies héritent d'un cycle de vie inspiré par le cycle de vie des (bases de) connaissances. De par leur nature de primitives de modélisation, leur cycle de vie se répercute sur tous les modèles et les algorithmes qui les exploitent.

Comme le remarquent Guarino et Welty (2000) la branche philosophique de l'Ontologie évolue maintenant vers une ingénierie qui demande une méthodologie basée sur des principes forts. L'ingénierie ontologique s'occupe des aspects pratiques, essentiellement les méthodologies et les outils permettant d'appliquer des résultats de la théorie des ontologies à la construction d'ontologies, pour un cas précis et avec un but spécifique.

La conception et la maintenance d'une ontologie suivent un processus de maturation itératif. Comme pour tout projet de large envergure, la conception d'une ontologie doit être sujette à une gestion de projet adaptée à son cycle de vie (Fernandez et al., 1997). En considérant la norme IEEE 1074-1995, l'ouvrage d'Ian Sommerville sur l'ingénierie logicielle (Sommerville, 1995) et les travaux d'Asunción Gómez-Pérez (1998) et de Fernandez et al. (1997) nous allons structurer la présentation de ce cycle de vie en s'inspirant des cycles de développement logiciel : l'ontologie étant maintenant un objet logiciel, un certain nombre de questions et d'activités peuvent être importées de la gestion de projets logiciels pour la description de la conception et du cycle de vie d'une ontologie.

2.1 Modèles de cycle de vie

En conception logicielle la première question est celle du choix d'un modèle du cycle de vie parmi les deux grandes catégories connues : modèles séquentiels et modèles itératifs. Ces modèles donnent simplement la tendance générale du processus de conception et de réalisation : en pratique les étapes de ces modèles se recouvrent, interagissent, s'influencent. L'évolution des besoins, le retour des usages, la découverte de vices cachés, etc. induisent des modèles de vie cycliques à plusieurs niveaux. Dans le cas des ontologies, la détection d'incohérences, l'émergence de nouvelles notions, l'extension à de nouveaux scénarios, etc. sont des exemples d'événements qui introduisent des cycles dans l'évolution des ontologies

Les modèles séquentiels les plus connus sont ceux de (Royce, 1970) avec sa représentation en cascade comme montrée en Figure 1 et le modèle en "V" montré en Figure 2.

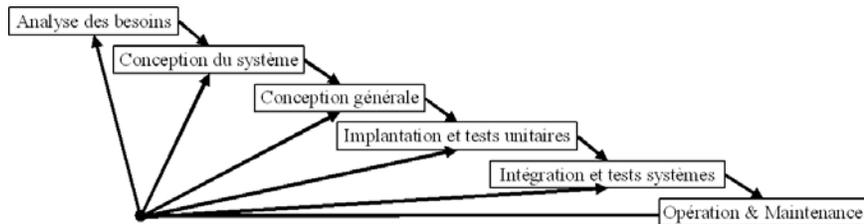


Fig. 1. – Modèle de développement en cascade (Royce, 1970)

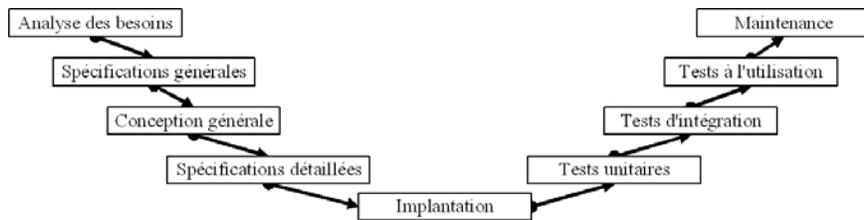


Fig.2 – Modèle en "V"

Ces deux modèles prônent un ordonnancement des tâches comme suit : l'analyse des besoins → la spécification → la conception → l'implantation → les tests → la maintenance. Bien que ces modèles soient très linéaires, la tâche de maintenance introduit une première forme de cycle par exemple : l'évolution des usages, les retours d'utilisation, la découverte de vices cachés, etc. relanceront le cycle respectivement au niveau de l'analyse des besoins, de la spécification ou de la conception.

Ces cycles de vie sont notoirement lourds, alors, pour de petits systèmes, pour des solutions à courtes durées de vie, pour des systèmes n'ayant pas de spécifications définies à l'avance ou faisant partie d'un système plus large, ont été proposées des approches de développement dites évolutives. Leur principe est le suivant : une implantation initiale rapide, une exposition aux utilisateurs immédiate et des itérations rapides pour converger vers une solution. Ces modèles de projets sont divisibles entre modèles de projets à base de prototypes évolutifs (exploration et implantation incrémentale de la solution) où l'itération se fait essentiellement autour de la conception et de la réalisation, et les projets à base de prototypes jetables (compréhension des besoins et explicitation des spécifications) où l'itération se fait essentiellement autour de l'analyse des besoins et de la spécification en reposant sur des maquettes rapides et pas forcément réutilisables. Enfin la livraison peut elle aussi être incrémentale : si la solution est modulaire certains modules peuvent être livrés et déployés avant d'autres. Les modèles évolutifs itératifs ne sont en fait pas très éloignés des modèles séquentiels précédents : ils sont définis sur des processus classiques en choisissant les séquences d'itérations i.e. en introduisant des micro-cycles dans l'ordonnancement des étapes de la Figure 3 :

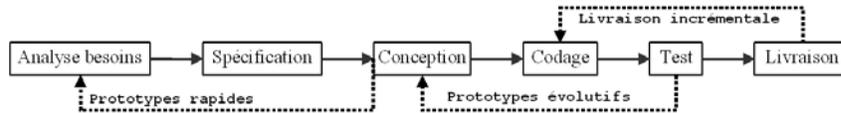


Fig. 3 – Micro-cycles des méthodes évolutives

Dans IDEF5, le développement à proprement parler est par exemple divisé en deux phases : le développement initial fournissant des proto-concepts (prototypes des concepts, relations et propriétés) et le raffinement couplé à l'évaluation où les proto-concepts sont affinés et testés.

Nous avons vu qu'il y avait des cycles dus à la maintenance et des micro-cycles dus aux contraintes de conceptions et à la nature du projet. Il existe aussi des cycles macroscopiques le plus connu étant le cycle en spirale de (Boehm, 1988) schématisé en Figure 4.

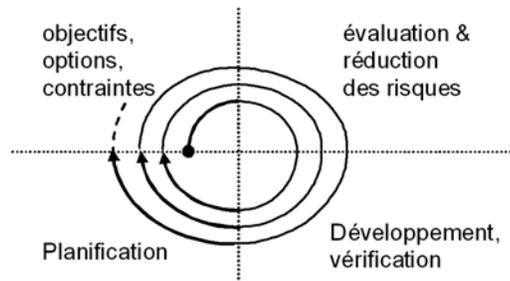


Fig. 4 – Le cycle en spirale

Le cycle en spirale prend explicitement en compte dans ses itérations l'évaluation des risques et la planification globale en plus des étapes de définition des objectifs et de développement et validation. Ce cycle se définit en fait un cran au dessus des modèles précédents : différents modèles de projets (en cascade, en "V", prototypiques, etc.) peuvent être utilisés pour effectuer différentes révolutions de la spirale, par exemple le cycle peut commencer avec plusieurs prototypages puis une approche en "V" une fois les spécifications bien établies.

Les méthodes évolutives sont très liées à la culture IA et c'est pourquoi la communauté des ontologistes prône souvent des cycles de vie prototypiques pour les ontologies. Fernandez et al. (1997) préconisent un cycle par prototypes. Selon eux, le cycle de vie par évolution de prototypes permet à l'ontologiste de retourner de n'importe quel état à n'importe quel autre si une certaine définition manque ou est erronée, permettant en permanence l'inclusion, le déplacement ou la modification de définitions. Cependant l'activité de maintenance ne justifie pas à elle seule le cycle prototypique : une ontologie, comme un logiciel, est livrée une fois jugée stable et opérationnelle c'est-à-dire une fois passé le stade de prototype mais elle continue à être maintenue après. Le cycle global est donc plutôt un cycle en spirale avec de premières révolutions par prototypes plus des révolutions sur des modèles classiques pour la finition et la maintenance.

2.2 Gestion du projet

Une fois le modèle de projet choisi, la prochaine étape est de démarrer l'activité de gestion du projet. Asunción Gómez-Pérez (1998) et Fernandez et al. (1997) parlent de la perspective de la gestion du projet et y reconnaissent essentiellement trois facettes :

- Planifier : identifier les principales tâches et planifier le temps et les ressources ;
- Contrôler que les tâches sont réalisées de la façon prévue ;
- Effectuer un contrôle qualité sur l'ontologie et sa documentation.
- Ils suivent en cela la norme IEEE 1074-1995 qui requiert :
- La mise en correspondance des étapes du modèle de projet avec des activités/tâches bien définies : pour chaque étape du modèle il faut identifier les tâches d'ingénierie ontologique correspondantes.
- L'affectation des ressources au projet et à ses activités : il s'agit ici d'une gestion classique des ressources allouées au projet.
- Le choix de l'environnement du projet (méthodologies, normes, outils, formalismes, etc.). Uschold et Gruninger (1996) distinguent deux phases préliminaires au codage à proprement parler : s'engager sur la méta-ontologie i.e. les primitives de base à utiliser pour spécifier l'ontologie (e.g. classe, entité, relation, etc.) et choisir un langage de représentation capable de supporter cette méta-ontologie. Il existe de nombreux langages de formalisation par exemple : la logique des prédicats ou logique du premier

ordre (le langage CycL utilisé pour Cyc en est proche (CycL), ainsi que KIF (Genesereth & Fikes, 1992)), les langages de programmation logique (Prolog, CLIPS, etc.), les graphes conceptuels (Sowa, 1984) (Sowa, 2002), les Topic Maps (Biezunski, 2001), les formalismes objets ou de frames (Ducourneau et al., 1998), les logiques de description (Ducourneau et al., 1998 ; Kayser, 1997) ;

- La planification de la gestion du projet : organisation de l'équipe, responsabilités, processus de gestion et de contrôle, échéancier et, budget. Le processus de construction d'une ontologie est un projet de conception et doit être traité comme tel, surtout si la portée de l'ontologie est large. Les techniques de gestion de projet et d'assurance qualité doivent être appliquées. (Fernandez et al., 1997) souligne qu'avant de construire une ontologie, il faut planifier les tâches principales, leur ordonnancement, leur durée et les ressources nécessaires (personnes, logiciel et matériel).

Une fois le projet démarré, la gestion du projet continue bien évidemment en parallèle avec, en particulier :

- le suivi et le contrôle du projet : analyse des risques, re-planification en réponse aux situations imprévues, mémorisation de l'historique des décisions, suivi des coûts et de l'échéancier.
- la gestion de la qualité : définition et suivi du plan d'assurance de qualité, définition des métriques, évaluation des besoins d'amélioration et décisions de correction.

Le coût de construction (recueil, conceptualisation, engagement, formalisation, opérationnalisation) doit être contrôlé et pour cela le travail sur l'ontologie doit être maintenu dans les limites de la portée nécessaire à l'application ; ce coût peut non seulement se révéler prohibitif à la planification mais aussi exploser lors de la réalisation si la circonscription au domaine de conception suffisant n'est pas précisément contrôlée (Dieng et al., 2001).

2.3 Conception et développement

La norme IEEE 1074-1995 définit trois stades de développement que Asunción Gómez-Pérez (1998) et Fernandez et al. (1997) rassemblent dans leur perspective technique : les activités de pré-développement, de développement et de post-développement.

Ces étapes sont à mettre en vis-à-vis avec celles proposées par Uschold et Gruninger (1996) pour la construction d'une ontologie : Identifier les objectifs et la portée de l'ontologie ; Construire l'ontologie (capturer/recueillir la connaissance ontologique, coder l'ontologie) ; Evaluer l'ontologie et Documenter l'ontologie.

Une autre décomposition est aussi donnée par IDEF5 : définition de la portée et planification, recueil des données, analyse, développement initial et raffinement et évaluation.

2.3.1 Les activités de pré-développement

Plusieurs activités peuvent être identifiées à ce stade dans la norme IEEE 1074-1995 :

- L'exploration du domaine qui inclut : l'identification des besoins, des approches possibles et des contraintes imposées, l'étude des opportunités, la finalisation de la description exhaustive des besoins.
- Affectation des fonctionnalités du système au matériel et au logiciel.

Une approche intéressante pour l'analyse des besoins passe par l'écriture de scénarios par les utilisateurs : ce sont des descriptions textuelles qui décrivent soit des exemples de problèmes courants (scénarios actuels) ou des solutions envisageables (scénarios souhaités) et capturent ainsi les motivations pour la construction d'une ontologie. Carroll (1997) explique que les scénarios sont une façon naturelle et efficace de capturer les besoins des utilisateurs dans leur contexte, ce qui est essentiel pour une intégration symbiotique du système à l'environnement de travail. Les principaux avantages que nous avons identifiés en utilisant des scénarios (Gandon, 2002) sont :

- Se focaliser sur les aspects spécifiques au cas d'application considéré et capturer ainsi la portée du travail effectué ;
- Capturer et présenter une image globale permettant de voir l'ontologie comme un composant d'une solution globale ;
- Rassembler un ensemble concret d'interactions, compréhensible et accessible pour tous les intéressés du projet et fournir un excellent point de départ pour construire des 'use case' ;
- Fournir un cadre pour évaluer et contrôler chaque nouvelle idée.

L'analyse par scénario peut être facilitée par la définition d'une grille suggérant les principaux aspects à considérer lors de la description d'un scénario (Gandon, 2002). Durant la phase de recueil, cette grille rappelle les thèmes généraux des informations à rechercher et les points qui pourraient être affinés. Le scénario aide à spécifier l'ontologie : elle doit fournir l'expressivité nécessaire aux interactions décrites dans les scénarios. L'analyse par scénario produit des traces, les rapports de scénario, qui sont extrêmement riches et donc de très bons candidats à inclure dans le corpus d'une étude terminologique comme nous l'évoquerons dans la section suivante.

Uschold et Gruninger (1996) utilisent des 'scénarios motivants' afin identifier l'ensemble des requêtes survenant et qui représentent des attentes envers l'ontologie et des besoins d'expressivité sous la forme de questions appelées 'questions de compétence'. L'ontologie visée doit être capable de représenter ces questions et leurs réponses ; les questions servent de contraintes de spécification et suggèrent des tests de validation de l'ontologie.

2.3.2 Les activités de développement

Le développement repose sur trois activités :

- Etablissement des exigences (Spécification des exigences des interfaces, Spécification des fonctionnalités, analyse des priorités des exigences). Les scénarios permettent de déterminer pourquoi l'ontologie est construite et quels sont les utilisateurs visés ; il s'agit ici d'en abstraire les spécifications de l'ontologie qu'ils requièrent.
- Conception (Architecture, Structure et gestion des données, Interfaces, Choix et développement des algorithmes, Conception détaillée) La conception de l'ontologie c'est le choix d'une conceptualisation i.e. la construction d'un modèle conceptuel du monde ;
- Mise en œuvre (Créer les données de test, Créer le code source, Générer le code objet, Rédiger la documentation d'opérateur, Planifier l'intégration, Réaliser l'intégration ; En termes de conception d'ontologie il s'agit essentiellement de la formalisation (construire un modèle formel de la conceptualisation) et de son implémentation (coder cette formalisation dans un langage).

Comme l'explique Mizoguchi et al. (1997) l'objectif du développement d'ontologie c'est de rendre explicite les conceptualisations qui jusqu'à maintenant étaient essentiellement implicites et codée en dur dans les logiciels et les choix de conceptions dont ils ont fait l'objet.

Une ontologie inclura nécessairement un vocabulaire de termes et une spécification de leur sens i.e. des définitions plus ou moins formalisées. Uschold et Gruninger (1996) utilisent les questions de compétence pour identifier l'ensemble des termes qui formeront la base de spécification de la terminologie. L'approche ici est dite centrifuge (middle-out) car l'objectif premier est d'identifier des termes pivots (ex: document, étudiant, université,...) autour desquels rassembler les autres (ex : livre, transparents, etc.)

Les spécifications d'une ontologie soulèvent les questions de la couverture/portée et de la granularité qu'elle doit offrir. En adaptant des critères utilisés pour caractériser des index en recherche d'information (Khorfhage, 1997) on peut proposer trois dimensions pour la caractérisation pour la spécification, l'évaluation ou la comparaison d'ontologies (Gandon, 2002) :

- Exhaustivité : c'est la largeur du domaine couvert i.e. l'étendu couverte par l'ontologie en terme de concepts et de relations nécessaires.
- Spécificité : la profondeur du domaine couvert i.e. dans quelle mesure l'ontologie fournit précisément les concepts et relations nécessaires.
- Granularité : le niveau de détail dans la définition formelle des concepts et des relations.

Une structure forte de l'ontologie, souvent appelée squelette ontologique, est la taxonomie des types de concepts. Les principes derrière cette structuration remontent à Aristote qui définissait une espèce en donnant son *genus* (genos) et son *differentia*

(diaphora). Le genus est un genre plus général sous lequel l'espèce considérée se place. Le differentia est ce qui caractérise l'espèce dans ce genre.

La façon de concevoir la structure taxinomique est encore sujette à beaucoup de discussions dans la communauté. Notre compréhension des différentes contributions faites jusqu'à maintenant est qu'il y a une tendance à distinguer trois options de construction (Gandon, 2002) :

- Approche ascendante : l'ontologie est construite par généralisation en partant des concepts des basses couches taxinomiques. Cette approche encourage la création d'ontologies spécifiques et adaptées.
- Approche descendante : l'ontologie est construite par spécialisation en partant de concepts des hautes couches taxinomiques. Cette approche encourage la réutilisation d'ontologies.
- Approche centrifuge : la priorité est donnée à l'identification de concepts centraux à l'application que l'on va ensuite généraliser et spécialiser pour compléter l'ontologie. Cette approche encourage l'émergence de domaines thématiques dans l'ontologie et favorise la modularité.

Ces approches peuvent aussi être considérées comme trois perspectives complémentaires d'une méthodologie complète : on considère alors qu'un ontologiste doit effectuer les tâches définies dans ces trois perspectives en parallèle (Gandon, 2002).

La conception de cette taxonomie de types est sans aucun, d'un point de vue méthodologique, la plus étudiée. On peut citer les contributions suivantes.

Après six ans de travail sur Cyc, Lenat et Guha (1990) ont publié un livre sur le projet et leur retour d'expérience. A l'époque il y avait peu de travaux sur les ontologies et par conséquent la question de la réutilisation de sources externes était assez limitée ; le approche globale se divisait en deux grandes étapes : (1) faire la partie haute de l'ontologie encore appelée ontologie globale (2) y relier les autres domaines.

Bachimont (2000) propose des principes aristotéliens étendus pour structurer les intensions. Ces principes poussent à éliminer le multi-héritage au profit de la multi-instanciation. Une solution à cette restriction serait d'explicitier des points de vue dans la taxonomie (Rivière, 1997) et de limiter l'application de ces principes étendus à l'intérieur d'un point de vue. Une approche est aussi proposée dans (Kassel et al., 2000) introduisant la notion d'axes sémantiques pour regrouper sous un genus les différentes dimensions ou natures de critères suivant lesquelles le differentia s'énonce. Par exemple sous le genus 'Document', pour l'axe sémantique 'support' on aura 'message électronique' et 'message papier' (differentia papier/électronique) et pour l'axe sémantique 'périodicité' on aura 'quotidien', 'hebdomadaire', 'mensuel'... (le differentia est la période)

De même, le travail de Guarino et de Welty (Guarino, 1992 ; Guarino & Welty, 2000) contribue à poser les bases fondamentales de l'ingénierie ontologique fournissant un cadre théorique, des définitions et des contraintes basées sur ces dernières permettant de guider la construction et de vérifier la taxonomie : un

ontologiste se fondant sur ces définitions peut ainsi contrôler certains aspects de validité de ses liens de subsumption.

Comme le remarque Sowa (2000b) un problème connexe au problème de la structuration taxonomique est la génération automatique de treillis comme étudiée dans le domaine de l'analyse formelle de concepts. Sowa montre (2000b) sur un exemple simple l'intérêt des algorithmes itératifs et incrémentaux mis au point dans ce domaine pour proposer des hiérarchies de types. (Delteil, 2002) propose un algorithme permettant de regrouper systématiquement les objets décrits par un même motif. Cette approche peut être vue comme une généralisation de l'Analyse Formelle de Concepts où les attributs sont des motifs relationnels. L'algorithme est incrémental dans le sens où la taille des motifs augmente graduellement d'une unité à chaque pas. Pour adapter l'algorithme à des situations réelles, des heuristiques sur la forme, l'expressivité ou la fréquence des motifs sont introduites.

Enfin outre les types de concepts, les types de relations sont aussi organisables en taxonomies. Staab et Maedche (2000) décrivent une méthode pour découvrir des relations non taxonomiques à partir de textes en utilisant des règles d'association généralisée et en exploitant la taxonomie de types de concepts. De plus ils proposent aussi une organisation des axiomes en introduisant des types d'axiomes et une catégorisation de ces types.

Coder l'ontologie, c'est la représenter explicitement dans un langage plus ou moins formel la conceptualisation capturée. Il ne s'agit pas de remplacer une version informelle de l'ontologie par une version formelle, mais de développer le code formalisant les aspects sémantiques nécessaires aux applications envisagées pour l'ontologie.

Uschold et Gruninger (1996) donnent un certain nombre d'instructions pour la génération de définitions :

1. Produire une définition textuelle en langage naturel aussi précise que possible ;
2. S'assurer de la cohérence avec les termes déjà utilisés, en particulier : utiliser au maximum les dictionnaires, thesaurus, et autres glossaires techniques ; Eviter d'introduire de nouveaux termes autant que possible.
3. Indiquer les relations avec d'autres termes utilisés communément et similaires à celui en cours de définition (ex : synonymes)
4. Eliminer les définitions circulaires ;

La définition de chaque terme vise à être nécessaire et suffisante autant que faire se peut en langue naturelle ; il faut donner toutes les clarifications et informations supplémentaires essentielles à la compréhension. L'adjonction d'exemples aide la compréhension des distinctions et la clarification des subtilités.

La formalisation ne doit pas transformer, mais augmenter l'ontologie informelle ; Mizoguchi et Ikeda (1997) insistent sur le fait que les ontologies doivent être intelligibles à la fois pour les machines et les humains. Cela joue un rôle très important pour la documentation et par extension pour l'appropriation, la réutilisation et la maintenance d'une ontologie.

Uschold et Gruninger (1996) notent qu'il y a des variations considérables dans le degré d'effort nécessaire pour se mettre d'accord sur les définitions et les termes. Les auteurs distinguent trois cas :

- Un terme correspondant à une et une seule définition : cas rêvé sans aucun problème ;
- Plusieurs termes pour une définition : les termes sont synonymes, on garde la liste des synonymes et on en choisit un pour se référer à cette définition ;
- Un terme correspondant à plusieurs concepts (plusieurs définitions utilisées) : ceci correspond à une ambiguïté.

Uschold et Gruninger (1996) gèrent l'ambiguïté de la façon suivante :

1. Suspendre l'utilisation du terme ;
2. Définir attentivement chaque concept dénoté par le terme en utilisant seulement des termes dont la définition est consensuelle ; il peut être utile de donner à ces concepts des labels sans signification x_1 , x_2 , x_3 , etc. de manière à pouvoir s'y référer de façon neutre
4. Déterminer les concepts suffisamment importants pour figurer dans l'ontologie ;
5. Choisir un terme de référence pour chaque concept en évitant d'utiliser tout terme ambigu.

Guarino (1997) remarque qu'un concept peut être 'pertinent' pour une tâche sans être nécessairement 'spécifique' à cette tâche. La réutilisation doit donc être possible et il faut la favoriser. Asunción Gómez-Pérez (1998) parle d'intégration d'ontologies, et distingue plusieurs types : l'intégration lors de la construction d'une nouvelle ontologie ; la fusion de plusieurs ontologies dans une seule ; l'intégration d'ontologies dans d'autres applications.

Dans tous les cas, pour décider s'il faut réutiliser des ontologies existantes et si ou comment, il est important que soient explicites les hypothèses sous-jacentes aux ontologies. Ici les questions de compétence, les scénarios motivant peuvent être utiles pour identifier les ontologies susceptibles d'être réutilisées.

Diverses opérations participent à l'intégration (Dieng et al, 2001) : l'inclusion telle quelle ; le raffinement des définitions ; la simplification des hypothèses sous-jacentes ; l'extension par de nouveaux concepts et relations ; l'inclusion de différents points de vue et de leurs interdépendances.

La réutilisation d'ontologies est à la fois séduisante (elle devrait permettre d'économiser du temps et des efforts et favoriser la normalisation) et difficile (les engagements et les conceptualisations doivent être réajustés entre l'ontologie réutilisée et l'ontologie désirée). Bernaras et al. (1996) ont étudié la faisabilité de la réutilisation de la connaissance, en concevant des ontologies pour un domaine particulier et en les réutilisant pour différentes tâches ou applications. Ceci peut être appelé une réutilisation orientée domaine. Les auteurs expliquent que la construction d'une ontologie montre que la modularisation est une très bonne stratégie de conception d'ontologie. Cependant des distinctions théoriques plus poussées peuvent être difficiles à maintenir en pratique. Ce que vérifient les auteurs ici c'est le paradoxe " utilisabilité / réutilisabilité " qui dit que plus une ontologie est réutilisable moins elle

est directement utilisable et vice-versa. Cela se vérifie aussi au niveau des concepts : les concepts les plus généraux sont plus facilement réutilisables que les concepts spécifiques ; mais les concepts spécifiques sont plus proches de l'application que ne le sont les concepts généraux.

Uschold et Gruninger (1996) proposent de construire des bibliothèques d'ontologies pouvant être adaptées à différentes classes de problèmes. Le problème qui se pose est de déterminer quelles ontologies sont les plus appropriées pour un problème donné. Ils proposent que les ontologies soient distinguées sur la base de leurs 'questions de compétence' ; c'est à dire qu'une ontologie peut être capable de représenter un ensemble de questions de compétence différent d'une autre ontologie. Les auteurs proposent alors de comparer et choisir les ontologies d'une bibliothèque en utilisant ces questions.

D'un point de vue formel, la réutilisation pose aussi des problèmes, en particulier, la traduction. Outre la difficulté technique qu'il peut y avoir à passer d'un formalisme à un autre, l'expressivité des différents langages n'étant pas la même, les traductions se font forcément avec pertes.

En pratique, Bachimont (2000) observe qu'il est toujours possible d'adapter une ontologie, mais jamais possible de la réutiliser telle qu'elle est.

2.3.3 Les activités de post-développement

Le Post-développement inclut : l'installation, le support, la maintenance, le retrait.

La maintenance d'une ontologie a des conséquences qui vont au-delà du cycle de vie de l'ontologie. Elle affecte tout ce qui a été construit en utilisant cette ontologie. Pour un logiciel où l'ontologie a été "codée en dur" une nouvelle version doit être éditée, pour des bases de connaissances la cohérence doit être préservée. Ainsi, bien que le problème de l'évolution de l'ontologie soit complexe en lui-même, il faut aussi prendre en compte le fait que les ontologies fournissent des briques de bases pour construire d'autres objets, et que par conséquent leur évolution a des répercussions sur ces autres objets (Dieng et al., 2001).

Une approche intéressante, passe par le typage des changements par exemple pour dissocier ceux qui affectent la hiérarchie des types de ceux qui ne l'affectent pas (Olivier et al., 1998).

2.3.4 Activités parallèles et intégrées

En plus de la gestion de projet, la norme identifie quatre processus d'ensemble intégrés appliqués à tous les autres processus :

- Vérification et validation (Préparation du plan de vérification et validation, Tâches de vérification et validation : révisions et audits, recueil et analyse des données métriques, Planification des essais, exigences des essais, effectuer les essais)
- Gestion des configurations du logiciel (Planification de la gestion des configurations, définition de l'identification des configurations, Contrôle des configurations, Rapports d'état des modifications)

- Documentation (Planification de la documentation, Réalisation de la documentation, Production et distribution de la documentation)
- Formation (Planifier le programme de formation, Développer le matériel de support, Valider le programme de formation, Réaliser la formation)

Cette dernière partie correspond pour Asunción Gómez-Pérez (1998) et Fernandez et al. (1997) à la perspective support ; pour eux, tout le long du processus de développement de l'ontologie doivent avoir lieu : l'acquisition des connaissances (elle peut se faire à partir d'experts humains, de documents, de figures, de tables, voire d'autres ontologies), l'évaluation de l'ontologie, l'intégration, la documentation de l'ontologie, la gestion des configurations.

Comme le propose Fernandez et al. (1997) l'acquisition de connaissances peut être vue comme une activité de support au sens où elle se déroule en parallèle avec les autres processus de conception qu'elle nourrit en connaissances. Différentes techniques d'acquisition de connaissances sont applicables à l'acquisition des connaissances ontologiques (Gandon, 2002) :

- Les entretiens libres (Fernandez et al., 1997) avec les experts, en particulier au début pour les premières itérations (scénarios et initialisation)
- Les entretiens structurées (Fernandez et al., 1997) pour compléter des points particuliers, pour vérifier et valider les modèles et effectuer des tests ;
- Les entretiens semi structurés (Gandon, 2002) en trois phases : une première partie en entretien libre ; une deuxième partie avec des questions structurées ainsi que des retours concernant la première partie; une phase de synthèse et résumé des points importants.
- Les observations en situation in situ ou en simulation.
- L'analyse semi-automatique des textes : l'application d'outils statistiques et de traitement de la langue naturelle pour extraire des termes et leur définition (c.f. paragraphe suivant)
- Des analyses informelles de documents (Fernandez et al., 1997) pour analyser les tables, les
- Le brainstorming et brainwriting : Uschold et Gruninger (1996) font appel au brainstorming pour identifier les termes pertinents et produire des définitions consensuelles. Pour eux, les étapes de la capture sont donc : identification des concepts et des relations clés ; identification des termes par lesquels on s'y réfère ; production de définitions textuelles non ambiguës ; validation d'un accord sur les points précédents.
- Les questionnaires : ils sont un moyen pratique et peut coûteux pour faire une acquisition déjà ciblée cependant l'élaboration du questionnaire est un travail critique et difficile (La France, 1992)

L'expression de l'intension d'un concept part presque systématiquement d'une définition en langue naturelle : définir une ontologie est donc une tâche de modélisation menée à partir de l'expression linguistique des connaissances (Bachimont, 2000). Dès lors il est intéressant d'essayer d'automatiser tout ou partie des tâches de l'analyse de textes que requiert la construction d'ontologies

Le projet Cyc (Lenat et Guha, 1990) a bien considéré l'utilisation de techniques du traitement automatique de la langue naturelle (TALN), mais à l'époque les outils de ce domaine n'étaient pas assez avancés, et de plus les auteurs étaient convaincus que sans un apport minimal de connaissances, les systèmes de TALN étaient inefficaces et qu'ils ne pourraient être utilisés que plus tard pour compléter et maintenir Cyc.

2.4 Construction d'une ontologie à partir d'un corpus

Certains travaux de recherche reposent sur l'analyse de textes (par exemple, analyse linguistique de documents techniques ou d'interviews d'experts) pour aider à la construction semi-automatique d'ontologies (Bourigault, 1994 ; Bourigault et Lépine, 1995 ; Assadi, 1998 ; Biébow et Szulman, 1999).

L'outil LEXTER (Bourigault, 1994 ; Bourigault et Lépine, 1995) construit, à partir d'un corpus de textes, un réseau de «candidats termes» à partir duquel le cogniticien (ou le terminologue) assisté de l'expert constitue l'ontologie.

Pour construire une ontologie à partir de documents textuels, on peut :

- soit reposer sur des ressources linguistiques externes comme WordNet ou d'autres ontologies,
- soit se baser entièrement sur le corpus constitué à partir des documents textuels.

La méthodologie de construction d'une ontologie à partir de textes proposée par le groupe TIA et décrite dans Aussenac-Gilles et al. (2000) repose sur les étapes suivantes :

- Constitution du corpus de textes, compte-tenu des objectifs visés par l'application.
- Analyse linguistique : choisir et appliquer sur ce corpus textuel des outils linguistiques adéquats comme des extracteurs de termes (Lexter, Nomino...) permettant de proposer des candidats termes ou des extracteurs de relations (Syntex) permettant de proposer des relations entre ces termes.
- Normalisation :
 - La normalisation linguistique permet au cogniticien de choisir, parmi les termes et les relations lexicales extraits précédemment, ceux qui seront modélisés dans l'ontologie. Le cogniticien associera à chaque terme et relation retenu(e) une définition en langue naturelle, si possible

proche du texte du corpus. Si un terme ou une relation ont plusieurs sens dans le domaine considéré (i.e. polysémie), le cogniticien décide quels sens attestés par le corpus seront gardés du fait de leur pertinence.

- La modélisation conceptuelle : les concepts et les relations sémantiques sont alors définis dans une forme normalisée utilisant les étiquettes des concepts et les relations déjà définis.
- Formalisation : elle consiste en la construction et validation de l'ontologie. Des ontologies existantes peuvent éventuellement aider à construire les niveaux les plus élevés de l'ontologie et à structurer cette dernière en grands sous-domaines. Les concepts et relations sémantiques sont alors formalisés et représentés dans le formalisme de représentation des connaissances choisi (par exemple, les logiques de description ou les graphes conceptuels). Au besoin, des concepts additionnels (i.e. concepts de structuration, non nécessairement attestés par les textes) peuvent être ajoutés pour structurer l'ontologie. Une validation globale pourra être effectuée dès que l'ontologie aura atteint un état stable.

Cette méthode est par exemple supportée par l'outil TERMINAE (Biébow et Szulman, 1999 ; Biébow, 2004) qui exploite des techniques linguistiques et des techniques d'ingénierie des connaissances pour construire une ontologie à partir de textes. Les techniques linguistiques permettent la définition de formes terminologiques à partir de l'analyse des occurrences des termes dans un corpus. Une forme terminologique définit chaque sens d'un terme (on appelle notion ce sens) en utilisant des relations linguistiques entre notions (e.g. la synonymie). TERMINAE offre également un éditeur et un navigateur pour construire et explorer l'ontologie. Il permet de représenter une notion comme un concept (appelé concept terminologique). D'autres concepts non terminologiques (i.e. ne provenant pas du corpus) pourront aussi être intégrés dans l'ontologie.

TERMINAE permet donc de construire les concepts terminologiques à partir de l'étude des termes correspondants dans le corpus, en reposant sur les étapes suivantes:

- Etablir la liste des termes : l'extracteur de termes LEXTER est utilisé pour proposer une liste de candidats termes, parmi lesquels le cogniticien assisté de l'expert choisit les termes effectifs.
- Conceptualiser chaque terme : le cogniticien analyse les diverses utilisations du terme dans le corpus, et définit toutes les notions associées à ce terme (c'est-à-dire les sens possibles de ce terme). Il fournit une définition en langue naturelle pour chaque notion et, sous TERMINAE, il décrit cette définition dans un formalisme de représentation des connaissances (une logique de description). Le nouveau concept terminologique ainsi créé peut ou non être inséré dans l'ontologie.

Cette approche TIA peut être adoptée exploités pour construire les ontologies du domaine à partir d'un corpus de textes, constitué à partir de documents bien choisis de l'entreprise.

Suivant cette méthodologie TIA, (Golebiowska et al, 2001 ; Golebiowska, 2002) proposent une approche pour construire et enrichir une ontologie en exploitant d'une part une base de donnée structurée, d'autre part les champs textuels d'une base de données. L'approche consiste à :

- Exploiter un outil de traitement linguistique (plus précisément un extracteur de termes) sur les champs textuels de cette base de données.
- Analyser d'une part les entretiens avec les experts et d'autre part, les candidats termes obtenus de façon à repérer les termes les plus importants pour initialiser l'ontologie avec des concepts représentant les termes importants repérés, et ce, avec validation par les experts.
- Analyser les structures des candidats termes, afin d'en déterminer les régularités linguistiques qui serviront pour construire des règles heuristiques, permettant d'enrichir l'ontologie initiale.
- Annoter semi-automatiquement les champs textuels de la base de données avec les concepts de cette ontologie.

Cette approche permet de construire semi-automatiquement non seulement une ontologie mais aussi les annotations sémantiques sur les documents ou bases de données.

Avant de clôturer cette présentation des approches de constructions, il est bon de noter que chaque domaine a ses particularités et il est toujours intéressant d'avoir, en plus d'une approche méthodologique standard, une approche empirique (Endres-Niggemeyer et al., 1997) et de savoir tirer parti des spécificités du domaine, par exemple les caractéristiques des expressions linguistiques utilisées dans le domaine comme le montre (Golebiowska, 2002).

2.5 La méthode de spécification OntoSpec

OntoSpec est une méthode de spécification semi-informelle d'ontologies (Kassel, 2002). Les concepts en OntoSpec sont organisés en une taxinomie, suivant une relation de subsumption (spécialisation) : le fils hérite de toutes les propriétés de son père. Par exemple, le concept d'ensemble fini hérite de toutes les propriétés du concept d'ensemble. Les relations permettent de lier différents concepts entre eux.

Pour rendre compte d'une entité conceptuelle, il faut décrire les propriétés des objets dénotés par cette entité, certaines Propriétés sont Essentielles (PE) à la définition de l'entité conceptuelle, dans le sens où la suppression de cette propriété entraîne la disparition de l'entité en tant que telle. D'autres propriétés peuvent n'être vérifiées que par une partie des objets dénotés par l'entité. Guarino ne considère comme ontologiques que les propriétés nécessaires (Guarino & Giaretta, 1995). OntoSpec admet dans l'ontologie des Propriétés Incidentes (PI), c'est-à-dire vraies seulement dans le cadre applicatif. Ces propriétés servent à compléter la définition de l'entité. OntoSpec définit aussi des primitives de modélisation permettant la classification des propriétés des entités en : Conditions Nécessaires (CN), Conditions

Suffisantes (CS), Conditions Nécessaires et Suffisantes (CNS). Si l'entité est définie par des CNS, alors sa définition est complète, elle suffit pour caractériser l'entité. Tel est le cas, par exemple, de la définition du concept d'ensemble fini : un ensemble fini est un ensemble qui possède un cardinal.

Dans la définition d'un concept, le *genus* est le père dont le concept fils hérite toutes les propriétés. Le *differentia* est constitué de tout ce qui différencie le fils de son père. Par exemple, une définition du concept ensemble infini : un ensemble infini est un ensemble qui ne possède pas de cardinal. Le *genus* ici est ensemble : un ensemble infini est une sorte d'ensemble. Par contre, le concept d'ensemble infini ne rassemble que les ensembles qui ne possèdent pas de cardinal : c'est le *differentia*.

Les concepts frères qui présentent des similitudes dans leur *differentia* sont organisés en axes sémantiques (AS). Par exemple, sous le concept d'ensemble, les concepts ensemble fini, ensemble infini, ensemble dénombrable, ensemble indénombrable, sous-ensemble, sur-ensemble sont définis suivant différents axes sémantiques (cf. Figure 6).

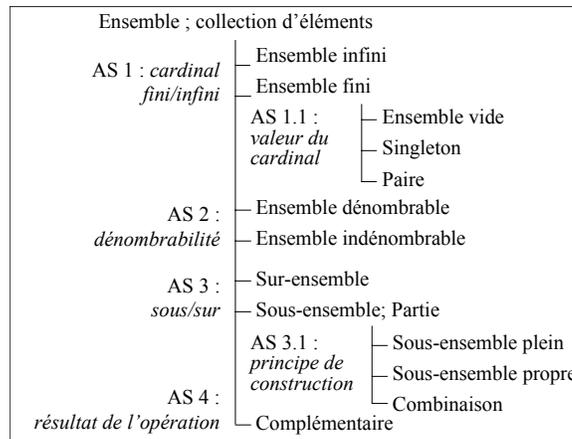


Fig. 6 : Présentation graphique de la sous-ontologie d'ensemble

La méthode de spécification OntoSpec est semi-informelle, elle propose une définition des entités conceptuelles en langue naturelle fortement structurée. La figure 7 présente des exemples de cette définition. Les étiquettes entre crochets correspondent aux distinctions effectuées entre les propriétés. Les mots soulignés représentent les relations entre concepts. Les synonymes sont séparés par des points virgules. Le terme principal est indiqué en premier. Il est fait référence à des commentaires (COM) pour des exemples (COM/EX), des contre-exemples (COM/CEX), des axes sémantiques (COM/AS) et aussi des éléments d'information d'ordre générique (COM/*). Ces commentaires apparaissent en italique.

<p>Concepts :</p> <ul style="list-style-type: none">• Ensemble ; collection d'éléments : [COM/AS] Le concept ENSEMBLE se spécialise en ENSEMBLE FINI et ENSEMBLE INFINI suivant la relation : POSSEDE CARDINAL. [COM/AS] Le concept ENSEMBLE se spécialise en ENSEMBLE DENOMBRABLE et ENSEMBLE INDENOMBRABLE suivant la relation : EST DENOMBRABLE. [COM/AS] Le concept ENSEMBLE se spécialise en SOUS-ENSEMBLE et SUR-ENSEMBLE suivant la relation : EST INCLUS DANS. [COM/AS] Le concept ENSEMBLE se spécialise en COMPLEMENTAIRE suivant la relation : A POUR RESULTAT. [COM/*] Deux ensembles sont égaux si et seulement s'ils ont les mêmes éléments.• Ensemble infini : [PE/CNS] Un ENSEMBLE INFINI est un ENSEMBLE qui ne possède pas de CARDINAL. [COM/AS] Le concept ENSEMBLE FINI se spécialise en PAIRE, ENSEMBLE VIDE et SINGLETON suivant la relation : VALEUR DU CARDINAL. [COM/EX] N est un ensemble infini.

Fig. 7 : Exemple de définitions de concepts spécifiées en OntoSpec

2.6 Evaluation d'une ontologie

Avant de publier une ontologie ou avant d'implémenter une application logicielle reposant sur une ontologie développée par d'autres personnes, il faut d'abord l'évaluer. Pour Fernandez et al. (1997) l'évaluation inclut la vérification et la validation. La vérification se rapporte au processus technique qui garantit l'exactitude d'une ontologie, de son environnement logiciel, et de sa documentation par rapport au cadre de référence (dans notre cas : le cahier des charges) pendant chaque phase et entre chaque phase du cycle de vie. La validation garantit que les ontologies, l'environnement logiciel et la documentation correspondent au système qu'ils sont censés représenter. La complétude de la spécification d'une ontologie est difficilement prouvable mais l'adéquation d'une ontologie à sa spécification est immanquablement nécessaire. L'adéquation de l'ontologie concerne les utilisateurs du système, par conséquent les intéressés doivent être impliqués dans le processus d'évaluation à chacune des phases.

Pour Uschold et Gruninger (1996), l'un des premiers aspects de la validation est de s'assurer de la cohérence. Les axiomes doivent être logiquement cohérents. La cohérence doit aussi être appliquée à la partie des définitions qui n'est pas axiomatique, telle que la documentation en langage naturel et les exemples. Pour cela les auteurs proposent de décrire formellement les spécifications de l'ontologie et de vérifier les objectifs de conceptions de l'ontologie. Les auteurs réutilisent aussi les questions de compétence qu'ils ont introduites pour la construction afin d'évaluer le consensus ontologique et de vérifier que l'ontologie est à la hauteur des besoins.

Uschold et Gruninger (1996) suggèrent plusieurs autres critères pour guider et évaluer la conception d'une ontologie :

- la clarté : les ambiguïtés doivent être minimisées sinon éradiquées, les distinctions entre concepts motivées et la compréhension des définitions facilitée par des exemples ;
- la cohérence : les axiomes doivent être logiquement consistants et les parties non axiomatiques des définitions doivent être rigoureuses et cohérentes ;

- l'extensibilité : on doit pouvoir étendre l'ontologie avec de nouveaux termes, concepts, relations, etc. ;
- des hypothèses ontologiques minimales : l'ontologie doit faire aussi peu d'hypothèses que possible sur le monde qu'elle modélise ;
- un biais de codage minimal : les choix de représentation faits pour des raisons d'implémentation doivent être minimaux.

(Noy et Hafner, 1997) proposent aussi des critères de comparaison et d'indexation des ontologies :

- Critères généraux : but, domaine, facilité d'intégration, taille, formalisme, plateforme et langage d'implantation, accès publique
- Critères de conception : méthode de conception, méthode d'évaluation
- Critères taxonomiques : organisation globale de la taxonomie, nombre de taxonomies, contenu (choses, processus, relations, propriétés), modélisation du temps, division du niveau supérieur, densité et enchevêtrement de la taxonomie.
- Critères de structuration interne des concepts et relations : existence d'une structure interne, propriété et rôles, autres types de relations entre concepts, présence d'une relation tout-partie
- Critères d'axiomatisation : axiomes explicites, représentation des axiomes
- Critères d'inférences : mécanismes d'inférences, logiques d'autre supérieures à un
- Critères applicatifs : mécanismes de recherche, interface utilisateur, applications de l'ontologie
- Critères de contribution : points forts, points faibles.

La vérification et la validation d'une ontologie sont aussi très liées à la dimension collective de l'ontologie et à son statut de consensus. Cela suggère que la création, la maintenance et l'acceptation de l'ontologie soient supportées par des collecticiels utilisés par la communauté des intéressés. De plus, d'un point de vue coopératif, deux premiers problèmes se posent, équivalents à ceux rencontrés par les experts de Cyc (Lenat et Guha, 1990) :

- "Ne pas se marcher sur les pieds" i.e. ne pas se tromper dans l'utilisation des notions entrées par une autre personnes ;
- "Ne pas se passer à côté l'un de l'autre sans se voir" i.e. ne pas rentrer plusieurs fois les mêmes notions mais sous des identifications différentes ; ceci peut être évité en recherchant activement les analogies dans l'ontologie.

Le travail collectif sur l'ontologie était, par exemple, un aspect du serveur Ontolingua qui proposait de gérer le travail de groupe et les mécanismes de notification des changements effectués par les autres participants (Farquhar et al., 1996). De même, (Euzenat, 1996) propose d'utiliser la revue par les pairs (comme

pour une publication) lors de la contribution d'une nouvelle connaissance à une base de connaissances ; cette contribution introduit un protocole pour gérer les conflits et assurer les consensus.

Enfin il est important de documenter l'ontologie : toutes les hypothèses importantes sous-jacentes, toutes les notions modélisées, la logique de conception, les primitives de la méta-ontologie si elle ne repose pas sur un standard.

2.7 Evolution d'une ontologie

(Stojanovic, 2004) définit l'évolution d'une ontologie comme une adaptation opportune de cette ontologie aux changements se produisant et la propagation consistante de ces changements aux artefacts dépendant de l'ontologie (*"Ontology Evolution is the timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artefacts."*).

Il existe plusieurs recherches sur l'évolution de schéma dans les bases de données. (Roddick, 1995) présente une excellente étude sur la gestion des versions (versioning) des schémas et précise quelques définitions concernant l'évolution du schéma. En adaptant ces définitions aux ontologies (Stojanovic, 2004) distingue la gestion, la modification, l'évolution et la gestion de versions de l'ontologie:

- Gestion de l'ontologie : c'est l'ensemble des méthodes et techniques nécessaires pour utiliser efficacement des variantes multiples d'une ontologie, éventuellement à partir des sources différentes en vue de résoudre des tâches différentes. Par conséquent, un système de gestion d'ontologie offre un cadre pour créer, modifier, gérer des versions, questionner, et stocker des ontologies. Il devrait permettre à une application de fonctionner avec une ontologie sans s'inquiéter de la manière dont l'ontologie est stockée et accédée, dont les requêtes sont traitées, etc.
- Modification de l'ontologie : quand un système de gestion de l'ontologie permet des changements sur l'ontologie utilisée, sans considérer les problèmes éventuels de consistance.
- Evolution de l'ontologie : quand un système de gestion de l'ontologie facilite la modification d'une ontologie en préservant sa consistance.
- Gestion des versions de l'ontologie : quand un système de gestion de l'ontologie permet la manipulation des changements de l'ontologie en créant et en gérant ses différentes versions.

Quelques travaux concernent le changement et l'évolution de l'ontologie, en particulier les contributions de Michel Klein et de Ljiljana Stojanovic. Nous allons examiner les deux approches de ces deux auteurs dans les parties suivantes.

2.7.1 Processus d'évolution d'une ontologie

(Stojanovic, 2004) définit diverses notions nécessaires pour l'évolution de l'ontologie, la consistance de l'ontologie et la taxonomie des changements de l'ontologie. (Stojanovic et al., 2002) souligne tout d'abord les conditions satisfaisantes par un système d'évolution d'ontologie puis dérive un processus d'évolution de six phases qui satisfait ces conditions. Le processus analyse systématiquement les causes et les conséquences des changements, et assure la consistance de l'ontologie ainsi que ses parties dépendantes après la résolution de ces changements.

Ensuite, les auteurs présentent la complexité sur la résolution de changement, une approche procédurale et déclarative pour résoudre des changements de l'ontologie, les solutions pour propager des changements dans l'environnement local et distribué. Ils discutent également les avantages et inconvénients concernant les besoins des ontologistes utilisant un système d'évolution d'ontologie.

(Stojanovic et al., 2002) et (Stojanovic, 2004) identifient un processus possible d'évolution en six phases : (1) capture du changement, (2) représentation du changement, (3) sémantique du changement, (4) propagation du changement, (5) implémentation du changement et (6) validation du changement.

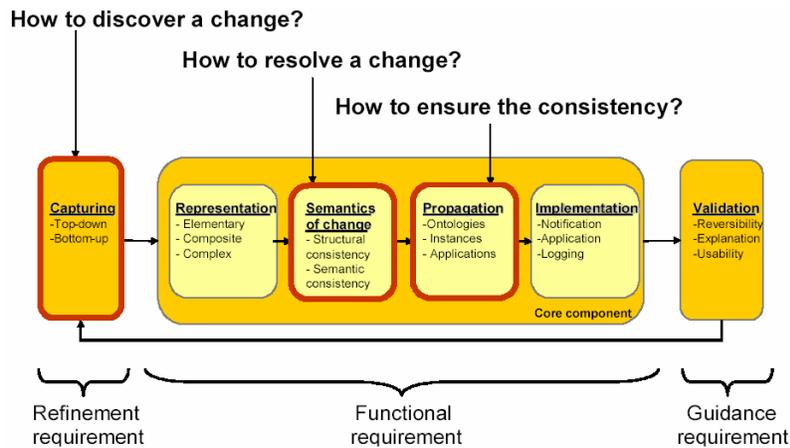


Fig. 8 - Processus d'évolution de l'ontologie en six phases (Stojanovic, 2004)

(1) Capture du changement

Le processus d'évolution de l'ontologie commence à capturer les changements à partir des exigences explicites (approche descendante) ou bien du résultat des méthodes de découverte des changements (approche ascendante) qui induisent des changements à partir des données existantes. (Stojanovic, 2004) définit trois types de découverte des changements : dirigée par la structure (structure-driven), dirigée par l'utilisation (usage-driven) et dirigée par les données (data-driven). Considérant que les changements dirigés par la structure peuvent être déduits de la structure de l'ontologie elle-même, les changements dirigés par l'utilisation sont découverts par des patterns d'utilisation (usage patterns) créés à travers le temps. Les changements dirigés par les données sont générés par des modifications sur l'ensemble de données fondamentales, tel que des documents textuels ou une base de données, représentant la connaissance modélisée par une ontologie.

Pour la phase de capture du changement, (Stojanovic, 2004) propose aussi une approche de découverte des changements en analysant les différentes sources de données liées au comportement de l'utilisateur qui incluent des informations sur ses goûts, ce qu'il n'aime pas, ses préférences ou la manière dont il se comporte. En se basant sur l'analyse de ces informations, un ontologiste peut suggérer de faire certains changements dans l'ontologie pour mieux l'adapter aux besoins des utilisateurs. Certaines demandes d'adaptation pourraient être indiquées explicitement, par exemple le besoin d'un nouveau type de client dû à un changement de la stratégie commerciale. D'autres changements sont représentés implicitement et pourraient être découverts grâce à l'utilisation de l'application. Ces changements découverts sont très importants pour optimiser la performance d'une application.

(2) Représentation du changement

Pour résoudre les changements, ils doivent être identifiés et représentés dans un format approprié. Les changements peuvent être représentés à différents niveaux de granularité, en distinguant par exemple : des changements élémentaires (i.e. modifications isolées d'une ontologie), des changements composés (i.e. représentant un groupe de changements élémentaires appliqués ensemble) et des changements complexes (i.e. pouvant être décomposés en n'importe quelle combinaison comprenant au moins deux changements élémentaires et composés) (Stojanovic, 2004).

(3) Sémantique du changement

Cette phase permet de résoudre des changements induits d'une manière systématique en assurant la consistance de toute l'ontologie. La tâche de la phase de sémantique du changement permet de résoudre des changements induits d'une manière systématique, assurant la consistance de toute l'ontologie. Selon le dictionnaire standard d'ordinateur de l'IEEE, la consistance est le degré d'uniformité, de standardisation, et d'absence de contradiction parmi les parties d'un système ou d'un composant. Du point de vue de la logique, la consistance est un attribut du

système (logique) satisfaisant le fait qu'aucun des faits déductibles du modèle n'est contraire aux autres.

Tout d'abord, (Stojanovic, 2004) donne une définition d'un modèle de consistance d'une ontologie simple : celle-ci est consistante par rapport à son modèle si et seulement si elle préserve les contraintes définies pour le modèle d'ontologie de base. Stojanovic et al. définissent le modèle de consistance de l'ontologie comme l'union de 16 invariants, de 2 contraintes souples et de 4 contraintes définies par l'utilisateur. Les invariants sont des règles de consistance valables pour chaque ontologie. Chaque changement d'une ontologie doit maintenir l'exactitude des invariants mais une ontologie ne doit pas nécessairement satisfaire toutes les contraintes souples et les contraintes définies par l'utilisateur.

La phase de sémantique du changement peut être réalisée en complétant les changements exigés avec un ensemble de changements supplémentaires, qui garantit le transfert d'une ontologie consistante initiale à un autre état consistant. Le défi de cette phase est comment trouver les changements supplémentaires qui préservent la consistance de l'ontologie. Si un ontologiste peut facilement résoudre une certaine partie de la modification globale, il n'est toutefois pas facile de générer manuellement des changements supplémentaires nécessaires pour garder la consistance. D'ailleurs, si cela était laissé à la charge de l'ontologiste, le processus d'évolution serait trop inexact et long, car il est peu réaliste que les humains puissent comprendre toute l'ontologie et ses interdépendances. Cela explique le besoin de méthodes pour réaliser cette tâche automatiquement. (Stojanovic, 2004) adapte à l'évolution d'ontologie deux approches inspirées de la communauté Base de données, pour assurer la consistance : l'approche procédurale et l'approche déclarative de l'évolution du schéma.

- *Approche procédurale* : cette approche est basée sur les contraintes qui définissent la consistance d'un schéma, et les règles définies qui doivent être suivies pour maintenir des contraintes satisfaites après chaque changement. L'approche procédurale pour la phase de sémantique du changement est réalisée par un mécanisme procédural qui incorpore la sémantique des changements d'ontologie. Elle permet de fournir la réponse à une classe large des demandes d'utilisateurs pour un changement. Puisqu'un changement d'ontologie peut être résolu de plusieurs manières, cela signifie que différents ensembles de changements supplémentaires peuvent être générés. Chacun de ces ensembles de changements entraîne à une différente ontologie consistante finale. Ainsi il serait trop restrictif de forcer des ontologistes à résoudre un changement d'une manière unique. Par conséquent, les ontologistes ont besoin d'un mécanisme pour contrôler les changements selon des stratégies accomplissant certaines contraintes (par exemple nombre minimal des changements...). Pour ce faire, la notion des stratégies d'évolution (Stojanovic et al., 2002) permet à un ontologiste de personnaliser le processus de la génération des changements additionnels selon ses besoins. Ces stratégies d'évolution constituent une méthode pour "chercher" une ontologie consistante qui satisfasse les besoins d'un ontologiste. Afin de générer des changements supplémentaires, Stojanovic définit formellement le changement d'ontologie et détermine les dépendances entre

les changements d'ontologie (relations cause-effet) qui construit le graphe de dépendance de changement. Ce graphe de dépendances de changement peut être également représenté sous forme de matrice de dépendances qui en est une vue simplifiée. (Stojanovic, 2004) présente également un ensemble de notions aidant un ontologiste à modifier une ontologie selon ses préférences : un point de résolution représente un dilemme qui pourrait se produire pendant la résolution de changement, une stratégie d'évolution élémentaire est un ensemble de manières possibles pour résoudre un point de résolution, la méthode de résolution est un ensemble de stratégies d'évolution élémentaires possibles pour résoudre un point de résolution particulier. Une stratégie d'évolution est un ensemble de couples (point de résolution, stratégie d'évolution élémentaire définie pour ce point de résolution).

Ainsi, pour résoudre un changement, le processus d'évolution doit déterminer des réponses à plusieurs points de résolution pendant la résolution de changement où la prise de différents chemins produira différents résultats. Chaque réponse possible correspondant à chaque point de résolution est une stratégie d'évolution élémentaire. La politique commune se compose d'un ensemble de stratégies d'évolution élémentaires, donnant chacune une réponse pour un point de résolution, est une stratégie d'évolution. Les points de résolution et leurs stratégies d'évolution élémentaires couvrent toutes les manières possibles qu'un ontologiste peut suivre pour appliquer des changements sur l'ontologie. Ce processus s'appelle également le processus dirigé par l'utilisateur d'évolution de l'ontologie (Stojanovic et al., 2002), où l'utilisateur est un ontologiste qui peut spécifier une stratégie d'évolution afin de réaliser l'évolution de l'ontologie selon ses besoins.

- *Approche déclarative* : cette approche est basée sur l'ensemble d'axiomes (accompagné un mécanisme d'inférence) qui formalise la dynamique de l'évolution.

Afin de présenter l'approche déclarative, (Stojanovic, 2004) prend encore l'exemple de supprimer un concept dans une ontologie. Par exemple, après une suppression de concept, ses sous-concepts peuvent être aussi bien préservés ou supprimés. S'ils sont préservés, ils doivent être rattachés aux concepts parents du concept supprimé ou au concept racine. Cependant, il se pourrait parfois qu'un utilisateur veuille garder certains sous-concepts et supprimer les autres. Le nombre de solutions possibles augmente alors considérablement si le changement est plus complexe. Chaque solution répond aux besoins de certains ontologistes, mais pas de tous. Puisque les besoins d'un ontologiste ne peuvent pas être prévus, il est également difficile de déterminer exactement de quelle manière résoudre une requête. D'ailleurs, si nous suivons l'approche procédurale, la détermination de la bonne stratégie d'évolution qui satisfasse le besoin de l'ontologiste

exige une compréhension profonde sur la stratégie d'évolution. Par conséquent, un système d'évolution de l'ontologie utile et efficace doit résoudre les deux problèmes suivants (Stojanovic et al., 2003) :

- Comment un ontologiste peut-il spécifier sa demande pour un changement de manière plus facile, plus exacte et plus déclarative sans avoir besoin une connaissance profonde sur l'ontologie et l'évolution de l'ontologie? Cela est contraire à toutes les approches existantes, où l'ontologiste peut seulement choisir un changement dans un ensemble prédéfini des changements d'ontologie, qui ne couvrent pas tous les besoins des ontologistes.
- Comment cette demande peut-elle être réalisée? Quoique les stratégies d'évolution offrent plus de flexibilité en permettant à un ontologiste de contrôler et personnaliser la méthode de résolution d'un changement, elles ne couvrent pas toutes les possibilités. La solution n'est donc pas de spécifier à l'avance les manières possibles pour résoudre un changement, mais de permettre au système de calculer toutes les manières qui satisfont aux besoins des ontologistes.

Tout d'abord, contrairement aux systèmes d'évolution de l'ontologie existants où un ontologiste spécifie sa demande par un ensemble de changements d'ontologie à exécuter, (Stojanovic, 2004) permet d'exprimer une demande par un ensemble de changements d'ontologie à exécuter et un ensemble de changements d'ontologie ne devant pas être exécutés. Cette approche permet à un ontologiste de spécifier une demande d'une manière déclarative et d'assurer sa résolution.

En considérant l'évolution de l'ontologie comme une tâche de résolution des problèmes de reconfiguration-conception, le problème se réduit à une recherche sur un graphe dont les noeuds sont des ontologies évolutives et les arcs représentent les changements qui transforment du noeud source à une autre cible (Stojanovic et al., 2003). Les auteurs définissent formellement le graphe d'évolution représentant la sémantique des changements. Afin de résoudre un problème, il faut préciser l'espace de recherche et appliquer un algorithme de recherche dans cet espace. Puisque ce problème de recherche n'est pas résolu facilement en général, il faut exploiter des connaissances heuristiques qui guident le processus de recherche des solutions désirées. La recherche est guidée par les contraintes fournies partiellement par un ontologiste et partiellement par un ensemble de règles définissant la consistance d'ontologie. (Stojanovic, 2004) fournit quelques types d'heuristiques pour améliorer la recherche telle que l'heuristique de localité, l'heuristique de détection de cycle et l'heuristique définie par l'utilisateur...

Afin de comparer les approches proposées pour la sémantique du changement (Stojanovic, 2004) définit un critère subjectif : l'efficacité qui est le degré avec lequel un système exécute la demande désirée d'un changement en consommant le minimum

de ressources (par exemple temps, espace, etc.). Ce critère est choisi pour évaluer comment un système d'évolution de l'ontologie satisfait les besoins des ontologistes utilisant ce système pour modifier des ontologies.

- L'approche procédurale peut être considérée comme une approche basée sur des commandes puisque l'ontologiste doit comprendre la sémantique des changements d'ontologie et donc doit faire attention dans son choix des changements à utiliser. Le problème est que cette approche se concentre sur le processus d'édition plutôt que sur le résultat d'édition. Pour cette raison, l'efficacité des spécifications d'une requête dépend de la complexité de cette requête. Si une requête peut être représentée sous forme d'un changement (élémentaire ou composé), alors l'efficacité des spécifications de cette requête est grande. Cependant, pour des requêtes plus complexes, ce type d'efficacité est très pauvre.
- L'approche déclarative permet à un ontologiste de modifier une ontologie selon les besoins, puis de calculer tous les changements supplémentaires. Cette approche se focalise sur le résultat d'édition plutôt que sur le processus d'édition. L'avantage est que l'ontologiste peut spécifier sa requête d'une manière déclarative. Par conséquent, l'efficacité des spécifications de requête est toujours élevée. En comparaison de l'approche procédurale, l'approche déclarative est plus efficace (par rapport aux spécifications de la requête).

(4) *Propagation du changement*

La phase de propagation du changement du processus d'évolution de l'ontologie doit, après avoir effectué une mise à jour de l'ontologie, assurer la consistance des artefacts dépendant de l'ontologie : les ontologies dépendantes, les instances ainsi que les programmes d'application concernant cette ontologie.

- L'impact des changements sur les ontologies dépendantes : une mise à jour d'ontologie pourrait corrompre d'autres ontologies qui dépendent de l'ontologie modifiée. Ces ontologies sont construites à partir de l'ontologie modifiée ou elles l'importent. Ce problème peut être résolu par une procédure récursive en appliquant des changements sur ces ontologies afin de préserver leur consistance conceptuelle, structurale et comportementale.
- L'impact des changements sur les instances ontologiques : quand l'ontologie est modifiée, les instances doivent être changées d'une manière que l'ontologie et les instances restent conformes l'une à l'autre. Fondamentalement, si l'ontologie est modifiée alors ses instances doivent être transformées pour se conformer à l'ontologie modifiée.
- L'impact des changements sur les applications : les changements de l'ontologie pourront influencer les applications basées sur l'ontologie modifiée.

En général, on peut distinguer deux dimensions affectant le problème global d'évolution de l'ontologie : d'une part, le nombre d'ontologies ayant évolué et d'autre part, la localisation physique des ontologies évolutives. (Stojanovic, 2004) élargit le

problème de l'évolution d'une ontologie simple à celui de l'évolution de multiples ontologies dépendantes centralisées (i.e. dépendant d'une autre ontologie résidant dans le même noeud sur le réseau) puis distribuées (i.e. dépendant d'une ontologie résidant dans un noeud différent sur le réseau).

(5) *Implémentation du changement*

Le rôle de cette phase est (i) d'informer un ontologiste de toutes les conséquences d'une requête de changement, (ii) d'appliquer tous les changements (demandés et dérivés) et (iii) contrôler sur les changements exécutés.

(6) *Validation du changement*

Cette phase permet la justification des changements exécutés et de les annuler suite à la demande de l'utilisateur. Deux approches sont possibles pour vérifier la consistance de l'ontologie (Stojanovic, 2004):

- la vérification a posteriori, où les changements sont d'abord exécutés, puis on vérifie si l'ontologie mise à jour satisfait les contraintes de consistance.
- la vérification a priori, qui définit un ensemble de préconditions relatives à chaque changement. Pour chaque changement, la consistance sera maintenue si (1) l'ontologie est consistante avant une mise à jour et (2) les préconditions sont satisfaites.

2.7.1.1 Framework pour l'évolution d'ontologie

(Klein & Noy, 2003) et (Klein, 2004) définissent un framework pour l'évolution de l'ontologie. Ils définissent un changement de l'ontologie comme une action sur l'ontologie entraînant l'obtention d'une ontologie différente de la version originale. (Klein, 2004) distingue plusieurs types de changements d'ontologie :

- Changement conceptuel: un changement dans la conceptualisation;
- Changement de spécification : un changement des spécifications d'une conceptualisation ;
- Changement de représentation: un changement de la représentation des spécifications d'une conceptualisation.

Ces changements sont alors représentés de différentes manières à différents niveaux de détail :

- les ontologies seulement : l'ancienne version Vanc et la nouvelle version Vnouv de l'ontologie ; cela ne fournit aucune information explicite de changement, mais cela peut être utilisé comme la base pour trouver d'autres informations de changement ;

- log des changements : une liste des changements appliqués sur Vanc pour obtenir en résultat Vnou ;
- différence structurelle : une projection entre les concepts et les propriétés dans une version et leurs équivalents dans la nouvelle version, ainsi qu'une liste des concepts ajoutés et supprimés ;
- relations conceptuelles : une spécification explicite des relations conceptuelles entre les concepts dans Vanc et les concepts correspondants dans Vnou.

(Klein & Noy, 2003) propose un framework pour l'évolution de l'ontologie en décrivant ses éléments principaux :

- Méta-ontologie des opérations de changement : cette ontologie des opérations de changement spécifie divers changements standards dans une ontologie.
- Opérations de changement complexes : elles se composent de multiples opérations de base, ou incorporent des connaissances additionnelles sur les changements. Les opérations complexes fournissent un mécanisme pour regrouper un certain nombre d'opérations de base qui constituent ensemble une entité logique.
- Ensemble des transformations $T(Vanc, Vnou)$: il fournit un ensemble d'opérations de changement qui spécifie comment Vanc peut être transformé en Vnou. Il utilise les opérations de l'ontologie des changements. Celles-ci peuvent être effectuées dans n'importe quel ordre, sauf une exception : toutes les opérations qui créent de nouvelles classes, propriétés, et instances sont exécutées d'abord. Un ensemble des transformations n'est pas unique, car il existe souvent plusieurs façons de construire un ensemble de transformations pour un changement spécifique. (Klein, 2004) définit un ensemble minimal de transformations comme un ensemble d'opérations suffisant et nécessaire pour transformer Vanc en Vnou. L'ensemble minimal des transformations fournit une spécification condensée des changements et peut être utilisé pour re-exécuter le changement et pour dériver l'information supplémentaire.
- Template pour la spécification de changement : Le template peut être utilisé pour décrire comment les deux versions d'ontologies sont reliées.

2.8 Synthèse globale sur le cycle de vie d'une ontologie

Pour Fernandez *et al.* (1997) la vie d'une ontologie passe par les états suivants : spécifications, conceptualisation, formalisation, intégration, implantation, et maintenance (Figure 9).

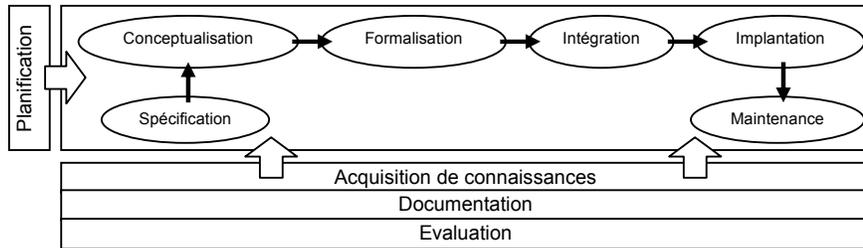


Fig. 9 – Cycle de vie pour Fernandez et al. (1997)

De façon générale, la connaissance à un cycle de vie passant par des étapes : de production, de mémorisation, de diffusion et d'utilisation. (Figure 10).

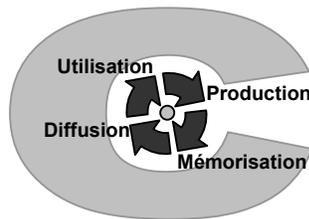


Fig. 10 – Cycle de vie simplifié des connaissances.

La Figure 11 résume notre état de l'art sur le cycle de vie complet d'une ontologie (Gandon, 2002a) (Gandon, 2002b) qui étend celui proposé dans (Dieng et al, 2001) ; brièvement, il rassemble sept activités :

- L'activité de *Détection* des besoins lors de la conception et l'activité d'*Evaluation* lors du cycle de vie ont en commun un certain nombre de tâches et de méthodologies de recueil (entretiens, questionnaires, sondages) et d'identification/analyse (ex : modélisation par scénarii). En complément la phase de détection des besoins demande un état des lieux initial approfondi car elle ne peut reposer sur des études précédentes ou des retours d'utilisation comme c'est le cas pour l'évaluation.
- La phase de *Conception initiale* et la phase d'*Evolution* lors du cycle de vie ont en commun un certain nombre d'activités : spécification des solutions ; acquisition des connaissances nécessaires ; conceptualisation/modélisation ; formalisation ; intégration de ressources existantes ; implantation.
- La phase de *Diffusion* s'intéresse au déploiement et mise en place de l'ontologie particulièrement dans le cas de solutions distribuées mais aussi du point de vue de l'information des utilisateurs sur la mise à disposition ou mise à jour de l'ontologie.

- La phase d'*Utilisation* regroupe toutes les activités reposant plus ou moins directement sur la disponibilité de l'ontologie, par exemple : l'annotation, la résolution de requête, l'application d'inférences, etc.
- L'activité permanente de *Gestion & Planification* souligne qu'il est important d'avoir un travail de suivi et une politique globale pour détecter ou déclencher, préparer et évaluer les itérations du cycle et s'assurer du maintien du consensus.

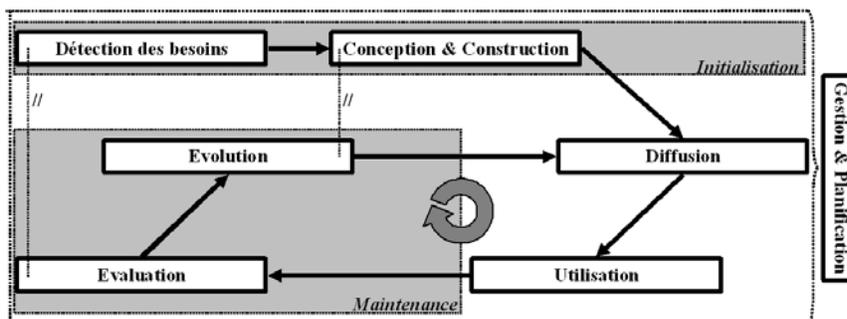


Fig. 11 – Schéma résumant le cycle de vie

2.9 Comparaison et intégration d'ontologies

Nous allons présenter plusieurs approches pour la comparaison et l'intégration d'ontologies.

2.9.1 Approches de matching

On peut distinguer deux types de «matching» : le «matching» de schéma et le «matching» d'ontologie. Les résultats de la recherche dans le domaine du «matching» de schémas peuvent être appliqués dans le domaine du «matching» d'ontologies. Une revue des approches déjà existantes a été effectuée dans le cadre du réseau d'excellence Knowledge Web et est présentée dans (Euzenat, 2004).

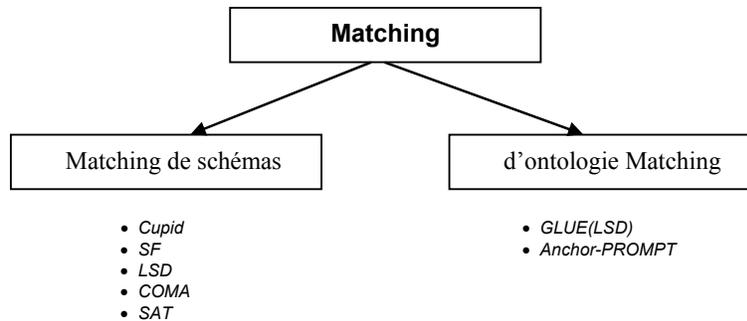


Fig. 12 – Types d’approches de «matching» et approches correspondantes

Similarity Flooding (SF) est un algorithme de «matching» de graphes (Melnik et al, 2001). Son application au «matching» de schéma est présentée dans (Melnik et al, 2002). SF convertit les schémas (SQL DDL, XML) en des graphes étiquetés orientés et puis il applique le calcul du point-fixe pour déterminer les nœuds similaires dans les graphes. Cette approche est basée sur une comparaison très simple des chaînes des caractères des noms des nœuds pour calculer des «mappings» initiaux, puis ces «mappings» sont fournis au module de «matching» structurel de SF. Bien que SF ait appliqué une nouvelle approche orientée structurelle basée sur l’intuition que les éléments de deux schémas distincts sont similaires quand leurs éléments adjacents sont déjà similaires pour propager la similitude de deux éléments à leurs voisins respectifs, il se fonde principalement sur des étiquettes des arcs dans les graphes. S’il n’y a aucune étiquette pour des arcs dans le graphe ou si ces étiquettes sont presque identiques, l’algorithme ne fonctionne pas bien. Sans utilisation d’un dictionnaire terminologique externe tel que WordNet (Miller, 1995), l’algorithme ne donne pas de bons résultats de «matching» au niveau linguistique dans la première phase, qui seront fournis à la deuxième phase, ce qui influencera les résultats finaux. En conséquence, l’application de SF dans le domaine de «matching» d’ontologie (où il est difficile d’établir les graphes étiquetés) posera des problèmes difficiles à approfondir.

Contrairement à SF, Cupid (Madhavan et al, 2001) a proposé une approche de «matching» combinant un module sophistiqué de «matching» des noms et un algorithme de «matching» structurel. Cupid est également un algorithme de «matching» des schémas génériques mais sa phase de «matching» structurel est principalement basée sur la similitude des éléments atomiques dans les graphes (c.-à-d. des feuilles). Ainsi s’il y a des différences significatives dans la structure des graphes donnés, Cupid ne peut pas trouver des «mappings» corrects. Par exemple, si un concept situé à la place d’une feuille dans le premier graphe (schéma) se trouve, dans le deuxième, à la place d’un élément non-feuille qui est la racine d’un sous-graphe, Cupid ne détectera pas qu’il s’agit du même concept.

GLUE (Doan et al, 2002) est la version évoluée de LSD (Doan et al, 2002) dont le but est de trouver semi-automatiquement des «mappings» entre des schémas pour l'intégration de données. Comme le LSD, GLUE utilise la technique d'apprentissage pour trouver des «mappings». Dans GLUE, il y a plusieurs modules d'apprentissage (learners), qui sont entraînés par des instances des ontologies. Après la phase d'apprentissage, différents modèles caractéristiques des instances et des règles de «matching» pour les éléments simples du schéma cible sont découverts. Les prévisions de différents modules de «matching» (matchers) sont combinées par un méta-module de matching, et de celui, le résultat final des «mappings» sera déduit. Un inconvénient de cette approche est qu'elle se fonde principalement sur les instances de données des ontologies, qui ne sont pas toujours abondamment disponibles pour plusieurs ontologies. Un autre inconvénient est que l'ontologie est modélisée comme une taxonomie des concepts et que chaque concept a quelques attributs. Avec cette organisation, GLUE n'emploie pas des informations contenues dans la taxonomie (hiérarchie) des relations. GLUE fait également une utilisation modeste des informations sur la taxonomie des concepts.

Anchor-PROMPT (Noy & Musen, 2001) construit un graphe étiqueté orienté représentant l'ontologie à partir de la hiérarchie des concepts (appelés classes dans l'algorithme) et de la hiérarchie des relations (appelées slots dans l'algorithme), où les noeuds dans le graphe sont des concepts et des arcs sont des relations dénotant des liens entre les concepts (les étiquettes des arcs sont les noms des relations). Une liste initiale des paires d'ancres (des paires de concepts correspondants) définies par les utilisateurs ou automatiquement identifiées par le «matching» lexicologique sert d'entrée à l'algorithme. Anchor-PROMPT analyse alors les chemins dans les sous-graphes limités par les ancres et il détermine quels concepts apparaissent fréquemment en positions similaires sur les chemins similaires. En se basant sur ces fréquences, l'algorithme décide si ces concepts sont sémantiquement des concepts similaires. Cependant, Anchor-PROMPT ne cherche que des «mappings» des concepts, pas des «mappings» des relations. En outre, il emploie des noms de relation pour des étiquettes sur les arcs et la comparaison des chaînes de caractères de ces étiquettes n'est qu'une comparaison simple. Ainsi si les noms de relation sont différemment définis, l'algorithme ne fonctionnera pas bien. Les résultats retournés par l'algorithme seront également limités si les structures des ontologies sont différentes (par exemple l'une est profonde avec beaucoup de concepts au milieu, et l'autre est peu profonde). L'algorithme aura des problèmes pour fonctionner correctement si une hiérarchie a seulement quelques niveaux et si la plupart des relations sont associées aux concepts au-dessus de la hiérarchie.

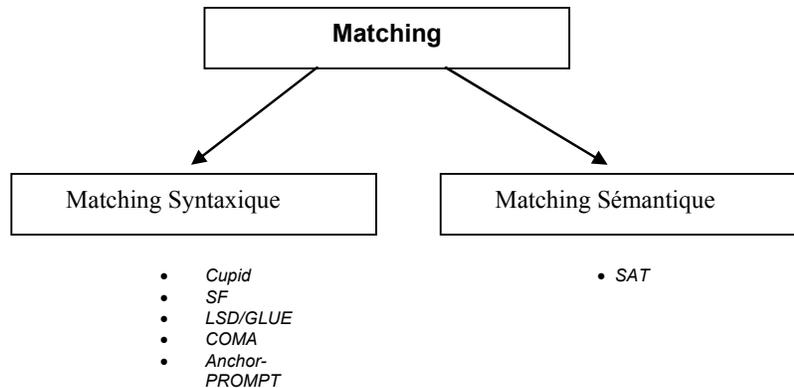


Fig. 13 – «matching» syntaxique vs «matching» sémantique

SAT (Giunchiglia & Shvaiko, 2003) prend en entrée deux graphes de concepts, et produit en sortie des liens entre les concepts tels que l'équivalence, le recouvrement (overlapping), la disparité (mismatch), le lien « plus général » ou « plus spécifique ». L'idée principale de cette approche est d'utiliser la logique pour coder le concept d'un nœud dans le graphe et d'appliquer SAT pour trouver des liens. Le concept associé à un nœud, qui est alors transformé en formule propositionnelle, est la conjonction de tous les concepts des nœuds sur le chemin de la racine du graphe jusqu'au nœud en question. Le concept d'un nœud est extrait à partir de WordNet (Miller, 1995). Le lien entre deux concepts qui doit être prouvé est également converti en formule propositionnelle. Le résolveur SAT calculera sur l'ensemble des formules propositionnelles pour vérifier si le lien supposé est vrai ou non. Une des difficultés principales de cette approche est le choix du meilleur sens entre les sens d'un terme (d'un nœud) dans le graphe à partir de l'annuaire lexicologique WordNet. En consultant WordNet, une liste des divers sens avec leur ensemble de synonymes est retournée pour chaque terme demandé. La question est quel sens sera le sens le plus proche du sens du terme (du nœud) dans le contexte où le nœud est placé dans le graphe. SAT n'a pas encore résolu ce problème de manière satisfaisante.

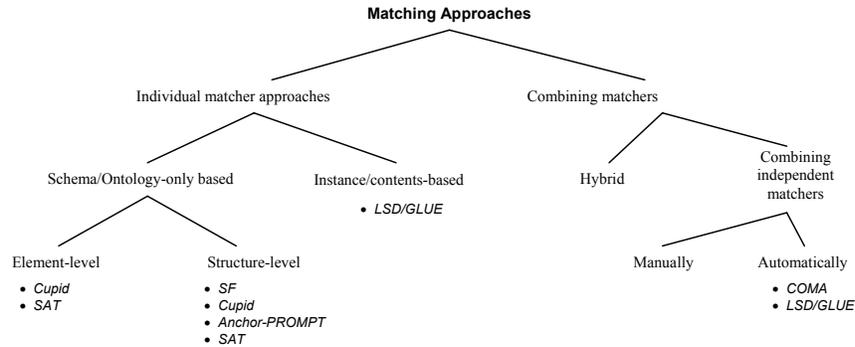


Fig 14 – Typologie des approches de « matching »

(Cohen et al, 2003) compare des métriques de distance entre des chaînes de caractères. Les méthodes évaluées sont les méthodes appliquées dans les domaines de comparaison des mots, des termes, des expressions et des paragraphes. Les fonctions métriques appliquées pour des chaînes de caractères courtes (ex. pour comparer des noms de gens) sont des fonctions métriques de distance ou de similarité telles que Levenstein, Smith-Waterman, Monger-Elkan (basées sur un modèle « edit-distance », modèle dans lequel la distance est le coût de la meilleure séquence des opérateurs de modification pour transformer une chaîne de caractères vers une autre, un coût étant affecté à chaque opération), ou Jaro, Jaro-Winkler (qui ne sont pas basées sur le modèle edit-distance, mais se basent sur le nombre et l'ordre des caractères communs entre deux chaînes). Pour comparer des chaînes de caractères plus longues, des ensemble de mots ou des paragraphes, on applique des métriques de distance basées sur des tokens (mots) telles que Jaccard, TF/IDF, Jensen-Shannon, Fellegi-Sunter. Les méthodes hybrides sont aussi évaluées, telles que Monge-Elkan, SoftTF/IDF. Elles sont la combinaison des méthodes au-dessus. (Cohen et al, 2003) conclut que la méthode la plus performante est la méthode hybride SoftTF/IDF combinant la méthode TF/IDF (laquelle est beaucoup utilisée dans la recherche d'information) avec la méthode Jaro-Winkler.

MULTIKAT (Dieng & Hug, 1998) est un système implémenté en C++ au-dessus de la plate-forme de gestion de graphes conceptuels COGITO (Haemmerlé, 1992), et qui permet de comparer deux ontologies sur le même domaine représentées dans le formalisme des graphes conceptuels (Sowa, 1984). MULTIKAT comprend un algorithme de construction d'une hiérarchie commune de types de concepts et d'une hiérarchie commune de types de relations, à partir de deux ontologies. Cet algorithme repose sur la comparaison de types de concepts et de types de relations en deux étapes: la première exploite les noms des types de concepts et leurs synonymes, la seconde est basée sur des fonctions de similarité, et la comparaison des contextes des types de concepts à comparer (i.e. leurs parents, leurs enfants, leurs frères). La construction du graphe fusionné repose sur la stratégie de fusion choisie par

l'utilisateur : par exemple, la stratégie de plus grande spécialisation (resp. généralisation) ou de la plus grande compétence ou du consensus.

L'algorithme ASCO (Bach & Dieng-Kuntz, 2004) permet de comparer deux ontologies. Il trouve des «mappings» grâce un processus en 2 phases : la phase linguistique et la phase structurelle. Dans la première phase, la valeur de similarité entre deux entités (tels que des concepts ou des relations) provenant des deux ontologies est calculée à partir de différentes informations disponibles : leurs noms, leurs étiquettes (labels) qui fournissent une version compréhensible par un humain du nom du concept ou de la relation, et leurs descriptions. Le calcul de la valeur de similarité linguistique est effectué de plusieurs façons, par exemple par la métrique "string-distance", TF/IDF. Pour améliorer la précision du calcul, ASCO a intégré WordNet (Miller, 1995), pour exploiter les relations de synonymie ou hyperonymie entre termes. La seconde phase, la phase structurelle, exploite les informations taxonomiques dans les structures des ontologies. Elle utilise les informations heuristiques et les connaissances du domaine pour calculer les valeurs de similarité structurelle entre entités des deux ontologies. Les valeurs de similarité dans les deux phases sont combinées pour obtenir les valeurs de similarité finales entre les entités. Les «mappings» sont déduits en se basant sur ces valeurs.

2.9.1.1 Conclusions

Parmi tous ces travaux sur l'alignement d'ontologies, nous n'en avons trouvé aucun concernant directement le e-Learning. Mais de tels travaux seront utiles quand seront disponibles plusieurs ontologies pour le e-Learning.

3 Application des ontologies au cas particulier du e-Learning

3.1 Ontologies et e-Learning

Dans le cadre du e-Learning plusieurs aspects peuvent bénéficier de l'introduction d'ontologies :

- Le e-Learning implique des échanges de connaissances :
- Le e-Learning implique la conception et la maintenance de mémoires : indexation du matériel pédagogique, requêtes sur le matériel, etc.
- Le e-Learning implique une connaissance du monde de l'enseignement : descriptions des acteurs et de leurs rôles, des situations, des processus, des flots de documents, etc.

Les ontologies présentent un intérêt dès lors qu'un scénario s'intéresse à une communication entre personnes et/ou organismes en particulier lorsqu'elle est asynchrone, que les émetteurs et récepteurs ne sont pas connus à l'avance et que l'échange se fait au travers d'une application logicielle. C'est par exemple le cas pour la mise à disposition et l'accès au matériel pédagogique dans un scénario d'enseignement. Du fait de leurs expériences passées, de leur formation et de leurs besoins différents, les acteurs d'une organisation peuvent avoir des points de vue très différents sur un même sujet. Ils peuvent utiliser des jargons différents et reposer sur des concepts, des schémas et des méthodes qui, dans le meilleur des cas, peuvent se recouvrir ou, dans le pire des cas, sont incompatibles. Le manque d'une compréhension partagée nuit à la communication, l'interopérabilité et la réutilisabilité. Il apparaît dès lors intéressant qu'une solution logicielle aide à réduire, voire éliminer, la confusion conceptuelle et terminologique et tendre vers une compréhension partagée.

Le développement et l'implantation d'une représentation explicite d'une compréhension partagée (c'est à dire, pour Uschold et Gruninger, une ontologie) dans un domaine donné, peut améliorer la communication, qui à son tour permet une plus grande réutilisation, un partage plus large et une interopérabilité plus étendue. L'ontologie fournit ce cadre unificateur et fournit des primitives descriptives et structurantes pour les échanges. D'autre part, dans une problématique de recherche d'information, reposer sur le vocabulaire conceptuel commun défini dans une ontologie peut aider à améliorer la précision de cette recherche d'information, en évitant des ambiguïtés au niveau terminologique et en autorisant des inférences diminuant le bruit et augmentant le rappel des moteurs de recherche.

L'ontologie est donc une théorie logique qui rend compte partiellement mais explicitement d'une conceptualisation, la conceptualisation étant une structure sémantique intensionnelle qui capture les règles implicites contraignant la structure d'un morceau de réalité. (Guarino & Giarretta, 1995)

Un concept est un constituant de la pensée (un principe, une idée, une notion abstraite) sémantiquement évaluable et communicable. L'ensemble des propriétés d'un concept s'appelle sa compréhension ou son intension et l'ensemble des êtres qu'il englobe, son extension. (Dieng et al, 2001) Dans une ontologie, les représentations des intensions sont organisées, structurées et contraintes pour exprimer une théorie logique rendant compte des relations qui existent entre les concepts. L'ontologie est l'objet capturant les expressions des intensions et la théorie visant à rendre compte des aspects de la réalité choisis pour leur pertinence dans les scénarios d'application considérés. La représentation des intensions et de la structure ontologique peut faire appel à des langages plus ou moins formels selon l'opérationnalisation envisagée pour l'ontologie. La construction formelle de l'intension donne une représentation précise et non ambiguë de la manière dont on peut concevoir son sens, ce qui permet sa manipulation logicielle et son utilisation comme une primitive de représentation de connaissances pour matérialiser la mémoire (Dieng et al, 2001).

Dans l'ontologie, les concepts sont souvent organisés en taxonomie car cette structure supporte les inférences élémentaires de classification ou identification (le fait de déterminer si quelque chose appartient à une classe) et de catégorisation (le fait d'identifier les catégories existantes). Prenons l'exemple simple d'une conversation entre un étudiant et un professeur :

- " - Vous pouvez me conseiller un livre sur les équations différentielles ?
- Il y a le manuel de cours du professeur Carman sur l'Analyse à la bibliothèque.
- Merci !"

Dans une conversation aussi banale, l'étudiant a généralisé sa requête au concept de 'livre', qui représente la catégorie la plus abstraite recouvrant toutes les formes de réponses acceptables pour lui. Le professeur, sans même y prêter réellement attention, a utilisé sa taxonomie de concepts pour en déduire qu'un "manuel de cours" est un "livre", que les "équations différentielles" font partie de "l'analyse" et que par conséquent sa réponse est pertinente. Le fait que la taxonomie soit partagée apparaît implicite puisque le professeur suppose que sa réponse sera comprise et qu'elle l'est effectivement.

Une ontologie est donc particulièrement intéressante quand elle décrit les connaissances consensuelles partagées par une communauté. Quand des personnes s'accordent sur l'utilisation et la théorie spécifiée par l'ontologie, elles prennent "un engagement ontologique". Il est donc important d'obtenir un consensus sur cette conceptualisation qui soit partagé par toute la communauté des intéressés.

4 Ontologies et ressources terminologiques dédiées au e-Learning

4.1 Les ontologies dans le standard LOM

Le standard LOMv1.0 (LOM) est constitué d'un schéma de métadonnées pour décrire des ressources pédagogiques. En plus des informations "classiques" du type auteur, date de création etc., le standard propose un embryon de structure pour utiliser des schémas de type ontologique dans l'indexation des ressources. Une catégorie "classification" existe, offrant les éléments nécessaires pour référencer une classification hiérarchique. L'utilisation recommandée pour ces éléments mérite d'être détaillée.

L'élément Taxon recueille un terme d'une taxonomie correspondant à la ressource. La taxonomie en question est désignée par l'élément "Source". Le chemin complet où se trouve le terme dans la taxonomie peut être spécifié dans l'élément "Taxon Path". Si ce terme possède un identificateur particulier dans cette taxonomie c'est l'élément "ID" qui le recueille. Enfin il est possible d'indiquer le but ("Purpose") de l'association entre la ressource et le terme de la classification. Par exemple un

fichier contenant l'énoncé du théorème de Pythagore sera annoté par l'élément "Géométrie", choisi dans une taxonomie des domaines mathématiques. Le but de cette annotation sera indiqué dans l'élément "Purpose" par le terme standard "discipline" puisque l'annotation indique dans ce cas à quelle discipline ce document est relatif.

Cette proposition du standard se porte facilement dans le formalisme RDF. On peut en effet voire l'indexation précédente comme le fait de relier une ressource à un terme d'une classification par une relation à la sémantique prédéfinie. La représentation (binding) de LOM en RDFS proposée par (Nilsson et al, 2003) exploite cette technique : Le schéma RDFS comporte une relation "taxon", définie comme une relation générique exprimant le lien de classification existant entre les éléments de la hiérarchie (taxon peut endosser divers sens : sous-partie, sous-sujet, etc.). La racine de la hiérarchie est l'instance d'un concept particulier "Taxonomy", chaque élément de la taxonomie se présente comme une instance d'une classe prédéfinie (ex dcterms :MeSH) et les éléments sont hiérarchisés par la relation taxon. La taxonomie s'exprime ainsi sous la forme de triplés RDF (cf. figure 15).

```
<lom-cls :Taxonomy rdf :ID="MeSH">
  <dc :title>Medical Subject Headings</dc :title>
  <lom-cls :taxon>
    <dcterms :MeSH rdf :ID="A">
      <rdf :value>A</rdf :value>
      <dc :title>Anatomy</dc :title>
      <lom-cls :taxon>
        <dcterms :MeSH rdf :ID="A01">
          <rdf :value>A01</rdf :value>
          <dc :title>Anatomy</dc :title>
          ...
    </dcterms :MeSH>
  </lom-cls :taxon>
</lom-cls :Taxonomy>
```

Fig. 15 – Exemple de définition d'une taxonomie en RDF

Comme son nom l'indique la classification ainsi représentée est une taxonomie, avec une seule relation de hiérarchie entre les éléments. On notera que le "multi-héritage" n'est pas autorisé par (Nilsson et al, 2003), alors qu'il est possible de l'exprimer en RDF.

La relation taxon s'interprète pour certains uniquement comme un relation entre sujets d'un domaine. Une discussion est en cours au sein du W3C pour déterminer les bonnes pratiques à observer pour modéliser les relations délicates qui existent entre instances d'un concept et sujets d'un domaine.

Il est possible d'utiliser plusieurs taxonomies et de les relier aux ressources par plusieurs liens. L'association ("binding") de LOM et RDF prévoit d'ailleurs différentes relations entre l'objet d'apprentissage ("learning object") et les instances des taxonomies (cf. figure 16) en s'inspirant de l'élément "Purpose" de LOM.

```
dc :subject,  
lom-cls :prerequisite,  
lom-cls :educationalObjective,  
lom-cls :accessibilityRestrictions  
lom-cls :educationalLevel  
lom-cls :skillLevel  
lom-cls :securityLevel  
lom-cls :competency
```

Fig. 16 – Relations entre objet d'apprentissage et instances de la taxonomie

Le standard LOM parle de systèmes de classification ("classification system") et non d'ontologie. Il est clair que la contrainte sur la structure et le manque de spécificité du lien "taxon" réduisent l'intérêt de cette représentation. Le travail sur l'association LOM/RDF est toujours en cours, et le résultat du travail du working group W3C sera intéressant à intégrer.

Parmi les taxonomies utilisées dans des projets de e-learning et susceptibles d'être mises en œuvre avec LOM on peut citer (Brase & Nedjl, 2002) :

- ACM CSS¹, Association for Computing Machinery Computer and Communications Security, taxonomie pour classer les articles parus dans le domaine de l'informatique.
- SWEBOK² Software Engineering Body of Knowledge, pour décrire le domaine de l'ingénierie logicielle.
- Bloom's Taxonomy³ publiée en 1956, cette classification de termes permet de décrire des objectifs pédagogiques, notamment dans le domaine cognitif.

Le Dublin Core⁴, qui propose un schéma pour annoter tout type de document, définit des classes particulières pour 5 classifications couramment utilisées dans les bibliothèques :

- LCSH Library of Congress Subject Headings⁵ : taxonomie de sujets proposée par la bibliothèque du congrès américain.
- MESH Medical Subject Headings⁶ : l'arborescence des termes de médecine.
- DDC Dewey Decimal Classification⁷ : présenté comme le système de classification le plus répandu dans les bibliothèques.

¹ <http://www.acm.org>

² <http://www.swebok.org>

³ <http://faculty.washington.edu/krumme/guides/bloom.html>

⁴ <http://dublincore.org>

⁵ <http://lcweb.loc.gov/cds/lcsh.html>

⁶ <http://www.nlm.nih.gov/mesh/meshhome.html>

- LCC Library of Congress Classification⁸ : la classification proposée par la bibliothèque du congrès américain, utilise des chiffres et des lettres.
- UDC Universal Decimal Classification⁹ : traduction multilingue de DDC.

L'argument majeur en faveur de cette représentation est son statut de standard qui permet d'envisager des échanges de métadonnées plus intéressantes pour la pédagogie que celles concernant l'auteur et la date de création. Plusieurs liens possibles (cf. fig. 16) entre l'objet annoté et les termes de l'ontologie font déjà partie du standard, on peut donc envisager une interopérabilité jusqu'à ce niveau. Ensuite l'interopérabilité peut être mise en œuvre en reposant soit sur des ontologies standard partagées (ce qui n'est pas impensable pour certains domaines, voire les exemples cités plus haut), soit sur des traductions ou mises en correspondance ("mappings") entre les ontologies propriétaires. Un exemple de «mapping» est celui proposé par (Simon et al, 2004) et le projet ELENA.

4.2 Les ontologies utilisées pour le e-learning

L'utilisation d'ontologies pour développer des systèmes éducatifs n'est pas récente (Mizoguchi et al, 1997). Aujourd'hui le recours à une modélisation, partagée par au moins l'ensemble des acteurs d'un système, est quasi incontournable. Le partage entre plusieurs applications reste problématique.

Dans la majorité des cas l'ontologie sert à expliciter les buts pédagogiques poursuivis. C'est le cas dans l'exemple donné au dessus pour illustrer l'utilisation de LOM. D'après (Woukeu et al, 2003) la recherche se focalise désormais sur l'utilisation d'ontologies dans les stratégies pédagogiques. Il existe donc des tentatives pour définir des ontologies du e-learning, mais un des problèmes est de savoir ce que le ou les ontologies doivent décrire.

(Stojanovic et al, 2001) propose une ontologie de "cours" qui se décompose en 3 sous parties décrivant le contenu, le contexte et la structure. Le contenu est communément appelé l'ontologie du domaine. Les deux autres parties sont liées à l'aspect pédagogique (structuration en chapitres, nature des parties, etc.). Il existe de nombreuses variations sur ce thème mais globalement deux directions de modélisation sont à considérer : le domaine étudié et la pédagogie. Éventuellement ces domaines peuvent être réunis au sein d'une ontologie unique (Abel et al, 2004).

Concernant la pédagogie, une ontologie intéressante est celle proposée par (Barros et al, 2002) qui met l'accent sur l'aspect coopératif de l'apprentissage. Elle permet, d'après les auteurs, de décrire des systèmes d'enseignement collaboratifs. Plusieurs exemples d'application sont cités, notamment le projet DEGREE (Distance Environment for Group Experiences) qui s'intéresse à l'analyse des processus d'interaction entre étudiants pour améliorer la collaboration.

⁷ <http://tnrdlib.bc.ca/dewey.html>

⁸ <http://www.loc.gov/catdir/cpsolcco/lcco.html>

⁹ <http://www.udcc.org>

Les ontologies de domaine sont quand à elles plus souvent utilisées pour décrire le contenu des ressources (souvent désignées par le terme "Learning Objects" - LOs). Mais cela n'est pas sans conséquence sur la pédagogie pour autant. Pour (Breuker et al, 1999) les ontologies de domaine ne sont ni plus ni moins que l'expression des buts pédagogiques. Car ce qu'exprime l'ontologie du domaine c'est toute la structure qu'on veut justement faire comprendre et transmettre à l'apprenant. Breuker indique donc que l'introduction d'ontologies dans l'apprentissage pourrait changer la vision classique de l'enseignement. L'acquisition des compétences se trouve remplacée par la compréhension du domaine au travers de l'ontologie. L'exemple cité dans ces travaux est celui de l'utilisation de l'ontologie FOlaw pour l'apprentissage du droit. Dans le même esprit (Stutt & Motta, 2004) propose une vision d'un système de e-learning où l'apprenant peut visualiser directement des idées sous forme de graphiques ("Knowledge Charts"). Les graphiques permettent des visualisations de connaissances qui peuvent aussi être vues comme des instanciations d'ontologies.

Dans le projet Ontoport (Woukeu et al, 2003), l'ontologie sert de support à la navigation et vient compléter les liens hypertextes par des liens sémantiques. Dans ce projet l'ontologie dite de "domaine" modélise le matériel pédagogique disponible (cours, exercices, etc.) et les possibilités de navigation offertes (par auteur, par type de ressource, etc.).

Dans le projet IMAT (Desmoulins & Grandbastien, 2000), tourné vers l'utilisation d'ontologies pour l'indexation, la manipulation de document nécessite l'emploi d'une ontologie dite "document", qui vient s'ajouter aux ontologies classique du domaine et de la pédagogie.

De nombreux projets éducatifs font donc appel aux ontologies, on arrive aujourd'hui à percevoir le rôle que peut jouer l'ontologie dans la conceptualisation et la conception (Barros et al, 2002) des systèmes. En revanche il semble exister encore peu de résultat sur l'impact à grande échelle de leur utilisation.

4.3 Les langages de modélisation

L'ontologie fournit un ensemble de concepts et de relations modélisant un domaine, une stratégie pédagogique, etc. Des travaux existent pour spécifier des modélisations plus "pratiques" et directives sous forme de langage de modélisation (Educational Modeling langage).

La contribution la plus importante à ce niveau est le langage EML-OU (EML - Educational Modeling Langage) (Koper, 2001). Cette proposition de Open University of the Netherlands, est intégrée depuis 2003 dans la spécification IMS Learning Design Information Model¹⁰. On ne parle pas d'ontologie dans la spécification, mais d'un schéma UML séparant en 3 niveaux l'activité d'enseignement :

¹⁰ IMS Global Learning Consortium, <http://www.imsglobal.org/>

-A- définition du vocabulaire nécessaire pour supporter la diversité pédagogique (ex : Learning Objective, Role, ...), chaque concept est accompagné de règles d'utilisation.

-B- l'ajout des concepts de Propriétés et de Conditions au niveau A, cela permet notamment une plus grande adaptabilité à l'utilisateur. (ex : utilisation de propriété pour définir des préférences personnelles)

-C- l'ajout du concept de Notification au niveau B, ce qui rend le système plus réactif aux actions de l'utilisateur (si un résultat est atteint, par exemple, cela peut déclencher la mise à disposition d'une nouvelle activité). Cela peut induire des dynamiques plus complexes car ce n'est pas nécessairement celui qui déclenche la notification qui la reçoit (le professeur peut être notifié d'un progrès de l'étudiant, par exemple)

Le schéma UML représente une abstraction de haut niveau pour décrire des méthodes d'apprentissage. Plusieurs méthodes pédagogiques peuvent être envisagées de ce point de vue : situations coopératives, basées sur un problème, etc. (Nodenot et al, 2003)

La spécification IMS Learning Design permet également de décrire la structure et les processus à l'intérieur d'une "unité d'apprentissage". Cette unité est composée de LOs avec des objectifs, des pré-requis, des activités d'apprentissage et d'enseignement associés. Le tout est présenté sous la forme d'un "workflow" : "à une personne se trouve attribué un rôle, dans ce rôle la personne travaille pour obtenir un résultat en effectuant des activités pédagogiques et/ou de support, dans un certain environnement." (IMSLD)

L'un des intérêts du schéma UML par rapport à d'autres formalismes est qu'il offre une connexion directe vers le monde de l'ingénierie logicielle. En outre il existe déjà des outils pour manipuler les modèles. De ces diagrammes découlent un vocabulaire et un schéma de métadonnées pour indexer les ressources en XML par rapport à ce modèle.

Dans le même esprit le langage de modélisation Xedu (Buendia & Diaz, 2003) propose de représenter les connaissances sous la forme d'une "structure didactique", celle ci permet d'exprimer à la fois des connaissances du domaine ainsi que de aspects pédagogiques. A cela s'ajoute les moyens de spécifier un profil d'utilisateur et des scénarios d'apprentissage.

Un autre langage proposé il y a déjà 4 ans dans le même esprit est le langage PALO (Rodriguez-Artcaho et al, 1999). Il permet d'exprimer le modèle de navigation, le contenu (en faisant appel à un modèle de référence), le degré d'interactivité et le suivi de l'utilisateur. Le modèle de référence s'exprime à l'aide de modèles (templates) prédéfinis qui séparent nettement le contenu pédagogique des moyens (questions, exercices, cours, etc.). On retrouve en quelque sorte la séparation entre ontologie du domaine et de la pédagogie présentée plus haut. Ce langage utilise SGML mais ne fait pas appel à une conceptualisation explicite sous forme d'ontologie ou de schéma UML ce qui rend ce travail difficilement réexploitable et comparable à EML-OU.

D'autres EMLs existent mais c'est EML-OU qui a été retenu pour servir de base à la spécification IMS-LD.

Finalement en termes de modélisation, les deux axes cités précédemment constituent bien le point de départ de toute ontologie du e-learning :

- La modélisation du contenu, ceci relève du domaine étudié (mathématiques, physique, biologie, etc.) mais la modélisation n'est pas neutre et a des répercussions sur l'apprentissage.
- La pédagogie, qui rassemble tous les aspects organisationnels, relationnels et même sociaux autour de l'acte d'apprentissage.

Il apparaît ici que les ontologies manquent peut-être de moyens pour guider l'aspect dynamique (workflow), qui s'exprime mieux au travers de langages de type EML. Mais il ne faut pas tracer de frontière stricte entre ces deux outils de modélisation, dans le projet ELENA (Simon et al, 2004) les concepts de IMSLD ont servi à la création d'une ontologie pour les besoins spécifiques du projet. La formalisation sous forme d'ontologie ou de langage dépend donc de l'usage. De toute façon il existe clairement une base de concepts communs.

Une réflexion similaire existe dans les travaux visant à uniformiser la modélisation d'un individu à l'aide d'ontologies comme FOAF. On note qu'il existe un large recoupement entre les différents standards mais aucun consensus ne s'est pour l'instant établi, c'est également le cas pour le e-learning.

Enfin, notons que divers modèles peuvent être intégrés dans des ontologies dédiées au e-Learning : apprenants, enseignants, ressources pédagogiques, stratégies pédagogiques, formation, modèles cognitifs, modèles d'interaction, etc.

5 Conclusion

Ce chapitre a présenté une synthèse assez détaillée des travaux liés aux ontologies, travaux de plus en plus florissants avec le Web sémantique, et qui à plus long terme devraient certainement bénéficier aux travaux sur les ontologies spécialement dédiées au E-learning. Les ontologies constituent l'une des bases les plus importantes de l'approche Web sémantique pour le e-Learning.

Références

- ABEL M-H. ET AL. (2004), Using an Organizational Memory for e-learning, Workshop of Knowledge Management and Organizational Memories, ECAI'2004, Valencia, Spain, 2004-08-22.
- ASSADI H. (1998), Construction d'ontologies à partir de textes techniques, Thèse de doctorat, 1998, Université Paris 6.
- AUSSENAC-GILLES N., BIEBOW B., SZULMAN S. (2000). Revisiting Ontology Design : a Method Based on Corpus Analysis, In Proc. EKAW'2000, Juan-les-Pins, p172-188.
- BACH T.-L., DIENG-KUNTZ R., GANDON F. (2004). On Ontology Matching Problems (For Building A Corporate Semantic Web In A Multi-Communities Organization). ICEIS 2004. Porto, 2004.

- BACHIMONT B. (2000): "Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances" ; In "Ingénierie des connaissances Evolutions récentes et nouveaux défis", J. Charlet, M. Zacklad, G. Kassel, D. Bourigault, Eyrolles 2000, ISBN 2-212-09110-9
- BARROS B., VERDEJO F., READ T. & MIZOGUCHI R. (2002), Applications of a Collaborative Learning Ontology, Second Mexican International Conference on Artificial Intelligence : Advances in Artificial Intelligence , 2002, pp 301 - 310.
- BENAYACHE A., BARRY C., CHAPUT B. & ABEL M.-H. (2004). Une ontologie d'application pour indexer les ressources d'une application e-learning. CIDE'04.
- BERNARAS A., LARESGOITI I. & CORERA J. (1996) Building and Reusing Ontologies for Electrical Network Applications, Proceedings of the European Conference on Artificial Intelligence (ECAI'96). ECAI 96. Publisher : John Wiley & Sons, pp298-302, 1996.
- BIÉBOW B. & SZULMAN S. (1999), TERMINAE : A linguistic-based tool for the building of a domain ontology, Proc. of the 11th European Workshop, Knowledge Acquisition, Modeling and Management (EKAW'99), Dagstuhl Castle, Germany, 26-29 Mai, 1999, p. 49-66
- BIEZUNSKI M. (2001), TOPIC MAPS AT A GLANCE, [HTTP://WWW.INFOLOOM.COM/TMARTIC.HTM](http://www.infoloom.com/tmartic.htm)
- BOURIGAUT D. (1994). LEXTER, un Logiciel d'Extraction de TERminologie. Application a l'acquisition des connaissances a partir de textes. Ph.D. Thesis, Ecole des Hautes Etudes en Sciences Sociales, Paris, 1994
- BOURIGAUT D. & LEPINE P. (1995). Utilisation d'un logiciel d'extraction de terminologie (Lexter) pour l'acquisition des connaissances à partir de textes. In N. Aussenac-Gilles, P. Laublet, C. Reynaud, editors, Acquisition et ingénierie des connaissances : tendances actuelles. Toulouse, Cépaduès.
- BRASE J. & NEIDL W. (2003), Ontologies for e-Learning, 2003.
- BREUKER J., MUNTJEWERFF A. & BREDEWEG B. (1999), Ontological Modelling for Designing Educational Systems, AIED'99 Workshop on Ontologies for Intelligent Educational Systems, 1999.
- BUENDIA F. & DIAZ P. (2002), A Framework for Educational Adaptive Hypermedia Applications, Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Malaga, Spain, 2002.
- CAROLL J. M. (1997), Scenario-Based Design", Chapter 17 "Handbook of Human-Computer Interaction" Second, completely revised edition, M. Helander, T.K. Landauer, P. Prabhu (eds), 1997, Elsevier Science B.V.
- COHEN W. W., RAVIKUMAR P. & FIENBERG S. (2003). A Comparison of String Distance Metrics for Name-Matching Tasks. IJCAI 2003, Workshop on Information Integration on the Web.
- CYCL. <http://www.cyc.com/cyc-2-1/ref/cycl-syntax.html>
- DESMOULINS C. & GRANDBASTIEN M. (2000), Des ontologies pour indexer des documents techniques pour la formation professionnelle, IC'2000, Toulouse, 10-12 mai 2000.
- KOPER R. (2001), Modeling units of study from a pedagogical perspective : the pedagogical meta-model behind EML
- DELTEIL A. (2002), Représentation et apprentissage de concepts et d'ontologies pour le Web, Thèse de doctorat en informatique, Université de Nice Sophia Antipolis (UNSA), 26 novembre 2002
- DIENG R., HUG S. (1998), Comparison of "personal ontologies" represented through conceptual graphs. Proc. of ECAI'98, Wiley & Sons, p. 341-345, Brighton, UK, 1998.
- DIENG-KUNTZ R., CORBY O., GANDON F., GIBOIN A., GOLEBIEWSKA J., MATTA N., RIBIERE M. (2001), Méthodes et Outils pour la Gestion des Connaissances. Une approche pluridisciplinaire du Knowledge Management (2nd Edition), Dunod Edition - INFORMATIQUES Série Systèmes d'information - ISBN 2 10 006300 6, 2001

- DOAN A., DOMINGS P. & LEVY A. Y. (2000). Learning source descriptions for data integration. In Proc. of WebDB.
- DOAN A., DOMINGS P. & HALEVY A. (2001). Reconciling Schemas of Disparate Data Sources : A Machine-Learning Approach. SIGMOD Conference.
- DOAN A., MADHAVAN J., DOMINGOS P. & HALEVY A. (2002). Learning to Map Between Ontologies on the Semantic Web. The 11th international world wide web conference. Honolulu, Hawaii, USA.
- DUCOURNEAU R., EUZENAT J., MASINI, NAPOLI A. (1998) Langages et modèles à Objets Etat des recherches et perspectives. INRIA, Collection Didactique, 1998, ISSN 0299 - 0733 ; ISBN 2 - 7261 - 1131
- ENDRES-NIGGEMEYER B., HERTENSTEIN B., VILLIGER C., ZIEGERT C. (1997): Constructing an Ontology for WWW Summarization in Bone Marrow Transplantation (BMT) 12th Annual Classification Research Workshop of the ASIS Conference in Washington 2001
- EUZENAT J. (1996), Corporate Memory through Cooperative Creation of Knowledge Based Systems and Hyper-Documents, Proceedings of KAW'96, p36-20, Banff, Canada.
- EUZENAT, J. Ed: State Of The Art On Current Alignment Techniques, December 2004, IST Knowledge Web Network Of Excellence No FP6-507482, Deliverable #KWEB/2004/D2.2.3.
- FARQUHAR A., FIKES R. & RICE J. (1996), The Ontolingua Server : A Tool for Collaborative Ontology Construction. Knowledge Systems Laboratory, September, 1996.
- FERNANDEZ M., GOMEZ-PEREZ A. & JURISTO N. (1997). METHONTOLOGY : From Ontological Arts Towards Ontological Engineering. In Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering, Stanford, USA, pages 33-40, March 1997.
- GANDON F. (2002) Distributed Artificial Intelligence and Knowledge Management : ontologies and multi-agent systems for a corporate semantic web, - Scientific Philosopher Doctorate Thesis In Informatics, Defended Thursday the 7th of November 2002 INRIA and University of Nice - Sophia Antipolis
- GENESERETH M. & FIKES R. (1992) Knowledge Interchange Format. Technical Report Computer Science Department. Stanford University, Logic 92-1.
- GIUNCHIGLIA F. & SHVAIKO P. (2003). Semantic Matching. CEUR-WS, vol. 71.
- GOLEBIEWSKA J. (2002), Exploitation des ontologies pour la mémoire d'un projet-véhicule, Méthode et outil Samovar (Système d'Analyse et de Modélisation des Validations des Automobiles Renault), thèse de doctorat en informatique, Université de Nice Sophia Antipolis (UNSA), 4 février 2002
- GOLEBIEWSKA J., DIENG R., CORBY O., MOUSSEAU D. (2001). Building and Exploiting Ontologies for an Automobile Project Memory, K-CAP, Victoria, Oct. 2001, p. 52-59.
- GÓMEZ-PÉREZ A. (1998). Ontological Engineering. Tutorial ECAI'98, Budapest, Hongrie, août 1998
- GUARINO N. (1992). Concepts, Attributes, and Arbitrary Relations : Some Linguistic and Ontological Criteria for Structuring Knowledge Bases. Data and Knowledge Engineering 8 : 249-261, 1992.
- GUARINO N. (1997), Understanding, Building and Using Ontologies. A Commentary to "Using Explicit Ontologies in KBS Development", by van Heijst, Schreiber, and Wielinga. International Journal of Human and Computer Studies vol. 46 n. 2/3, pp. 293-310, 1997
- GUARINO N., GIARETTA P. (1995). "Ontologies and knowledge bases, towards a terminological clarification", (in Mars N., eds., Towards very large knowledge bases : knowledge building and knowledge sharing), ISO Press, 1995, pp. 25-32.
- GUARINO N. & WELTY C. (2000). Towards a methodology for ontology-based model engineering. In Proceedings of ECOOP-2000 Workshop on Model Engineering. Cannes,

France. Available from <http://www.ladseb.pd.cnr.it/infor/ontology/Papers/OntologyPapers.html>

HAASE P. & SURE Y. (2004). D3.1.1.b State-of-the-Art on Ontology Evolution. SEKT/2004/D.3.1.1.b/v0.5. Institute AIFB, University of Karlsruhe.

HAASE P., SURE Y. & VRANDECIC D. (2004). D3.1.1 Ontology Management and Evolution – Survey, Methods and Prototypes. EU-IST Integrated Project (IP) IST-2003-506826 SEKT, Deliverable D3.1.1 (WP1.1) Document Id. SEKT/2004/D3.1.1/v1.0, December 20, 2004

IDEF5 Method Report, Knowledge Based Systems, Inc.,

IEEE 1074-1995 - IEEE Standard for Developing Software Lifecycle Processes

(IMSLD) IMS GLOBAL LEARNING CONSORTIUM, [HTTP://WWW.MSGLOBAL.ORG/](http://www.msGLOBAL.ORG/)

KASSEL G., ABEL M.-H., BARRY C., BOULITREAU P., IRASTORZA C., PERPETTE S. (2000) Construction et exploitation d'une ontologie pour la gestion des connaissances d'une équipe de recherche, Actes d'IC'2000, Toulouse, 10-12 Mai 2000, <http://www.irit.fr/IC2000/actes-enligne.html>

KASSEL G. (2002). OntoSpec : une méthode de spécification semi-informelle d'ontologies, Actes de IC'2002, 2002, Rouen, pp. 75-87.

KAYSER D. (1997) La représentation des connaissances, , ISBN 2866016475, 1997

KIRYAKOV A. & OGNANOV D. (2002). Tracking changes in RDF(S) repositories, in Proceedings of 13 International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002), Sigüenza, Spain, LNCS 2473, pp. 373-378, 2002.

KLEIN M. & FENSEL D. (2001). Ontology versioning for the Semantic Web. In Proceedings of the First International Semantic Web Working Symposium (SWWS), pages 75-91, Stanford University, California, USA.

KLEIN M. & NOY N.F. (2003). A component-based framework for ontology evolution. In Proceedings of the Workshop on Ontologies and Distributed Systems, IJCAI '03, Acapulco, Mexico. Also available as Technical Report IR-504, Vrije Universiteit Amsterdam.

KLEIN M., FENSEL D., KIRYAKOV A. & OGNANOV D. (2002b). Ontology versioning and change detection on the web. In 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02), number 2473 in LNCS, page 197 ff, Sigüenza, Spain.

KLEIN M., KIRYAKOV A., OGNANOV D. & FENSEL D. (2002d). Finding and characterizing changes in ontologies. In Proceedings of the 21st International Conference on Conceptual Modeling (ER2002), Springer-Verlag, LNCS 2503, pages 79-89, Tampere, Finland.

KLEIN M. (2002). Supporting evolving ontologies on the internet. In Lindner, W. and Stuller, J., editors, Proceedings of the EDBT 2002 PhD Workshop, number 2490 in LNCS, pages 597-606, Prague, Czech Republic.

KLEIN M. (2002a), Versioning of distributed ontologies, available as Deliverable D20 V1.1, EU/IST Project WonderWeb, <http://wonderweb.semanticweb.org/deliverables/documents/D20-1.1.pdf>, 2002.

KLEIN M. (2004). Change Management for Distributed Ontologies. PhD thesis, Vrije Universiteit Amsterdam

KORFHAGE R. (1997), Information Storage and Retrieval, John Wiley & Sons (ed), 1997, ISBN : 0471143383

LA FRANCE M. (1992) "Questioning Knowledge Acquisition". In "Questions And Information Systems" Edited by : Lauer, Thomas W, Peacock Eileen, Graesser, Arthur C. ; Editions : L. Erlbaum associates ; 1992 ;ISBN : 0-8058-11018-8.

(LOM) IEEE Learning Object

LENAT D. & GUHA R. (1990), Building Large Knowledge Based Systems : Representation and Inference in the Cyc Project. Addison-Wesley Publishing, 1990.

[HTTP://LTSC.IEEE.ORG/WG12/FILES/LOM_1484_12_1_V1_FINAL_DRAFT.PDF](http://LTSC.IEEE.ORG/WG12/FILES/LOM_1484_12_1_V1_FINAL_DRAFT.PDF)

MADHAVAN J., BERNSTEIN P. A. & RAHM E. (2001). Generic Schema Matching with Cupid. In Proceedings of the 27th International Conferences on Very Large Databases.

MELNIK S., GARCIA-MOLINA H. & RAHM E. (2001). Similarity Flooding : A Versatile Graph Matching Algorithm (Extended Technical Report). <http://dbpubs.stanford.edu/pub/2001-25>.

MELNIK S., GARCIA-MOLINA H. & RAHM E. (2002). Similarity Flooding : A Versatile Graph Matching Algorithm and its Application to Schema Matching. In Proc. 18th ICDE Conference.

MILLER G. A. (1995). WordNet: A lexical database for English. Communications of the ACM, 38(11) :39--41, 1995.

MIZOGUCHI R. & IKEDA M. (1997). Towards Ontology Engineering. In Proceedings of The Joint 1997 Pacific Asian Conference on Expert systems / Singapore International Conference on Intelligent Systems, pp. 259-266, 1997

MIZOGUCHI R., IKEDA M., & SINITSKA K. (1997). Roles of Shared Ontology in AI-ED Research - Intelligence, Conceptualization, Standardization, and Reusability –Proc. of AIED-97, pp.537-544, also as Technical Report AI-TR-97-4, I.S.I.R., Osaka University, 1997

NILSSON M., PALMÈR M. & BRASE J. (2003), The LOM RDF binding - principles and implementation, 3rd Annual Ariadne Conference, 2003.

NODENOT T., LAFORCADE P., MARQUESUZA C. & SALABERRY C. (2003), Knowledge modeling of Co-operative learning situations : Towards a uml profile, Artificial Intelligence in Education 2003, Sydney, Australia, 2003.

NOY N. F., HAFNER C. D. (1997). The State of the Art in Ontology Design : A Survey and Comparative Review. AI Magazine 18(3) : 53-74 (1997)

NOY N.F. & KLEIN M. (2004) Ontology evolution: Not the same as schema evolution. Knowledge and Information Systems, 6(4):428-440.

NOY N. F. & KLEIN M. (2004a). Visualizing changes during ontology evolution. In Proceedings of the International Conference on Intelligent User Interfaces (IUI 2004), Madeira, Portugal, 2004.

NOY N.F., KUNNATUR S., KLEIN M. & MUSEN M. A. (2004). Tracking changes during ontology evolution. In 3rd International Semantic Web Conference (ISWC2004), Hiroshima, Japan.

NOY N. F. & MUSEN M. A. (2001). Anchor-PROMPT : Using Non-Local Context for Semantic Matching. In the Proceedings of the IJCAI-2001 Workshop on Ontologies and Information Sharing, Seattle, WA.

OLIVIER D. E., SHAHAR Y., SHORTLIFFE E.H., MUSEN M.A. (1998), Representation of Change on Controlled Medical Terminologies, In Proceedings of AMIA, October 1998

RAHM E. & BERNSTEIN P. (2001). On matching schemas automatically.

RIBIERE M. (1999) Représentation et gestion de multiples points de vue dans le formalisme des graphes conceptuels. PhD from Univ. Nice Sophia-Antipolis

RODRÍGUEZ-ARTACHO M., VERDEJO F., MAYORGA J.I. & CALERO Y. (1999), Using a high-level language to describe and create web based learning scenarios, Frontiers In Education Conference FIE '99, San Juan, Puerto Rico, 1999-11.

SERAFINI B. L., BOUQUET P., MAGNINI B. & ZANOBINI S. (2003). An algorithm for matching contextualized schemas via SAT. Technical Report DIT-03-003, Informatica e Telecomunicazioni, University of Trento.

SIMON B. ET AL. (2004), Conceptualising Smart Spaces for Learning, Journal of interactive Media in Education, 2004-05-21.

SOMMERVILLE I. (1995), Software Engineering, Fifth Edition, Addison-Wesley, 1995, ISBN 0-201-42765-6

- SOWA J.F. (1984), *Conceptual Structures : Information Processing in Mind and Machine*, Addison-Wesley, 1984.
- SOWA J. F. (2000b) *Guided Tour of Ontology* ; <http://www.jfsowa.com/ontology/guided.htm>
- SOWA J.F. (2002) *Conceptual Graphs Standard*, ISO/JTC1/SC 32/WG2 N 000 <http://users.bestweb.net/~sowa/cg/cgstand.htm>
- STAAB S., MAEDCHE A. (2000). *Axioms are Objects, too -Ontology Engineering beyond the Modeling of Concepts and Relations*. Internal Report 399, Institute AIFB, Karlsruhe University, 2000
- STOJANOVIC L., MAEDCHE A., STOJANOVIC N. & MOTIK B. (2002). *User-driven ontology evolution management*. In *European Conf. Knowledge Eng. and Management (EKAW 2002)*, pages 285–300. Springer-Verlag, 2002.
- STOJANOVIC L., STOJANOVIC N. & HANDSCHUH S. (2002a). *Evolution of the Metadata in the Ontology-based Knowledge Management Systems*, *Proceedings of the 1st German Workshop on Experience Management: Sharing Experiences about the Sharing of Experience*, 2002.
- STOJANOVIC L., MAEDCHE A., STOJANOVIC N. & STUDER R. (2003). *Ontology evolution as reconfiguration-design problem solving*. In *CAP 2003*, pages 162–171. ACM, OCT 2003.
- STOJANOVIC L. (2004). *Methods and Tools for Ontology Evolution*. PhD thesis, University of Karlsruhe, 2004.
- STOJANOVIC L., STAAB S. & STUDER R. (2001), *Elearning based on the semantic web*, *WebNet2001 - World Conference on the WWW and Internet*, Orlando, Florida, USA, 2001-10.
- STUCKENSCHMIDT H. & KLEIN M. (2003). *Integrity and change in modular ontologies*. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico.
- STUTT A. & MOTTA E. (2004), *Semantic Learning Webs*, *Journal of interactive Media in Education*, 2004-05-21.
- USCHOLD M. & GRUNINGER M. (1996). *Ontologies : Principles, methods and applications*. *Knowledge Engineering Review*, Vol. 11 :2, 93-136, 1996. Also available as AIAI-TR-191 from AIAI, The University of Edinburgh.
- WOUKEU A., WILLS G., CONOLE G., CARR L., KAMPA S. & HALL W. (2003), *Ontological Hypermedia in Education: A framework for building web-based educational portals*, *ED-MEDIA 2003-World Conference on Educational Hypermedia and Telecommunications*, Honolulu, Hawaii, USA, 2003, vol 1, pp 349-357.

Langages pour le Web Sémantique et le e-Learning

Marie-Hélène Abel¹, Rose Dieng-Kuntz², Danièle Hérin³, **Dominique Lenne**¹,
Claude Moulin¹, Pierre Pompidor³, Anfana Traoré³

¹Laboratoire Heudiasyc - UMR CNRS 6599
Université de Technologie de Compiègne
BP 20529, 60205 Compiègne Cedex - FRANCE
{Marie-Helene.Abel, Dominique.Lenne, Claude.Moulin}@utc.fr

²INRIA, Projet ACACIA,
2004 route des Lucioles, BP 93, 06902 Sophia Antipolis cedex
{Rose.Dieng}@sophia.inria.fr

³L.I.R.M.M. UMR n 9928 Université Montpellier II / CNRS
Equipe Extraction de connaissances sur le Web,
161, rue Ada, 34 392 Montpellier Cedex 5
{dh, pompidor, traore}@lirmm.fr

L'objectif de ce chapitre est de présenter les langages du web sémantique et d'étudier leur adéquation aux besoins du e-learning. En effet, le e-learning propose des modèles conceptuels, mais pas à proprement de formalismes de représentation. De nombreux systèmes sont développés, mais la réutilisation d'éléments, ne serait-ce que de simple ressources, n'est possible que si ces systèmes sont compatibles avec des standards et des formalismes de représentation. Le web sémantique est dans cette perspective un atout majeur.

Au-delà de la réutilisation d'éléments conceptuels pour la construction de systèmes indépendants et autonomes, commencent à émerger une génération de systèmes interopérables, c'est-à-dire pouvant communiquer entre eux, bien qu'ayant été conçus complètement séparément. Ces systèmes offrent aux autres des fonctionnalités intéressantes et utiles et la communication n'est possible que parce que des standards sont respectés.

Dans ce chapitre, nous prenons en compte plusieurs éléments caractéristiques des modèles conceptuels, utilisés dans les applications de e-learning, comme la description de ressources à l'aide de métadonnées ou la représentation de connaissance à l'aide de graphes conceptuels ou d'ontologie et nous montrons comment certains formalismes issus principalement des initiatives du web sémantique ou qui peuvent être candidats à devenir des recommandations peuvent aider à les représenter.

Nous essayons de rattacher ces langages à leur destination principale même si dans certains cas, ils peuvent être utilisés à plusieurs fins, soit parce qu'ils sont utilisés partiellement, soit parce qu'ils sont facilement extensibles.

Nous essayons de montrer également les limites syntaxiques ou théoriques de certains langages qui imposent encore de trouver des compromis lors de la réalisation d'applications. A l'opposé, d'autres langages restent encore ou risquent de rester inutilisables par manque d'outils permettant de les exploiter.

Nous évoquons également plusieurs systèmes au cours de notre exposé afin d'illustrer les éléments conceptuels qui les sous-tendent ainsi que les formalismes qu'ils peuvent emprunter. Nous terminons par un exemple intégrateur dont l'architecture fédère plusieurs démarches issues de domaines a priori distincts.

1 Langages de description de ressources

La description des ressources nécessite des métadonnées et des formalismes pour représenter ces métadonnées.

1.1 Métadonnées

Les métadonnées sont étudiées par ailleurs dans ce document. Nous nous intéressons ici à l'aspect « langage » en considérant uniquement les descripteurs utilisés dans deux modèles principaux Dublin Core Education et LOM.

1.1.1 Dublin Core Education

Le modèle de métadonnées du Dublin Core (DCMI, 2005) est un ensemble d'éléments simples mais efficaces pour décrire une grande variété de ressources en réseau. Le modèle du Dublin Core comprend 15 éléments dont la sémantique a été établie par un consensus international de professionnels provenant de diverses disciplines telles que la bibliothéconomie, l'informatique, le balisage de textes, la communauté muséologique et d'autres domaines connexes.

Cet ensemble de 15 éléments de métadonnées a trait :

- au *Contenu*: Title, Description, Subject, Source, Coverage, Type, Relation
- à la *Propriété intellectuelle*: Creator, Contributor, Publisher, Rights
- à la *Version*: Date, Format, Identifier, Language

Le modèle du *Dublin Core* ayant été conçu comme un référentiel commun à diverses communautés intéressées par les métadonnées, sa terminologie peut apparaître un peu déroutante dans certains contextes. Le *Dublin Core* parle ainsi de *Créateur* (*Creator*) d'une ressource et non pas d'*Auteur*, plus habituel dans le domaine de l'écrit.

Une version plus évoluée du *Dublin Core* autorise l'usage de qualificatifs; par exemple, l'élément *Description* peut être raffiné à l'aide des qualificatifs *tableOfContents* et *abstract*.

Le modèle du Dublin Core a été proposé pour faciliter une recherche de ressources assez simple. Il ne prétend pas répondre aux besoins et à la complexité de tous les métiers. Dans la plupart des besoins professionnels, il doit être complété par d'autres schémas de métadonnées. L'objectif visé est que des groupes d'utilisateurs se servent du modèle Dublin Core comme d'un noyau et qu'ils mettent au point leurs propres extensions en fonction des besoins de leur discipline ou de leur établissement.

Divers groupes de travail sur la norme Dublin Core (éducation, architecture, administration, description des collections, etc.) travaillent à l'élaboration d'ensembles d'éléments propres à une discipline afin de compléter les éléments du noyau. Le groupe de travail sur l'éducation a fait approuver un ensemble d'éléments propre à l'éducation. Ainsi le Dublin Core Education est composé de métadonnées génériques et reprend des métadonnées de la partie « Educational » du LOM : type d'utilisateur, niveau et type d'interactivité, durée, type de médiateur, conformité à un standard éducatif, réflexion sur type de ressources, qualité, niveau, type de pédagogie.

1.1.2 Learning Object Metadata

La spécification IEEE LOM (IEEE-LTSC, 2002) définit un schéma de base constitué d'une hiérarchie d'éléments permettant de décrire un objet pédagogique.

Le standard LOM vise à définir un ensemble minimal de données permettant de gérer, localiser et évaluer un objet pédagogique. Il permet d'étendre localement ces éléments et de les rendre facultatifs. Au plus haut niveau de la hiérarchie sont représentées neuf catégories :

La catégorie *General* regroupe l'information générale décrivant un objet pédagogique dans son ensemble.

La catégorie *Lifecycle* regroupe les caractéristiques relatives à l'historique et à l'état courant de l'objet pédagogique ainsi que celles des modifications qu'il a subies.

La catégorie *Meta-Metadata* regroupe les informations sur les métadonnées elles-mêmes et non sur les objets pédagogiques décrits par ces métadonnées.

La catégorie *Technical* regroupe les exigences et caractéristiques techniques permettant d'exploiter l'objet pédagogique.

La catégorie *Educational* regroupe les caractéristiques pédagogiques de l'objet pédagogique.

La catégorie *Rights* regroupe les informations concernant les droits sur la propriété intellectuelle et les conditions d'utilisation de l'objet pédagogique.

La catégorie *Relation* regroupe les caractéristiques qui définissent les liens entre objets pédagogiques.

La catégorie *Annotation* regroupe les commentaires sur l'utilisation pédagogique de l'objet pédagogique ainsi que des informations sur la date de création et à qui ces commentaires sont destinés.

La catégorie *Classification* décrit l'objet pédagogique dans un système de classification particulier.

Pour chaque catégorie et sous catégorie, le standard LOM précise un certain nombre d'information dont : nom, explication, taille, exemple de valeur, type de donnée.

Bien que très utilisé ce standard nécessite cependant des améliorations. En effet, comment juger un niveau d'interactivité entre l'utilisateur et la ressource selon l'échelle proposée? Un autre exemple concerne le type de la ressource d'apprentissage pour lequel il est proposé comme valeur : figure, graphe, table mais aussi exercice, questionnaire etc. Une spécification approfondie est dans ces cas souhaitable afin de favoriser une meilleure réutilisation.

1.2 RDF-RDF/S : Formalisme pour la description de ressources

Il est possible de décrire une ressource par une séquence d'assertions représentant ses caractéristiques. Une forme habituelle est celle de triplets (sujet, relation, objet) dont le sujet est un identificateur de la ressource elle-même. C'est la forme sur laquelle se basent les formalismes RDF (RDF, 2004) (Resource Description Framework) et même OWL (OWL, 2004).

La représentation d'un ensemble de ressources est alors un graphe où les nœuds sont des URIs spécifiant des ressources ou bien des littéraux (uniquement pour des nœuds feuille). Les arêtes spécifient des relations (ou propriétés) entre ressources ou entre ressources et littéraux. Il est possible de réifier un triplet RDF en un seul nœud ce qui permet de définir des relations d'arité supérieure à deux.

RDF possède une représentation XML. Dans ce langage, un triplet RDF a la structure de base suivante :

```
<rdf:Description rdf:about='URI de la ressource'>
  <propriété> valeur de la propriété </propriété>
</rdf:Description>
```

RDF permet donc de représenter des métadonnées attachées à des ressources. En tant que standard, il favorise l'interopérabilité entre les applications qui échangent des ressources sur le Web en en facilitant le traitement automatique. Citons à titre d'exemple un extrait de la représentation RDF du standard LOM (Nilsson, 2003)

```
<!DOCTYPE rdf:RDF [
  <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
  ... ..
  <!ENTITY lom 'http://ltsc.ieee.org/2002/09/lom-base#'>
  <!ENTITY lom-edu
    'http://ltsc.ieee.org/2002/09/lom-educational#'>
]>
<rdf:RDF
  xmlns:rdf="&rdf;"
  ... ..
  xmlns:lom-edu="&lom-edu;"
  xml:base="&lom-edu;"
  ... ..
  <lom-edu:LearningResourceType rdf:ID="Exercise">
    <rdfs:label>Exercise</rdfs:label>
  </lom-edu:LearningResourceType>

  <lom-edu:LearningResourceType rdf:ID="Simulation">
```

```
<rdfs:label>Simulation</rdfs:label>
</lom-edu:LearningResourceType>

<lom-edu:LearningResourceType rdf:ID="Questionnaire">
  <rdfs:label>Questionnaire</rdfs:label>
</lom-edu:LearningResourceType>

<lom-edu:LearningResourceType rdf:ID="Diagram">
  <rdfs:label>Diagram</rdfs:label>
</lom-edu:LearningResourceType>
... ..
</rdf:RDF>
```

Le but général de RDF est de définir un modèle permettant de décrire les ressources sans préjuger du type d'utilisation qu'on pourra en faire. La définition du modèle est également neutre par rapport au domaine d'appartenance des ressources. Pour faciliter la définition des métadonnées, il est possible de définir des classes de ressources. C'est dans cette perspective que les Schémas RDF (RDF/S, 2004) ont été créés.

Une collection de classes est appelée un *schéma RDF*. Les classes sont organisées en hiérarchie, et offrent une extensibilité grâce à la spécialisation en sous-classes. Au travers de l'acceptation et le partage de schémas, RDF/S facilite la réutilisation des définitions de métadonnées.

Les schémas RDF vont garantir que les documents RDF sont sémantiquement consistants. Une classe est instanciée en rdf par *rdf:type*.

```
<rdfs:Class ID='Nom de la classe' />
<rdf:description rdf:about='URI de la ressource'>
  <rdf:type rdf:resource='#Nom de la classe' />
</rdf:description>
```

Cependant les schémas RDF ont des limitations qui ne permettent généralement pas leur emploi pour la description d'ontologies. Ces limitations concernent par exemple la définition de classes à partir d'opérateurs (intersection, complément) ou à partir de contraintes (restriction sur la cardinalité de propriétés).

1.3 Topic Maps

Les topic maps (TM) ont été créées au début des années 1990 par le groupe de documentalistes Davenport pour répondre à une problématique d'échange de documents électroniques et plus particulièrement celui de leurs index. Elles sont devenues un standard au début des années 2000 (ISO/IEC, 2002). Les topic maps sont essentiellement basées sur les notions de topics, d'associations et d'occurrences.

Il est possible depuis 2001 d'exprimer une TM selon la norme XTM 1.0 (XTM, 2001), définie comme un langage XML particulier.

Les Topics

Un topic, ou sujet, est une entité à laquelle on attache un identificateur de façon à pouvoir la réutiliser ultérieurement. Prévus initialement pour représenter des ressources, les topics peuvent servir aussi à représenter les entités d'une ontologie. Il est ainsi possible d'indiquer qu'un topic est instance d'un autre, ce qui permet de faire un premier classement des topics en type et instance. La définition des types des topics dépend bien sûr de leur utilisation, des besoins de l'application et de la nature de l'information présente dans les documents (Pepper, 2000). Plus généralement, un topic est la représentation informatique d'un sujet plongé dans un contexte particulier. Dans un sens générique, un topic est un objet composé de l'information qui le caractérise.

Les Occurrences

Un topic peut être lié à une ou plusieurs ressources (article, image, vidéo, etc). Ces ressources sont généralement disponibles sous formes de fichiers électroniques et donc externes à la topic map. Dans le cas de ressources constituant de courts commentaires sur un sujet, il est cependant possible de les insérer dans la topic map elle-même. L'accès aux ressources est décrit dans les occurrences au moyen de références à des URL. Chaque occurrence possède un type ce qui permet de concevoir diverses catégories : nature des ressources qu'elles adressent (texte, image, son), ou rôle repéré dans l'ontologie (introduction, statistique, etc.) par exemple.

Les Associations

Les notions de topic et d'occurrence permettent d'organiser les ressources d'information et de créer une indexation « directe ». La notion d'association permet de créer des liens entre les différents topics et ainsi de permettre de naviguer entre eux.

Une association permet de lier deux ou plusieurs topics, appelés membres de l'association. Chaque membre joue un rôle dans l'association. Les associations peuvent également être classées selon leurs types qui sont eux-même définis en tant que topics. Cette définition de type d'association, permet de regrouper l'ensemble des topics qui ont un lien commun avec un autre topic (un auteur et un ensemble de titres de publications par exemple). Cela améliore considérablement l'efficacité de la navigation.

Les Contextes

Les topic maps définissent également la notion de contexte. Un contexte permet de relier les caractéristiques d'un topic (noms, occurrences et associations) à un contexte particulier permettant de lever certaines ambiguïtés. Une utilisation particulière du contexte est de préciser le format d'une donnée comme une date, une comparaison n'ayant de sens que dans des formats identiques (Altheim, 2002).

Exemple d'utilisation des TM (Abel et al., 2003)

L'extrait suivant montre l'indexation d'une page HTML utilisée pour introduire la notion de tableau. Pour cela, il est nécessaire de définir le topic « array » :

```
<topic id="tt-array">
  <baseName>
    <baseNameString>Array</baseNameString>
  </baseName>
  <baseName>
    <scope>
      <topicRef xlink:href="#t-french" />
    </scope>
    <baseNameString>Tableau</baseNameString>
  </baseName>
  ... ..
</topic>
```

Le topic “french” est défini préalablement. Cette définition indique que le nom de ce topic est

Tableau en français et sera *Array* dans un autre contexte. Considérant que le topic “web-page” est défini préalablement, l'indexation d'une page HTML par ce topic est réalisée par un lien d'occurrence :

```
<topic id="tt-array">
  ... ..
  <occurrence>
    <instanceOf>
      <topicRef xlink:href="#tt-web-page" />
    </instanceOf>
    <resourceRef xlink:href="/cours/chap09/cours.htm"/>
  </occurrence>
</topic>
```

La notion d'association (super-type, sous-type), permet de représenter un lien de spécialisation. L'extrait suivant (légèrement simplifié pour une meilleure compréhension) montre la déclaration du concept générique « *Array* », spécialisation du concept « *DataStructure* ». Le concept *DataStructure* est d'abord déclaré :

```
<topic id="tt-dataStructure">
  <baseName>
    <baseNameString>Data Structure</baseNameString>
  </baseName>
</topic>
```

Le lien de hiérarchie est créé par une association dont le premier membre joue le rôle de super-type (ici *dataStructure*) et le second membre joue le rôle de sous-type (ici *array*) :

```
<association>
  <instanceOf>
    <subjectIndicatorRef xlink:href="#supertype-subtype" />
  </instanceOf>
```

```

<member>
  <roleSpec>
    <subjectIndicatorRef xlink:href="#psi-supertype" />
  </roleSpec>
  <topicRef xlink:href="#tt-dataStructure" />
</member>
<member>
  <roleSpec>
    <subjectIndicatorRef xlink:href="#psi-subtype" />
  </roleSpec>
  <topicRef xlink:href="#tt-array" />
</member>
</association>

```

1.4 Composition de ressources – Schémas XML

Le vocable de ressource recouvre un large éventail d'entités, du simple document à un système tout entier. Une ressource peut ainsi être utilisée dans des types d'application très différents. Dans le cas des LMS ou LCMS, elles peuvent servir par exemple de support aux activités proposées aux étudiants mais également à décrire la nature composite de certains documents ou encore à décrire le flux des activités gérées par le système.

Dans le cas des exercices fermés, un même document XML regroupe en général l'énoncé et la solution. C'est dans un autre document que l'on trouvera les métadonnées attenantes à celui-ci. On emploie généralement pour cette utilisation de ressources une description à l'aide de schémas XML. Un schéma XML (XML Schema, 2004) décrit la structure d'une classe de documents compatibles.

La définition de ce standard a favorisé le développement d'outils génériques qui facilitent les traitements que peuvent subir les documents XML. On peut citer par exemple la transformation en objets pour une exploitation immédiate par une application, ou encore la transformation en d'autres documents, XML ou non, après filtrage et réorganisation du contenu. Ce dernier cas est illustré par la transformation à l'aide de feuilles de style XSLT.

La façon de lire un document XML et de représenter la position relative ou absolue de chaque élément à l'intérieur est également standardisée. On recourt au langage XPATH. Ce langage sert également à retrouver des fragments XML dans les bases de données de documents XML.

2 Représentation des connaissances

Bien que très exploitées, les descriptions syntaxiques à base de méta-données présentent cependant des limites. Il est donc naturel de penser à la construction et à l'utilisation d'ontologies, afin de partager au sein d'une communauté, un vocabulaire commun et une représentation des connaissances commune.

2.1 Ontologies

Des recherches récentes (Abel et al., 2004) montrent que dans un contexte de e-learning, il faut généralement au moins deux ontologies, celle du domaine de la formation proprement dit (types de documents, média, etc.) et celle du domaine d'application sur lequel porte la formation où les concepts en représentent les éléments de connaissance.

La première utilisation de telles ontologies est l'indexation des documents et des ressources sur les concepts de celles-ci. On est dans ce cas dans la même utilisation que les métadonnées. Une seconde caractéristique rencontrée dans les mémoires de formation concerne les possibilités de navigation. Les documents sont toujours indexés sur les concepts, mais il est possible de passer d'un concept aux concepts voisins, d'accéder de proche en proche aux ressources associées et illustrant les éléments de connaissance de la formation.

La nature de l'ontologie permettant une telle navigation est simple même si le nombre de concepts peut-être important. Des relations lient les concepts de l'ontologie et les domaines et co-domaines d'une relation définissent des concepts voisins. Les constructeurs de mémoires peuvent limiter à un sous-ensemble les concepts et relations qui seront en fait accessibles. Ils définissent pour cela des concepts points d'entrée et la clôture par navigation à partir de ces concepts d'entrée représente l'ensemble des concepts accessibles.

La réutilisation d'ontologies, même celle de nature simple – l'ontologie ne possède pas de concepts construits à partir de contraintes, par exemple – n'est cependant pas immédiate. Même s'il est possible de représenter l'ontologie dans un formalisme particulier, il faut cependant dépasser le fait qu'une ontologie est toujours construite dans un but précis. Des extensions ontologiques toujours possibles permettent des adaptations locales.

2.2 OWL

Le langage d'ontologie Web OWL (OWL, 2004) sert à décrire des concepts (appelés dans ce cas des classes) et des relations ou propriétés.. La sémantique formelle OWL indique comment déduire des conséquences logiques, c'est-à-dire les faits qui ne sont pas littéralement présents dans l'ontologie mais qui peuvent en être déduits. Une ontologie diffère d'un schéma XML en ce qu'il s'agit de la représentation de connaissances, et non de celle de format de documents. Elle possède par ailleurs un caractère beaucoup plus expressif. Le langage OWL offre trois sous-langages, d'expressivité croissante.

Le langage OWL Lite concerne les utilisateurs qui ont principalement besoin d'une hiérarchie de classifications et de fonctionnalités de contrainte simples. Par exemple, bien qu'OWL Lite gère des contraintes de cardinalité, ce langage ne permet que des valeurs de cardinalité de 0 ou 1.

Le langage OWL DL concerne les utilisateurs qui souhaitent une expressivité maximum sans perdre la complétude du calcul (toutes les inférences sont assurées d'être prises en compte) et la décidabilité (tous les calculs seront terminés dans un

intervalle de temps fini) des systèmes de raisonnement. Le langage OWL DL inclut toutes les structures de langage de OWL, avec des restrictions comme la séparation des types (une classe ne peut pas aussi être un individu ou une propriété, une propriété ne peut aussi être un individu ou une classe). OWL DL est ainsi nommé en raison de sa correspondance avec la logique de description.

Le langage OWL Full est destiné aux utilisateurs qui souhaitent une expressivité maximale et la liberté syntaxique de RDF sans garantie de calcul. Une autre différence significative par rapport à OWL DL réside dans la possibilité de marquer un objet `owl:DatatypeProperty` comme étant un objet `owl:InverseFunctionalProperty`. Il est peu probable qu'un système de raisonnement puisse mettre en œuvre toutes les caractéristiques de OWL Full.

Chacun de ces sous-langages représente une extension par rapport à son prédécesseur plus simple, à la fois par ce qu'on peut exprimer légalement et par ce qu'on peut conclure de manière valide.

2.3 Limites d'utilisation du formalisme OWL

Actuellement, le standard OWL, semble le plus communément utilisé et permet la représentation d'ontologies simples. L'utilisation du formalisme OWL pose cependant essentiellement deux types de problèmes. Tout d'abord, il ne permet pas de représenter certaines modélisations de concepts. Deuxièmement, formalisme basé sur la logique de description, il entraîne des complications liées à cette logique lorsqu'il s'agit d'opérationnaliser une ontologie. Illustrons ces deux problèmes par deux exemples issus du domaine du e-learning.

OWL possède plusieurs possibilités de constructions de concepts, mais aucune ne peut être basée sur la comparaison des valeurs des propriétés de type données, numériques ou bien dates par exemple. Ainsi, on peut exprimer le concept de test auquel est associé une valeur numérique pour son évaluation (note absolue ou pourcentage de réussite), mais le concept de succès à un tel test lorsque la note est supérieure à une valeur v est inexprimable en OWL¹. Il faudrait pouvoir écrire une règle du type :

Si
un individu est de type *Test*, s'il a une évaluation n et que cette évaluation n est supérieure à v

alors
ce test est un individu de type *Succès*

L'opérationnalisation consiste à créer une base de connaissance dans laquelle cohabitent l'ontologie elle-même et une population de l'ontologie, et à permettre à un moteur d'effectuer des inférences sur l'une et l'autre. Sans entrer dans des détails trop techniques, nous pouvons cependant préciser que les contraintes dues à l'utilisation de la logique de description imposent une modélisation de l'ontologie compatible avec le niveau OWL-DL. Ceci permet dans un premier temps de vérifier la consistance de l'ontologie, nécessaire à tout moteur d'inférence.

¹ Sauf au prix de développer en OWL un langage de règles et d'avoir des outils interprétant sa sémantique.

Ainsi, une relation R définie entre un concept A et un concept B permet de définir des assertions entre des individus du concept A et des individus du concept B telles que $R(unA, unB)$. Mais, vouloir définir des assertions du type $R(unA, B)$ ou $R(A, B)$, entraîne une modélisation au niveau de OWL full et aucun moteur d'inférence ne supporte ce niveau. Par exemple, vouloir exprimer qu'un document traite du sujet Langage de programmation n'est pas possible dès lors que Langage de programmation est un concept. Une solution consiste à appliquer une modélisation particulière qui rend beaucoup plus difficile la compréhension de l'ontologie et par suite sa réutilisation.

Le formalisme OWL ne permet pas non plus d'exprimer certaines caractéristiques ontologiques pourtant intéressantes et couramment employées. Supposons avoir résolu le problème du sujet d'un document d'un point de vue logique. Il est possible de définir par exemple un document portant sur un sujet particulier. Si ce sujet spécialise un sujet plus général, il ne sera pas possible d'exprimer en OWL que ce document traite également de ce sujet plus général. Ce problème est en fait un cas particulier de l'impossibilité de définir la composition de relations en OWL.

Il apparaît donc nécessaire de définir en plus de l'ontologie elle-même, des règles permettant d'étendre et d'adapter les capacités des moteurs d'inférences. Si l'on veut réutiliser une ontologie à des fins d'opérationnalisation, sur différentes populations, il est donc nécessaire de pouvoir d'une part exporter l'ontologie dans un formalisme tel que OWL, et d'autre part donner l'ensemble des règles étendant moteur d'inférence en utilisant un formalisme particulier.

2.4 Les graphes conceptuels

Le formalisme des graphes conceptuels (GC) est un formalisme de représentation des connaissances, de la famille des réseaux sémantiques. Il a été proposé par Sowa en 1984. Une formalisation des graphes conceptuels simples a été proposée par (Chein & Mugnier, 1992).

2.4.1 Présentation du formalisme

Le support

Le vocabulaire conceptuel d'un domaine est décrit sous forme d'un support constitué d' :

- une hiérarchie de types de concepts (notée T_c), muni d'une relation d'ordre partiel \leq . Le type universel est le type maximal, noté T .
- un ensemble partiellement ordonné de relations (noté Tr) éventuellement organisé en hiérarchie.
- un ensemble de graphes étoiles en bijection avec Tr et indiquant la signature de chaque type de relation (c'est-à-dire son arité et le type de concept maximal pour chacun des nœuds-concepts liés par une telle relation),

- un ensemble M de marqueurs individuels : ces marqueurs permettent de désigner les instances,
- une relation de conformité entre les types de concepts et les marqueurs. Elle permet de préciser pour chaque marqueur le plus petit type de concept compatible avec ce marqueur. Cela permet donc de préciser le type de concept associé à l'instance désignée par ce marqueur.

Remarques :

- Sowa imposait que la hiérarchie de types de concepts ait une structure de treillis : elle était donc dotée d'un type maximal T (le type universel, noté T), d'un type minimal (le type absurde, noté \square) et tout couple de types de concepts t1 et t2 admettait un plus petit super-type commun et un plus grand sous-type commun. Mais les travaux ultérieurs, en particulier ceux de Chein et Mugnier, ont relâché cette contrainte de treillis.
- De même, certains auteurs imposaient également à Tr une structure de treillis.

Outre les types élémentaires, on peut aussi définir des types (appelés types définis) à partir d'un type de concept ou d'un type de relation et ce grâce à un graphe de définition.

Par exemple,

```
deftype Cours-anglais (x) is [Cours : x] - (concernant)-
[Matière : anglais]
```

La base de Graphes conceptuels

Une fois établi ce support (qui correspond à la connaissances terminologique et permet de représenter une ontologie), on peut construire une base de graphes conceptuels (ces GCs correspondant à la connaissance assertionnelle).

Un graphe conceptuel construit sur un support est un graphe bipartite étiqueté, connexe, composé de deux types de nœuds (ces nœuds sont étiquetés, les étiquettes respectant un certain nombre de contraintes) :

des nœuds-concepts (étiquetés chacun d'une part par un type de concept figurant dans la hiérarchie de types de concepts, et d'autre part par un référent correspondant soit à un marqueur individuel dans le cas d'une instance bien identifiée, soit à un marqueur générique),

et des nœuds-relations, étiquetés chacun par un type de relation.

La Figure 1 montre un exemple de graphe, G1, exprimant que dans le lycée Masséna, il existe un professeur *x qui enseigne un cours *c concernant l'anglais comme matière et ce, au niveau de la classe Terminale A.

Le même graphe peut également s'exprimer sous la forme linéaire suivante :

```
[Professeur : *x ]
```

```
- ( dans) - [Lycée : Masséna)
- (enseigne) - [Cours : *c]
  - (concernant) - [Matière : anglais]
  - (niveau) - [Classe : TerminaleA]
```

Plusieurs opérateurs sont définis sur les graphes conceptuels et permettent de dériver un GC à partir d'un ou plusieurs autres GCs :

- copie d'un graphe,
- opérations élémentaires de spécialisation :
 - o simplification par suppression de nœuds jumeaux,
 - o restriction sur les étiquettes d'un nœud-concept ou d'un nœud-relation,
 - o jointure élémentaire sur 2 sommets ayant la même étiquette,
- opérations de généralisation :
 - o addition d'un nœud jumeau,
 - o projection d'un graphe conceptuel sur un autre graphe conceptuel,
 - o jointure étendue,
 - o jointure maximale de deux graphes conceptuels.

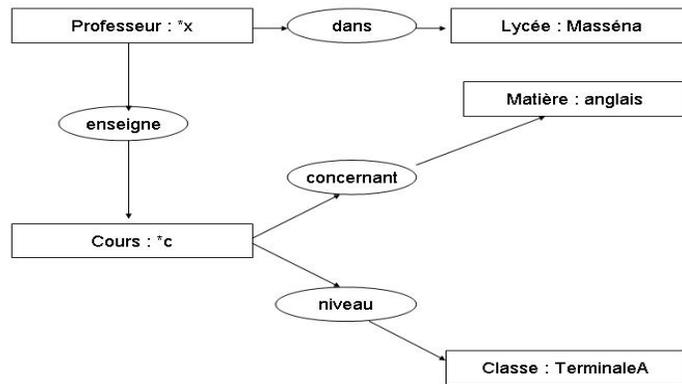
L'opérateur de projection permet de faire projeter un GC sur un autre GC. Plus précisément, c'est un morphisme de graphes, admettant que l'image d'un concept soit d'un type égal ou plus spécialisé que le type du concept et / ou ait un référent individuel et non plus générique.

Par exemple, si au niveau de la hiérarchie de types de concepts, Professeur est un sous-type d'Enseignant et Lycée un sous-type d'Etablissement-scolaire, alors le graphe G2

```
[Enseignant : *ens]- ( dans) - [Etablissement-scolaire : *z]
- (enseigne) - [Cours : *cours] - (concernant)- [Matière : *mat]
```

va se projeter sur le graphe G1 précédent en faisant les correspondances suivantes : *ens ↔ *x, *z ↔ Masséna, *cours ↔ *c, *mat ↔ anglais,

La contraction permet de remplacer le graphe définissant un type de concept par ce dernier, alors que l'expansion permet de remplacer un type de concept défini par son graphe de définition.



.Figure 1 : Exemple de graphe conceptuel simple

Par exemple, la contraction du graphe G1 précédent permettra d'obtenir le graphe:

```
[Professeur : *x ]
  - ( dans ) - [Lycée : Masséna)
  - (enseigne) - [Cours-anglais : *c] - (niveau) - [Classe :
TerminaleA]
```

L'expansion de ce dernier graphe permettra de retrouver G1.

(Sowa, 1984) définit la fonction ϕ qui permet d'associer à tout graphe conceptuel une formule de la logique du premier ordre (en particulier une formule positive conjonctive dans le cas des graphes conceptuels simples).

Par exemple, au graphe G1 précédent :

```
[Professeur : *x ]
  - ( dans ) - [Lycée : Masséna)
  - (enseigne) - [Cours : *c]
                    - (concernant) - [Matière : anglais]
                    - (niveau) - [Classe : TerminaleA]
```

correspondra la formule logique suivante:

```
 $\phi (G1) =$ 
 $\exists x \exists c : \text{type} (x, \text{Professeur}) \wedge \text{type} (c, \text{Cours}) \wedge \text{type} (\text{Masséna},$ 
 $\text{Lycée}) \wedge \text{dans} (x, \text{Masséna}) \wedge \text{enseigne} (x, c) \wedge \text{concernant} (c,$ 
 $\text{anglais}) \wedge \text{type} (\text{anglais}, \text{Matière}) \wedge \text{niveau} (c, \text{TerminaleA}) \wedge \text{type}$ 
 $(\text{TerminaleA}, \text{Classe})$ 
```

(Chein & Mugnier, 1992) ont doté le modèle des GCs d'une sémantique logique complète.

(Chein & Mugnier, 1997) ont ensuite introduit la notion de graphes imbriqués. On peut ainsi exprimer des contextes (par exemple, situation, proposition) par des concepts dont le référent est un GC imbriqué : on les appelle concepts contextuels.

La famille SG dont la complexité est approfondie dans (Baget et Mugnier, 2002a, 2002b) permet de décrire :

- des graphes conceptuels simples,
- des graphes conceptuels imbriqués,
- des règles de graphes conceptuels : une règle $G1 \Rightarrow G2$ est constituée de deux graphes conceptuels ($G1$ correspondant à une hypothèse, $G2$ à une conclusion) et d'un ensemble de points d'attache correspondant à des liens de connexion entre $G1$ et $G2$.
- des contraintes : on distingue les contraintes positives, décrivant des contraintes à respecter par la base de GCs) et des contraintes négatives (correspondant à des situations ne devant pas se trouver dans la base de GCs.

Notons enfin qu'une standardisation a été proposée : CGIF (Conceptual Graph Interchange Format) .

2.4.2 Exemples de quelques travaux Français sur les CGs

Le système CGKAT (Martin, 1997) permet d'associer une base de GCs à un document structuré, en reposant sur une ontologie basée sur une extension de WordNet. L'utilisateur peut poser des requêtes soit sur la base de GCs, soit sur le contenu des documents et CGKAT exploite alors la projection sur les GCs pour retrouver les éléments de documents pertinents comme réponses à cette requête.

(Rivière & Dieng-Kuntz, 2002) proposent des modèles pour introduire la notion de point de vue dans le formalisme des graphes conceptuels : le modèle C-VISTA pour décrire des points de vue dans un support et le modèle CG-VISA pour les représenter dans une base de GCs.

(Dieng & Hug, 1998) proposent le système MLTIKAT permettant de comparer et intégrer deux ontologies représentées dans le formalisme des GCs et de comparer et intégrer deux bases de graphes conceptuels en exploitant diverses stratégies pour résoudre des conflits lors de leur intégration.

Parmi les plates-formes de développement de GCs, citons :

- Cogito (Haemmerlé & Guinaldo, 1999),

- Cogitant (une bibliothèque C++ pour développer des applications de GCs dans le cadre de la famille SG) (Genest, 2000),
- (Southey & Linders, 1999) en Java.

Le lien entre les GCs et RDF(S) est étudié dans (Corby et al., 2004)

2.4.3 Graphes conceptuels et Web sémantique

Lors de l'avènement du Web sémantique, plusieurs chercheurs ont adopté directement le formalisme des GCs pour représenter ontologies et annotations sémantiques dans le contexte du Web sémantique : par exemple, le système WebKB (Martin & Eklund, 2000) – qu'on peut considérer comme une extension de CGKAT orientée au Web - permet d'indexer des documents Web par des graphes conceptuels et de les interroger en exploitant la projection ainsi qu'une ontologie reposant sur une extension de WordNet.

D'autres chercheurs ont préféré reposer sur une correspondance entre les GCs et RDF(S) : par exemple, le moteur de recherche sémantique CORESE (Corby et al., 2004) décrit dans un autre chapitre est basé sur une telle correspondance.

Dans (Gerbé & Mineau, 2002), les auteurs suggèrent d'utiliser les graphes conceptuels comme langage intermédiaire pour permettre la traduction automatique de structures de connaissances entre différents formalismes de représentation des connaissances ; ils décrivent le métamodèle de RDF(S) dans les GCs pour illustrer l'intérêt de cette approche dans le cadre du Web sémantique.

Des extensions de RDF(S) pour décrire des connaissances contextuelles (par exemple des axiomes) et inspirées des GCs ont par ailleurs été proposées (Delteil et al., 2003).

2.4.4 Graphes conceptuels et eLearning

Quelques travaux exploitent le formalisme des GCs : (Kokkoras et al., 2001) suggèrent une approche à base de connaissances pour organiser et accéder à des ressources éducatives et étudient un portail Web pour des « learning objects » où sont encodées des métadonnées dans le formalisme des GCs, de manière à utiliser ensuite les mécanismes d'inférence pour offrir par exemple la génération de cours dynamique basée sur le profil de l'utilisateur.

Kokkaros et Vlahava (2002) proposent un agent intelligent qui code sous forme de GCs les métadonnées des « learning objects » auxquelles accède l'utilisateur, les stocke dans un répertoire personnel GC-LOM, puis utilise des connaissances du domaine exprimées sous forme de règles de graphes, pour jouer auprès de l'utilisateur le rôle d'un assistant lui offrant une mémoire éducative personnalisée. Cet agent est implémenté au-dessus de Cogitant.

De tels travaux, bien que peu nombreux encore, montrent l'intérêt des graphes conceptuels tant pour le Web sémantique que pour le e-Learning.

3 Modélisation d'activités

Le standard qui semble s'imposer actuellement pour la modélisation des activités est IMS Learning Design. Il a été précédé par plusieurs propositions et notamment par EML ou encore Palo (Rodríguez-Artacho & Verdejo Maíllo, 2004).

EML (Educational Modelling Language) (Koper, 2001) est un modèle intégrateur de métadonnées (en XML) qui prend en compte non seulement des éléments pour décrire les ressources pédagogiques et leur contenu (texte, tâches, tests, devoirs) mais aussi le rôle, les liens, les interactions et les activités des étudiants et des apprenants. Le modèle EML, n'est pas un standard mais il intègre des idées venant des standards IMS, IEEE-LTSC, Dublin Core et ADL-SCORM. EML est le langage à l'origine du standard IMS Learning Design (IMS, 2003).

IMS Learning Design (IMS LD) est un langage qui favorise la description des processus d'apprentissage. Il aide les concepteurs pédagogiques à modéliser le contenu, les rôles des différents acteurs dans une activité pédagogique et les services nécessaires pour atteindre les objectifs d'apprentissage. Il soutient également les divers types d'apprentissage collaboratif. Cet aspect est particulièrement important pour sa reconnaissance à la fois dans le cadre de la formation à caractère commercial et globalement dans l'éducation.

Alors que EML intègre dans un unique langage tous les aspects pour la modélisation d'un processus d'apprentissage, IMS LD est compatible avec plusieurs autres spécifications comme IMS Learning Resource Meta-data (IMS, 2004) et IMS Question and Test Interoperability (IMS, 2003). De plus, contrairement à EML, IMS Learning Design intègre trois niveaux de représentation (A, B, C) qui seront décrits dans cette section.

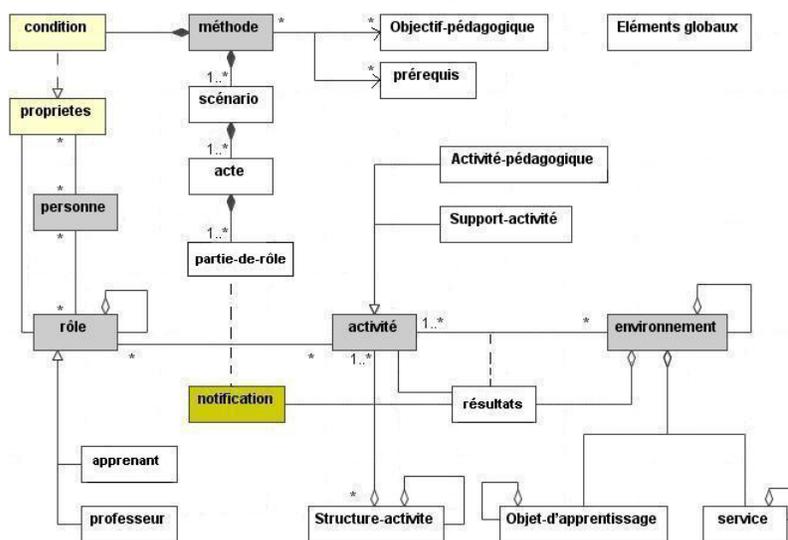


Figure 2 : Modèle conceptuel du niveau A

Le schéma de la Figure 2 est une représentation formelle d'Unité Pédagogique. Il y a trois niveaux de représentation qui s'intègrent de la manière suivante : le niveau A est intégré dans le niveau B et ce dernier est intégré dans le niveau C.

Le niveau A est celui de base. Il est représenté à la figure 1, par les cases blanches. Ce niveau permet la spécification des activités pédagogiques ordonnées dans le temps et effectuées par les *apprenants* et par les *professeurs* dans le contexte d'un milieu qui consiste en *objets d'apprentissage* et *services*.

L'élément de base est la *méthode* qui est une séquence d'éléments permettant de définir la dynamique du processus d'apprentissage. Une méthode part des conditions initiales (pré requis) qui doivent être satisfaites et qui a comme but d'atteindre certains objets d'apprentissage. Elle consiste en un ou plusieurs scénarios qui sont exécutés en parallèle.

En général, deux rôles peuvent être interprétés : étudiant et professeur. Les activités qui doivent être effectuées dans une méthode sont référencées par un identificateur, celles-ci étant stockées à l'extérieur de la méthode. Parmi les types d'activités utilisées on distingue les activités d'apprentissage, les activités de support et les structures d'activités. Ces dernières regroupent en séquences d'autres activités (d'apprentissage ou de support). En général, les activités d'apprentissage sont effectuées par les apprenants et celles de support par les professeurs. Les activités se déroulent dans un certain milieu qui met à disposition des services et/ou des objets d'apprentissage.

Le niveau B intègre le niveau A en ajoutant des propriétés et des conditions permettant une représentation du processus d'apprentissage plus avancée et plus flexible. Les propriétés sont utilisées pour stocker les informations sur une personne ou sur un groupe de personnes. Les conditions permettent de décider de l'évolution d'un scénario pédagogique à un moment donné. Par l'évaluation d'une expression on peut décider, en fonction de son résultat, quel parcours suivra le scénario.

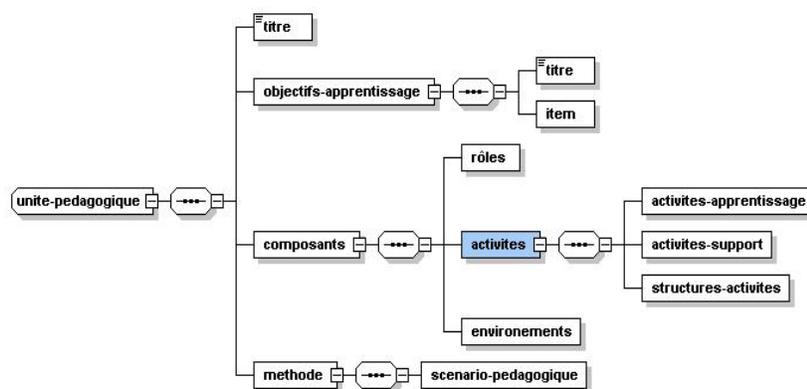


Figure 3 : La structure générale d'une Unité Pédagogique

Le niveau C intègre le niveau B en y ajoutant les *notifications*. Ce niveau C permet la transmission des messages d'un rôle ou l'ajout de nouvelles activités associées à un rôle, conséquences de l'apparition des événements pendant le processus d'apprentissage.

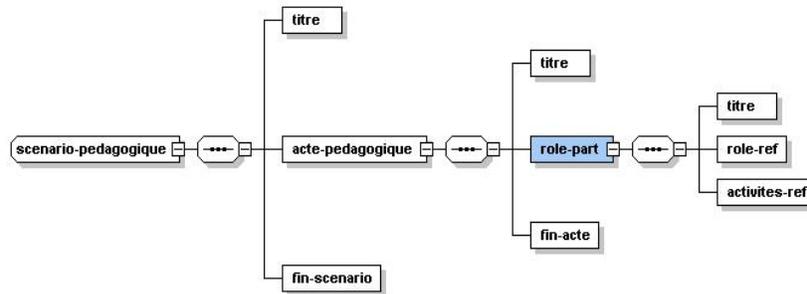


Figure 4 : La structure générale d'un Scénario Pédagogique

Les Figure 3 et Figure 4 représentent les structures générales d'une Unité Pédagogique ainsi que celles d'un Scénario Pédagogique, tel que définies dans le modèle conceptuel IMS LD. Le projet Cepiah (Trigano & Giacomini, 2004) s'en est inspiré pour la conception de modèles pédagogiques implantés dans des sites web éducatifs.

4 Langages de règles et de requêtes

La section 2.3 illustrant les insuffisances de OWL montre l'impossibilité de représenter une ontologie uniquement à l'aide de documents OWL. En particulier, le formalisme du typage des données OWL n'est nettement pas assez expressif pour exploiter les types de données lors de la définition des concepts, (utilisation du prédicat « supérieur à » pour les nombres) des ontologies du Web sémantique. En attendant le développement d'extensions de ce formalisme, il convient donc de représenter une ontologie lorsque cela est nécessaire par au moins deux documents : un premier document OWL, décrivant classes, relations et comprenant les règles exprimables en OWL et un second document contenant les règles non exprimables en OWL.

Il est par conséquent utile d'exprimer ces règles en utilisant un langage de règles standard du W3C ou encore à l'aide d'un langage reconnu par un moteur d'inférence recommandé. Il n'est pas le lieu ici de développer une catégorisation générale des règles, mais précisons simplement que nous nous intéressons aux règles d'intégrité (ou contraintes) et aux règles de dérivation.

4.1 Langages de règles

4.1.1 Moteur d'inférence Jena2

L'éditeur d'ontologie « Protege² », acceptant le formalisme OWL, est associé au moteur d'inférence Jena. Il est possible d'exprimer dans le langage de règles lié à ce raisonneur des conditions utilisant les opérateurs binaires relationnels ainsi que certaines fonctions mathématiques. Ainsi, la règle décrite en 2.3 où v correspond à une valeur numérique, s'exprimera de la façon suivante :

```
[success:  
(?x rdf:type Test), (?x value ?a),  
greaterThan(?a, v) -> (?x rdf:type Success]
```

4.1.2 OWL-E

Il est possible d'étendre OWL DL avec des prédicats et expressions portant sur des types de données. Ainsi Jeff Z. Pan et Ian Horrocks proposent OWL-E³, un langage permettant aux utilisateurs de définir leur propres types de données et leurs propres prédicats dans leurs ontologies. Bien que plus expressif que OWL-DL, OWL-E reste décidable.

OWL-E ne répond pas aux exigences de généralité attendues. En effet, si l'on veut définir deux comparaisons, il faut construire autant de prédicats ($>_{10}$, $>_{15}$ par exemple). De plus, aucun support opérationnel n'est disponible. Aucun raisonneur actuel n'est en mesure d'interpréter de tels prédicats.

4.1.3 RuleML

Disposer d'un langage de règles est devenu un élément crucial pour le Web depuis la mise en pratique des règles de transformation pour la génération de documents à partir d'une base XML. L'utilisation de règles joue un rôle important aussi bien dans les systèmes à base de connaissance que dans le domaine des agents intelligents. Il semble nécessaire que les efforts soit engagés afin de converger vers un langage de règles à balise partagé.

RuleML⁴ répond à cette exigence. C'est un langage XML qui permet de représenter différentes catégories de règles. Il existe bien évidemment une DTD et un schéma XML pour décrire la structure de ce langage. Des traducteurs vers les formats utilisés par les principaux systèmes experts existent.

² <http://protege.stanford.edu/>

³ dl-web.man.ac.uk/Doc/IMGTR-OWL-E.pdf

⁴ <http://www.ruleml.org/>

4.1.4 SWRL

SWRL⁵ (Semantic Web Rule Language) est une proposition de standard pour la spécification de règles, soumis au W3C. Il combine les sous langages OWL DL et OWL Lite de OWL avec certains sous langages de RuleML. En particulier, SWRL définit plusieurs opérateurs mathématiques. Une ontologie OWL et un schéma RDF existent, mais fournissant seulement une description partielle de la syntaxe SWRL, l'ontologie OWL étant plus complète que le schéma. Actuellement, aucune API Java ne supporte la syntaxe SWRL, bien qu'un éditeur de règles soit disponible pour Protégé.

4.2 Langages de requêtes

Il existe de nombreux langages de requêtes mais étant donné le contexte de l'étude nous nous intéressons aux langages de requêtes basés sur RDF (RDF-Query, 2004). Ces langages offrent de nombreuses similarités. RDF décrivant des structures basées sur des triplets (sujet, prédicat, objet), il est possible de concevoir une requête comme un graphe RDF (les variables de la requête peuvent être des nœuds ou des arcs du graphe) mis en correspondance avec un autre graphe représentant une base de connaissances.

Une intéressante comparaison des divers langages de requêtes RDF peut être trouvée en (RDF-Query, 2004), mais citons simplement deux langages de requêtes RDF utilisés dans des projets de e-learning, QEL (Query Exchange Language) développé au cours du projet Edutella⁶, et TRIPLE⁷ utilisé dans le projet (Personal reader for e-Learning⁸) dont l'objectif est de fournir à l'apprenant une interface personnelle pour naviguer parmi des ressources.

RDQL (RDQL, 2004) est un autre langage de requête RDF, qui semble le plus utilisé. Il résulte d'une évolution de plusieurs langages et a été soumis au W3C par Hewlett-Packard. On peut trouver diverses implémentations de RDQL (Java, PHP, Perl).

Mentionnons également le projet OWL-QL (Fikes et al., 2003), qui poursuit le développement du langage de requête DQL pour le langage de représentation OWL. Il s'agit de développer un langage de requête utilisés par des agents lors de dialogue autour de connaissances représentées en OWL. OWL-QL est un langage formel qui spécifie les relations sémantiques entre une requête, la réponse et la base de connaissances contenant les éléments de réponse. OWL-QL possède une implémentation en java.

⁵ <http://www.w3.org/Submission/SWRL/>

⁶ <http://edutella.jxta.org/spec/qel.html>

⁷ <http://triple.semanticweb.org/>

⁸ <http://www.personal-reader.de/>

4.3 Langage de description de services

Si l'on considère qu'un service est une ressource, le développement des services web et surtout des services web sémantiques impose la nécessité de décrire ces services de façon sémantique et d'utiliser pour cela un formalisme spécifique. Dans le cadre des services web utilisés dans le domaine commercial, différents standards ont vu le jour, concernant par exemple le transport des informations (SOAP⁹) et la description technologique (WSDL¹⁰). De même, dans le cadre du web sémantique, on dispose de deux langages correspondant à deux modèles de services différents, permettant de décrire de façon sémantique les services. Il s'agit de OWL/S¹¹ et de WSMO¹².

Les services qui pourraient être utilisés dans le cadre du e-learning ne semblent pas pouvoir échapper à cette nécessité. Les descriptions sont stockées dans des registres qui sont des applications pouvant répondre à des requêtes. Celles-ci ont plusieurs usages. En premier lieu, elles permettent à des agents humains de connaître les caractéristiques des services disponibles. En second lieu, elles servent à des agents ou à d'autres applications lorsqu'ils appellent les services correspondant. On atteint là le domaine de la composition de services.

5 Exemple

Un travail très intéressant faisant l'intersection entre le web sémantique et l'enseignement à distance est Edutella¹³ dont l'objectif est de permettre à des clients P2P d'émettre des requêtes décrites en RDF, et à utiliser des méta-données décrites aussi en RDF pour spécifier la sémantique de leurs données. La première application d'Edutella (qui a forgé son nom sur cet aspect (Ed(ucation) + Gn(utella)), se focalise sur l'échange de ressources pédagogiques bâties sur des schémas IMS/LOM et SCORM.

Le principal service proposé par Edutella est un service de requête : chaque pair doit poster les requêtes dans le langage d'échange Edutella Query Exchange Language (QEL) spécifié en RDF et sérialisé en XML. Ce langage présente plusieurs niveaux de complexité (RDF-QEL-1, RDF-QEL-2, etc ...) dépendants des capacités du pair qui répondra à la requête. Le fonctionnement standard proposé est que chaque pair recevant la requête l'exprime sous une forme logique, les connaissances qu'il gère devant être elles-mêmes décrites sous la forme de clauses logiques de Horn (ou plus simplement par la description des méta-données LOM).

Ainsi, la requête (exprimée en RDF-QEL-1) (tirée de l'exemple officiel) portant sur l'acquisition de livres d'Intelligence Artificielle :

⁹ <http://www.w3.org/TR/soap/>

¹⁰ <http://www.w3.org/TR/wsdl>

¹¹ <http://www.w3.org/Submission/2004/07/>

¹² <http://www.wsmo.org/index.html>

¹³ <http://edutella.jxta.org/>

```
<edu:QEL1Query rdf:ID='AI_Query_1'>
  <edu:hasVariable rdf:resource='#X' />
</edu:QEL1Query>
<edu:variable rdf:ID='X' >
  <rdf:type rdf:resource='www.lit.edu/types#AIBook' />
</edu:variable>
```

sera traduite en :

```
aibook(X) :- type(X, AIBook).
```

Tandis que la requête :

```
<edu:QEL1Query rdf:ID='AI_Query_2'>
  <edu:hasVariable rdf:resource='#Y' />
</edu:QEL1Query>
<edu:variable rdf:ID='Y' >
  <rdf:type rdf:resource='www.lit.edu/types#Book' />

<dc:title> Artificial Intelligence </dc:title>
</edu:variable>
```

sera traduite en :

```
aibook(X) :-
  title(X, ' Artificial Intelligence '), type(X, Book).
```

L'architecture d'Edutella, nécessite une infrastructure permettant de gérer les connaissances des pairs et pour cela l'environnement KAON (KAON, 2004) est un bon candidat. KAON est une plate-forme modulaire de gestion d'ontologies conçue à l'université de Karlsruhe. Elle est « open-source » et son développement a utilisé très massivement la technologie Java. KAON permet :

- d'éditer et de partager des ontologies définies en RDF(S), les triplets RDF pouvant être aussi bien stockés dans des bases de données natives XML que relationnelles.
- d'appliquer sur celles-ci un langage de requête spécifique
- et surtout d'exporter ces ontologies sous une forme logique (clauses de Horn de type Datalog)

Le fait que KAON permette à la fois d'utiliser des ressources de stockage de la connaissance complètement hétérogènes (bases de données relationnelles, natives XML, entrepôts de données), et propose une interface en quasi-logique du premier ordre pour accéder à celles-ci et leur appliquer des inférences (qui peuvent donc être des inférences autres que celles de classification) adosse donc l'échange de ressources pédagogiques proposé par Edutella. Il est à regretter que KAON n'offre aucune perspective de compatibilité réelle avec OWL.

Notons enfin que Edutella propose d'autres langages de requêtes pour appliquer une requête QEL sur les connaissances du pair : RQL qui est un langage de requête RDF, de même que TRIPLE qui permet en plus l'inférence logique, et enfin l'utilisation plus classique de XPath sur des bases de données natives XML, voire même de SQL.

6 Conclusion

Notre objectif était est de présenter certains langages du web sémantique, de vérifier leur utilisation effective dans le domaines du e-learning et d'essayer de les associer à une fonction principale. Leur adéquation aux besoins du e-learning s'améliore mais reste encore insuffisante, certainement parce que de nouvelles nécessités apparaissent, mais aussi parce que certaines difficultés théoriques subsistent.

Certes, l'utilisation massive du langage XML favorise la communication entre applications, mais elle met aussi en évidence les modèles conceptuels sur lesquels sont basés certains formalismes. Traduire d'un langage dans un autre pose de réelles difficultés et des choix doivent être adoptés lorsque les puissances expressives ne sont pas exactement les mêmes. Ainsi, il est possible de migrer de RDF vers OWL Lite ou encore des TM vers RDF mais à condition de respecter certaines contraintes.

La normalisation autour d'XML, la volonté de laisser en libre accès (sous réserve d'applications de licences) un maximum de résultats et le besoin de standards a favorisé le développement d'outils dont le e-learning bénéficie également. Cependant, sa syntaxe rébarbative rend très difficile la lecture de documents XML. Il est par exemple presque impossible de comprendre une ontologie à la seule lecture d'un fichier OWL.

Nous avons essayé de montrer également que les standards répondent à beaucoup d'attentes mais nous avons voulu en montrer également les limites.

Références

- ABEL M.-H., BARRY C., BENAYACHE A., CHAPUT B., LENNE D., & MOULIN C. (2004). Ontology-based Organizational Memory for e-learning. *Educational Technology & Society*, vol 7(3).
- ABEL M.-H., LENNE D., MOULIN C., & BENAYACHE A. (2003). Gestion des ressources pédagogiques d'une e-formation. *Document Numérique*, 7(1-2), 111-128.
- ALTHEIM M. (2002). *Data types for XML topic maps*, from <http://kmi.open.ac.uk/psi/datatypes.html>
- CHEIN M., & MUGNIER M.-L. (1992). Conceptual Graphs: Fundamental Notions. *Revue d'Intelligence Artificielle*, 6(4), 365-406.
- CHEIN M., & MUGNIER M.-L. (1997). Positive Nested Conceptual Graphs. In D. Lukose, H. Delugach, M. Keeler, L. Searle & J. F. Sowa (Eds.), *Lecture Notes in Artificial Intelligence, proceedings of ICCS'97* (Vol. 1257, pp. 95-109). Seattle, USA: Springer.

- CORBY O., DIENG-KUNTZ R., & FARON-ZUCKER C. (2004). *Querying the Semantic Web with the CORESE Search Engine*. Paper presented at the ECAI'2004, Valencia.
- DCMI. (2005). *Dublin Core Metadata Initiative*, from <http://dublincore.org/documents/2005/01/10/dcmi-terms/>
- DELTEIL A., FARON C. & DIENG-KUNTZ R. (2003). Le modèle des Graphes Conceptuels pour le Web Sémantique: Extensions de RDF et RDFS basées sur le Modèle des Graphes Conceptuels. *L'Objet, logiciel, base de données, réseaux (RSTI série)*, 3(9), p. 95-122.
- DIENG R., & HUG S. (1998). Comparison of "personal ontologies" represented through conceptual graphs. *Proc. of ECAI'98, Wiley & Sons*, p. 341-345.
- FIKES R., HAYES P. & HORROCKS I. (2003). *OWL-QL - A Language for Deductive Query Answering on the Semantic Web*, from <http://ksl.stanford.edu/projects/owl-ql/>
- GENEST D. (2000). *Extension du modèle des graphes conceptuels pour la recherche d'informations*. Université Montpellier II.
- GERBE O., & MINEAU G. W. (2002). *The CG Formalism as an Ontolingua for Web-Oriented Representation Languages*. Paper presented at the ICCS'2002, Borovetz.
- HAEMMERLÉ O., & GUINALDO O. (1999). CoGITO v3.3 : plate-forme de développement d'applications sur les graphes conceptuels. *Technique et Science Informatiques*, 18(9), 933-968.
- IEEE-LTSC. (2002). *LOM standard, document 1484.12.1*, from http://ltsc.ieee.org/wg12/files/IEEE_1484_12_02_D01_LOM_11404_binding.doc
- IMS. (2003). *Learning Design Specification, Version 1.0, Final Specification*, from <http://www.imsglobal.org/learningdesign/index.html>
- IMS. (2004). *Learning Resource Meta-data Specification*, from <http://www.imsglobal.org/metadata/index.html>
- ISO/IEC. (2002). *ISO/IEC 13250 Topic Maps, 2005*, from http://www.y12.doe.gov/sgml/sc34/document/0322_files/iso13250-2nd-ed-v2.pdf
- KAON (2004). *The Karlsruhe Ontology and Semantic Web Framework*, from <http://kaon.semanticweb.org/>
- KOKKORAS F., SAMPSON D., & VLAHAVAS I. (2001). *CG-PerLS: Conceptual Graphs for Personalized Learning Systems*. Paper presented at the 8th Panhellenic Conference on Informatics, Nicosia, Cyprus.
- KOKKORAS F., & VLAHAVAS I. (2002) "eLPA: An e-Learner's Personal Assistant", Proc. ICCS'02 Workshop on Applications of Conceptual Graphs, (electronic proceedings), Borovets, Bulgaria, 2002.
- KOPER R. (2001). Modelling units of study from a pedagogical perspective: the pedagogical meat-model behind EML.
- MARTIN P. (1997). *CGKAT: A Knowledge Acquisition and Retrieval Tool Using Structured Documents and Ontologies*. Paper presented at the ICCS'97.
- MARTIN P., & EKLUND P. (2000). Knowledge Retrieval and the World Wide Web. *IEEE Intelligent Systems*, 3(15), 18-25.
- NILSSON M. (2003). *RDF binding of LOM metadata*, from <http://kmr.nada.kth.se/el/ims/metadata.html>
- OWL (2004) Web Ontology Language from <http://www.w3.org/TR/owl-features/>
- PEPPER S. (2000). *The TAO of Topic Maps : Finding the way in the age of infoglut*. Paper presented at the XML Europe 2000 Conférence, Paris.
- RDF (2004) Resource Description Framework, from <http://www.w3.org/RDF/>
- RDF-QUERY. (2004). *RDF Query Survey*, from <http://www.w3.org/2001/11/13-RDF-Query-Rules/>

RDF/S (2004) RDF Vocabulary Description Language 1.0: RDF Schema from <http://www.w3.org/TR/rdf-schema/>

RDQL. (2004). *A Query Language for RDF*, from <http://www.w3.org/Submission/RDQL/>

RIBIERE M. & DIENG-KUNTZ R. (2002). A Viewpoint Model for Cooperative Building of an Ontology. *LNAI 2393, ICCS'2002*, p. 220-234.

RODRÍGUEZ-ARTACHO M. & VERDEJO MAÍLLO M. F. (2004). Modeling Educational Content: The Cognitive Approach of the PALO Language. *Educational Technology & Society*, 7(3), 124-137.

Southey F., & Linders J. G. (1999). *Notio - A Java API for developing CG tools*. Paper presented at the ICCS'99.

TRIGANO P., & GIACOMINI E. (2004). Toward a Web based environment for Evaluation and Design of Pedagogical Hypermedia. *Journal of Educational Technology & Society, IEEE Learning Technology Task Force, Vol. 7(n° 3)*.

XML Schema (2004) <http://www.w3.org/TR/xmlschema-1/>
<http://www.w3.org/TR/xmlschema-2/> (-2/)

XTM. (2001). *XML Topic Maps (XTM) 1.0*, 2004, from TopicMaps.org

Les Outils du Web Sémantique et du E-Learning

Faïçal Azouaou¹, Tuan-Dung Cao², Sylvain Dehors², Cyrille Desmoulins¹,
Rose Dieng-Kuntz², Catherine Faron-Zucker³

¹CLIPS-IMAG

BP53, 38041 Grenoble cedex 9

{Faical.Azouaou, Cyrille.Desmoulins}@imag.fr

²INRIA, ACACIA Project,

2004 route des Lucioles, BP 93, 06902 Sophia Antipolis cedex

{Tuan-Dung.Cao, Sylvain.Dehors, Rose.Dieng}@sophia.inria.fr

³MAINLINE, I3S, UNSA

930 route des Colles, bât ESSI, BP 145, 06930 Sophia Antipolis cedex

faron@essi.fr

Dans la perspective du Web sémantique pour des applications de E-Learning, nous présentons dans ce chapitre du rapport WebLearn des outils dédiés à la construction des composants du Web sémantique (outils de développement d'ontologies et outils de construction d'annotations), des outils dédiés à l'exploitation des ontologies et des annotations (outils de raisonnement, de recherche d'information) et des outils spécifiquement dédiés au E-Learning. Soulignons que les outils présentés offrent plusieurs facettes et pourraient intervenir dans plusieurs catégories à la fois. En effet la plupart des travaux d'envergure proposent des plateformes « intégrées » disposant d'une palette d'outils interopérables.

1 Outils de développement d'ontologies

WebODE, KAON et PROTEGE sont parmi les plates-formes les plus récentes et les plus complètes de développement d'ontologie pour le Web sémantique.

1.1 WebODE

WebODE est dédié au développement, à la gestion et à la population d'ontologies (Arpirez *et al.*, 2003) (Corcho *et al.*, 2002). Il est basé sur la méthode de construction d'ontologies Methontology. Il intègre également des fonctionnalités facilitant l'intégration d'ontologies dans des applications dédiées au Web Sémantique ou le développement d'applications à base d'ontologies.

Le modèle des connaissances de WebODE permet la représentation de concepts et de leurs attributs (de classe ou d'instance), de groupes de concepts définissant des partitions de classes, de relations entre concepts, de propriétés de relations, de constantes, d'axiomes et de règles (de la logique du premier ordre). Les relations entre

concepts sont celles prédéfinies dans WebODE (subclass-of, part-of, etc.) et celles binaires définies par l'ontologiste. Les propriétés des relations sont celles prédéfinies dans WebODE (propriétés algébriques) et celles définies par l'ontologiste.

Pour le développement d'ontologies, la plateforme WebODE intègre un ensemble de services pour (1) l'édition d'ontologies, (2) la construction d'axiomes et de règles et (3) leur application dans un moteur d'inférence, (4) l'interopérabilité des ontologies construites et enfin (5) leur documentation.

L'éditeur d'ontologies de WebODE est dédiée à la construction collaborative d'ontologies. Il est muni d'une interface Web qui permet la navigation dans une ontologie et l'édition d'une ontologie : un formulaire à remplir ou modifier par l'utilisateur est défini pour chaque type de composant du modèle des connaissances de WebODE. Cet éditeur intègre en outre un outil graphique d'aide à la construction de taxinomies de concepts et à la création de relations entre concepts.

L'éditeur d'axiomes et de règles de WebODE permet de créer des axiomes et des règles de la logique du premier ordre à l'aide d'une interface graphique. WebODE vérifie qu'un axiome ou une règle créée par l'utilisateur est syntaxiquement correcte et que le vocabulaire utilisé est défini dans l'ontologie ; l'axiome ou la règle est alors traduite en clause de Horn puis en règle Prolog.

Le moteur d'inférence de WebODE permet d'interroger une ontologie sur ses composants à l'aide de primitives dédiées, d'enrichir une ontologie par de nouvelles connaissances à l'aide des expressions Prolog fournies par l'éditeur d'axiomes et de règles, de détecter d'éventuelles inconsistances dans une ontologie en vérifiant les contraintes imposées par les règles et axiomes.

Les services d'interopérabilité de WebODE consistent en une API d'accès aux ontologies WebODE, des modules d'import/export d'ontologies WebODE dans les langages XML, RDF(S), OIL, DAML+OIL, FLogic, un module d'export d'ontologies WebODE dans le système JESS, un module d'export des axiomes et règles WebODE dans la syntaxe Prolog.

Enfin, WebODE intègre un service de documentation automatique des ontologies sous différents formats: tables HTML représentant les représentations intermédiaires de Methontology, taxinomies de concepts HTML, XML.

Pour l'intégration d'ontologies dans des applications dédiées au Web Sémantique et pour le développement d'applications à base d'ontologies, WebODE intègre des services de fusion d'ontologies, de génération de catalogues électroniques à partir de ses ontologies et d'importation de standards de classifications de produits et services.

1.2 KAON

KAON (KArlsruhe ONtology and Semantic Web framework) est une plate-forme dédiée à la gestion d'ontologies, spécialisée dans les applications d'entreprises. Elle est centrée sur l'intégration des technologies de gestion des ontologies avec celles propres à ce domaine d'application particulier, et notamment les bases de données relationnelles. KAON est utilisé dans des projets de recherche qui s'intéressent particulièrement à la réutilisation d'ontologies et l'évolution d'ontologies.

Le modèle d'ontologie de KAON est une extension de RDF(S) qui intègre des propriétés algébriques de relations (inverse, symétrique, transitive, etc.), des cardinalités, une modularité des connaissances, une méta-modélisation et enfin une représentation des informations lexicales.

La méta-modélisation permet la définition d'une ontologie par l'instanciation de la méta-ontologie du modèle de KAON qui intègre un ensemble de méta-données. Ce mode de définition d'une ontologie et la représentation d'informations lexicales attachées aux concepts ont pour but de faciliter la réutilisation des ontologies: KAON offre des fonctionnalités de recherche d'une ontologie à partir de ses méta-données et des informations lexicales associées à ses concepts.

L'architecture centrale de KAON est une API constituée d'un ensemble d'interfaces pour accéder aux éléments d'une ontologie (concepts, propriétés, instances) et un ensemble de modules de gestion des ontologies :

- Le module de « stratégie d'évolution » s'assure que tout changement opéré sur une ontologie préserve la consistance de celle-ci.
- Le module de « réversibilité des changements » conserve l'historique des changements licites opérés sur une ontologie de sorte que le système peut revenir à une version antérieure d'une ontologie à la demande de l'utilisateur.
- Le module de « détection de conflits de concurrence » détecte et résout les conflits résultant de mises à jour concurrentes de différents utilisateurs.
- Le module de « chargement optimisé » charge et transporte vers un client plusieurs éléments d'une ontologie en une requête unique.
- Le module de « réponse à une requête » retrouve les éléments d'une ontologie qui répondent à des critères donnés.

L'API KAON est pourvue de plusieurs implémentations dédiées chacune à un format différent de stockage des informations :

- Un module permet d'accéder à des ontologies représentées en RDF.
- Un module permet d'accéder à des ontologies représentées dans des bases de données relationnelles ; il est optimisé pour la gestion d'un grand nombre de créations/destructions de concepts dans une ontologie.
- Un module permet le passage du modèle des bases de données relationnelles au modèle des ontologies de KAON ; les tables d'une BD sont mises en correspondance avec des concepts et relations d'une ontologie et les opérations définies dans l'API KAON sur des ontologies sont traduites en opérations sur les BD.
- Un module permet l'interrogation d'une ontologie.

1.3 PROTEGE

Protégé (Noy *et al.*, 2001) est une plate-forme de développement d'ontologies et d'acquisition de connaissances développée par Stanford Medical Informatics, Stanford University School of Medicine ; c'est un logiciel libre (open source).

Protégé est muni d'une interface graphique qui permet aux développeurs d'ontologies de se concentrer sur leur tâche de modélisation conceptuelle, sans connaître ni penser à la syntaxe d'un langage de représentation.

Le modèle des connaissances de Protégé est flexible et son architecture plugin est extensible. Ces deux caractéristiques rendent Protégé rapidement et facilement adaptable à de nouveaux langages du Web Sémantique.

L'API Protégé permet à d'autres applications d'utiliser, d'accéder et d'afficher des bases de connaissances créées avec Protégé.

2 Les outils d'annotations

Il existe de nombreux outils d'annotation de documents Web. Notre objectif est d'en présenter une synthèse, en partant d'une catégorisation proposée pour classer ces outils.

Différentes classifications des outils d'annotations Web ont été proposées, parmi les plus utilisées celle introduite par (Koivunen *et al.*, 2000) et qui distingue les outils d'annotations selon la solution technique adoptée : les outils basés sur un Proxy et ceux basés sur la modification d'un navigateur Web.

Les outils de la première catégorie utilisent un Proxy pour stocker et fusionner les annotations, et prétraitent ensuite les documents annotés afin qu'ils soient affichables sur un navigateur Web standard. Les outils de la deuxième catégorie, sont basés sur la modification d'un navigateur Web, afin qu'il puisse fusionner le document avec ses annotations juste avant son affichage pour l'utilisateur.

Une deuxième approche pour classer les outils, que nous adoptons dans la suite, est basée sur l'opposition faite par (Caussanel *et al.*, 2002) à propos du Web Sémantique entre un pôle « computationnellement sémantique » et un pôle « cognitivement sémantique ». « Dans le Web Cognitivement Sémantique, l'accent n'est pas mis prioritairement sur la problématique d'une sémantique opérationnelle destinée à l'inférence automatique ou à l'exploitation par des agents logiciels, mais sur l'utilisation du Web par des utilisateurs humains engagés dans des activités de navigation » - où il est important donc que les annotations soient des objets graphiques visible à l'interface.

Dans le cas du Web cognitivement sémantique, l'utilisateur interagit directement avec les annotations, par conséquent, les outils d'annotation doivent être d'une grande utilisabilité, afin qu'ils ne soient pas rejetés par l'utilisateur final.

En adoptant cette dernière classification du Web sémantique cognitif et computationnel, nous pouvons identifier trois catégories d'outils d'annotations Web :

- Outils d'annotation graphique (Web non sémantique) ;
- Outils d'annotation graphique sémantique pour un Web sémantique cognitif ;
- Outils d'annotation sémantique pour un Web sémantique computationnel.

Pour chacune de ces catégories, il existe une multitude d'outils qui souvent se ressemblent par la solution technique adoptée et surtout par les fonctionnalités

offertes. Pour ces raisons nous avons choisi de ne présenter que deux ou trois outils d'annotation par catégorie pour l'illustrer.

2.1 Les outils d'annotation graphique (Web non sémantique)

Ces outils permettent à l'utilisateur d'annoter des pages Web avec des objets graphiques (soulignement, surlignement, post-it, ...). Ces annotations sont stockées en tant qu'objets graphiques sans aucune sémantique.

2.1.1 Imarkup

iMarkup est un programme qui permet aux membres d'un groupe d'annoter des pages Web, d'organiser et de partager ces annotations. Il inclut différents types d'annotations, tels que les notes collantes (Post-It.), des dessins libres et des annotations textuelles sur n'importe quelle page Web même sur des pages de format PDF. Il permet aussi d'annoter avec de la voix, et cela en enregistrant des sons, qui seront attachés à l'endroit voulu. Il suffit après de cliquer sur l'icône de l'annotation pour écouter le commentaire. Il est possible d'attacher des fichiers (de tout type) à un endroit précis de la page web. Les annotations seront stockées dans un serveur dédié, et à chaque fois que la page annotée est appelée, le programme, récupère l'annotation et l'attache à l'endroit précis. IMarkup inclut une fonction qui permet d'extraire l'ensemble des annotations faites sur un document ou plusieurs documents et de générer une vue synthétique. Cette visualisation est alors censée aider le lecteur à accéder rapidement aux points importants du document. Toutes les communications (requêtes) utilisent XML (iMarkup, 2005)

2.1.2 CritSuit

Dans CritSuit, les annotations sont stockées sur un serveur et peuvent être récupérées. Cet outil a été développé pour Foresight Institute afin de permettre des discussions critiques et d'améliorer la productivité de l'utilisation du web. Il se compose de trois modules logiciels intégrés qui permettent aux utilisateurs de commenter et de visualiser les documents hypertextes existants (CritLink), naviguer le Web en utilisant une interface graphique (CritMap), et enfin archiver les échanges des emails en utilisant les caractéristiques des l'hypertextes (CritMail). Quand nous utilisons CritSuite pour annoter un endroit précis d'un document, le texte annoté sera entouré de deux petits marqueurs colorés. La couleur du marqueur indique la nature du commentaire : vert pour le support, rouge pour un résultat, bleu pour un commentaire.

CritSuit utilise l'approche Proxy et il est simplement utilisé en préfixant l'url par <http://crit.org> par exemple pour visualiser la version annoté de semanticweb.org il suffit d'aller sur <http://crit.org/http://semanticweb.org>.

D'autres outils font partie de cette catégorie comme ThirdVoice, ComMentor, CoNote, Grant, ...

2.2 Outils d'annotation graphique sémantique

Ces outils permettent d'annoter graphiquement une page Web, tout en sauvegardant la sémantique de chaque annotation.

2.2.1 KIM

La plate-forme KIM (Knowledge and Information Management) (Popov *et al.*, 2003) est un produit de Ontotext Lab Bugarie. Il fournit infrastructure et services pour l'annotation, l'indexation, et la recherche sémantique automatique du contenu non structuré et semi-structuré. Les ontologies et les bases de connaissances sont manipulées dans les dépôts RDF(S), les intergiciels (middle ware) d'ontologie. KIM est basé sur l'outil de traitement automatique de langage naturel GATE, et est équipé d'une ontologie de niveau supérieur (KIMO Ontology) concentrée sur les entités nommées et d'une base de connaissance (KIM World KB) fournissant la couverture approfondie des entités d'importance générale. KIM est en cours de développement. Jusqu'à maintenant, cet outil permet d'annoter automatiquement des entités nommées dans le texte, d'effectuer une recherche basée sur la restriction sémantique.

Un composant d'annotation de KIM appelé KIMplugin réservé à Internet Explorer est disponible. Les annotations sont gérées par un serveur d'annotation. Les ontologies sont dans le format RDF(S) mais susceptible d'évoluer vers OWL.

2.2.2 Yawas

Yawas est un outil d'annotation de pages Web, qui exploite la technologie DOM ; il est réalisé en java script. Il inclut plusieurs fonctions, les plus intéressantes sont HIGHLIGHT, qui permet de mettre en surbrillance un texte mais surtout ANNOTATE, qui contrairement à la fonction précédente, crée des annotations informelles, non structurées sémantiquement. ANNOTATE permet de réaliser des annotations sémantiquement formelles, explicites et structurées. Chacune des annotations se compose de : type de document + thème +genre de sélection + commentaire + auteur. Yawas stocke les annotations sur le poste client ; le navigateur Web modifié peut ensuite retrouver ces annotations et les afficher (Laurent, 2000).

2.2.3 Annotea

Annotea est un projet du W3C, ayant pour but l'amélioration de la collaboration grâce au partage d'annotations sur le Web. On peut annoter un document complet ou une partie de ce document sans modifier le document lui-même. Annotea permet à tout internaute d'ajouter à une page Web des commentaires, des explications, qui sont stockés sur des serveurs dits « d'annotations », pour permettre à une communauté d'apprenants par exemple, de partager des commentaires sans que les autres internautes y aient accès ; les annotations peuvent être aussi stockées en local sur la machine de l'utilisateur, pour un usage personnel. Annotea exploite les standards avancés du W3C, comme les technologies RDF pour décrire les annotations et XPointer pour les localiser (Annotea, 2005).

2.3 Outils d'annotation sémantique

L'objectif de ces outils est d'assister l'utilisateur dans sa tâche d'annotation des pages Web, ces annotations étant destinées à être ensuite exploitées par des programmes informatiques, et non directement par l'utilisateur. La plupart de ces outils utilisent une interface divisée en deux, l'une offrant les méta-données contenues et structurées dans une ontologie, et l'autre partie affichant le document Web qu'on voudrait annoter.

L'utilisateur sélectionne la partie qu'il veut annoter sur le document, choisit une méta-donnée correspondante et valide ensuite l'annotation sémantique.

2.3.1 MnM

MnM est un outil d'annotation développé dans le cadre du projet AKT (Vargas-Vera *et al.*, 2002), par Open University, Royaume Uni. Il supporte l'annotation sémantique semi-automatique et automatique des pages Web. MnM intègre un navigateur avec un éditeur d'ontologie (RDF et DAML-OIL). Il fournit des APIs ouvertes permettant de se connecter à des serveurs d'ontologie et à des outils d'extraction d'information (grâce à une interface « plugin »). MnM travaille sur les documents textuels du Web et supporte quatre activités principales :

- *Navigation* : L'utilisateur peut naviguer dans une bibliothèque de modèles de connaissances qui se trouve dans un serveur d'ontologies « basé sur le web ».
- *Markup* : Cet ensemble est sélectionné pour former la base d'un mécanisme d'extraction d'information. Un corpus de documents est marqué manuellement. (L'utilisateur choisit des segments de texte dans le document et puis insère des balises sémantiques de l'ontologie dans le document).
- *Apprentissage* : Un algorithme d'apprentissage est lancé sur le corpus pour apprendre les règles d'extraction. Dans les versions précédentes MnM intégrait des outils d'extraction d'information : Marmot, Badge et Crystal de l'Université de Massachussets. La version courante utilise Almicare de l'Université de Sheffield.
- *Test et Extraction* : Le mécanisme d'extraction d'information est exécuté sur un corpus d'essai pour évaluer sa précision et son rappel. Puis il est exécuté sur un ensemble de documents pour retourner ces textes avec des annotations ajoutées dans le contenu.

MnM permet de choisir entre plusieurs ontologies, et offre un outil d'extraction d'information. Basé sur Amilcare, il est capable de reconnaître et puis annoter des informations concernant des entités nommées dans le texte. Une application typique de MnM est d'annoter des annonces des visiteurs du Kmi.

2.3.2 CREAM et Ont-O-Mat

Ont-O-Mat est l'implémentation de référence du framework CREAM. Il assiste l'utilisateur dans la tâche de créer les annotations à base d'ontologies. Ont-O-Mat comporte un navigateur d'ontologie pour l'exploration de l'ontologie et de ses

instances et un navigateur HTML qui affiche les parties du document à annoter. Il propose aussi une interface ouverte permettant l'ajout de nouveaux plugins. L'utilisateur au lieu d'annoter en éditant manuellement le texte de son annotation, peut sélectionner le texte à annoter sur le document, ensuite utiliser la technique du copier-coller des concepts de l'ontologie, afin d'en créer une nouvelle instance. Ont-O-Mat utilise le langage DAML-OIL, comme langage d'annotation (Handschuh *et al.*, 2001). Ont-O-Mat utilise également Amilcare comme module d'extraction d'information. La différence entre les deux outils est que MnM supporte des APIs pour accéder au serveur d'ontologie et peut gérer plusieurs ontologies en même temps. Ont-O-mat peut stocker des pages Web annotées en DAML+OIL en utilisant OntoBroker comme serveur d'annotation. Enfin, les annotations sont attachées avec le document sous la forme des balises XML dans le premier cas, comme des fichiers indépendants dans le second.

2.3.3 OLR3-Editor

L'approche utilisée par OLR3-Editor est similaire à celle de Ont-O-Mat, à la différence que, contrairement à Ont-O-Mat, OLR3-Editor exploite les navigateurs standards du Web, et par conséquent chaque utilisateur peut l'utiliser sans installer de nouveaux logiciels sur le poste client. La version actuelle de cet éditeur ne gère que les annotations stockées en local (Kuntze *et al.*, 2002).

Dans cette catégorie des outils d'annotation sémantique, nous pouvons inscrire également OntoMarkup, OntoAnnotate, SHOE Knowledge Annotator, ...

Nous avons présenté ici une multitude d'outils en les catégorisant selon qu'ils visent à créer des annotations graphiques, sémantiques, ou les deux à la fois (annotations graphiques sémantiques). Cette nouvelle classification offre l'avantage de différencier les outils produisant des annotations destinées à être manipulées par l'utilisateur des outils produisant des annotations orientées pour être exploitées dans les inférences des programmes informatiques (agents) sur le Web.

Cette classification nous permet aussi d'identifier les outils d'annotation graphiques non sémantiques et qui produisent donc des annotations sans aucune sémantique explicite (ex : IMarkup). Certains outils offrent des possibilités d'annotation semi-automatiques, pour MnM et Ontomat qui reposent sur Amilcare, après l'évaluation le taux de succès sur l'extraction automatique d'information par apprentissage atteint 70% sur les documents textes. Pour KIM qui utilise GATE et leur propre module d'extraction le résultat pour la tâche d'extraction d'information (pour les Name Entity) est 86% pour la précision et 82% pour le rappel.

3 Outils de raisonnement pour la recherche sémantique

Il existe différents outils de raisonnement pour le Web sémantique, généralistes ou spécialisés, qui permettent une recherche sémantique d'information, par la

manipulation formelle des ontologies et des annotations des ressources du Web. Nous présentons quatre plate-formes basées sur différents langages de requête pour RDF et différents formalismes de représentation des connaissances.

3.1 CORESE

Corese (COnceptual REsource Search Engine) (Corby *et al.*, 2004) est un exemple de moteur de recherche sémantique qui offre une recherche d'information guidée par des ontologies et basée sur des raisonnements dans le modèle des graphes conceptuels. Il permet grâce à son approche par traduction de RDF(S) de bénéficier à la fois des aspects « standard Web » de RDF et « formalisme de représentation » des graphes conceptuels.

Plus précisément, le logiciel Corese permet de charger des schémas RDFS et des annotations RDF, de les interpréter dans le formalisme des graphes conceptuels et d'interroger ainsi la base d'annotations en utilisant l'opérateur de projection des graphes conceptuels (Corby *et al.*, 2000). Les résultats obtenus sont traduits en RDF pour être retournés en réponse à la requête ; ils peuvent être groupés suivant différents critères.

Grâce à un algorithme de projection étendue, Corese permet également des recherches approchées grâce à des calculs de similarité entre les concepts de la requête et ceux de l'ontologie.

Le langage de représentation des connaissances de Corese permet d'exprimer les propriétés de transitivité, symétrie, réflexivité et relation inverse de OWL ainsi que les propriétés des classes telles que owl:intersectionOf, owl:disjointWith. Il prend aussi en compte les types de données de XML Schema.

Le langage de représentation des connaissances de Corese permet également de décrire des règles d'inférence avec une syntaxe XML/RDF (ces règles correspondant à des règles de graphes conceptuels) et Corese dispose d'un moteur d'inférence permettant de compléter la base d'annotations RDF avec les déductions issues de l'application des règles.

Corese dispose de deux langages de requêtes : l'un basé sur RDF, avec des variables et des opérateurs (tels que des comparateurs), dans une syntaxe compatible avec RDF ; l'autre offrant un pattern classique « select from where ».

Le moteur Corese est intégré dans un serveur web sémantique (serveur de servlets) qui le rend accessible via un navigateur Internet standard. Ce serveur comprend l'ontologie qui permet de structurer l'information.

Corese a été appliqué dans le cadre de plusieurs applications : le projet européen CoMMA, SAMOVAR avec Renault, Aprobation avec CSTB, KMP avec Telecom Valley, Gene Ontology, le projet MEAT sur les biopuces, le projet Ligne de Vie dans le domaine médical, et une application avec EADS.

3.2 SESAME

Sesame (Broekstra *et al.*, 2002) est une plate-forme générique pour le stockage persistant de données RDF(S) dans un SGBD et leur interrogation. Sesame est

indépendant du SGBD sous-jacent utilisé grâce à son API SAIL (Storage and Inference Layer) permet de coupler Sesame avec une grande variété de SGBD sans avoir à changer aucun autre des composants de la plate-forme.

Le langage de requête de Sesame est RQL. Il s'agit d'un langage d'interrogation de données RDF(S) défini par un ensemble de requêtes primitives, d'un ensemble de filtres élémentaires et d'un mécanisme de construction de nouvelles requêtes. Les requêtes primitives permettent de retrouver toutes les classes de l'ontologie, de retrouver toutes les relations de l'ontologie et de retrouver toutes les instances d'une classe donnée. Des requêtes plus complexes peuvent être composées en utilisant le constructeur *select-from-where* de RQL dont la clause *from* admet des expressions de chemins de graphe RDF.

Le module de requête RQL de Sesame permet d'interpréter une requête RQL, de construire un modèle optimisé d'arbre pour une requête, et d'évaluer ce modèle d'arbre par un ensemble d'accès au SGBD via l'API SAIL, ces accès étant ordonnés selon la structure d'arbre dans laquelle la requête a été décomposée.

3.3 JENA et RDQL

RDQL est le module de requête de la plate-forme Jena d'outils pour RDF. Il est une implémentation du langage SquishQL (Miller *et al.*, 2002). SquishQL est un langage simple d'interrogation de données RDF basé sur des constructeurs similaires à ceux de SQL. Dans le modèle de SquishQL, un graphe RDF est représenté par l'ensemble des triplets qui le composent. Une requête est un ensemble de triplets dont les ressources, propriétés et littéraux peuvent être remplacés par une variable. La réponse à une requête est une paire composée d'un sous-ensemble de triplets de la base de connaissances qui correspondent (*match*) à ceux de la requête et d'une table d'ensembles de valeurs possibles pour les variables de la requête. La sémantique de RDFS n'est pas représentée dans SquishQL de sorte que les triplets d'une réponse à une requête sont ceux explicitement présent dans la base de connaissance; ceux qui pourraient être déduits de la sémantique de RDFS (par exemple par la sémantique de la propriété *subClassOf*) ne sont pas recherchés.

3.4 DAMLJessKB et OWLJessKB

DAMLJessKB (Kopena & Regli, 2003) et son successeur OWLJessKB sont des outils de raisonnement pour le Web Sémantique. Les deux systèmes intègrent le système de production Jess (Java Expert System Shell) pour représenter les inférences contenues dans la sémantique des langages du Web Sémantique. Leur base de connaissances est constituée (1) de faits qui sont la traduction des triplets RDF contenus dans un ensemble donne d'annotations et (2) de règles qui implémentent la sémantique des langages du Web Sémantique: celle de RDFS, XSD et DAML (dans DAMLJessKB) ou OWL-Lite (OWLJessKB).

DAMLJessKB et OWLJessKB contiennent en outre des modules dédiés au chargement et à la validation d'ontologies DAML ou OWL-Lite.

DAMLJessKB ou OWLJessKB permettent de gérer une base de connaissances, de l'interroger et d'y appliquer des règles spécifiques à un domaine d'application particulier. Plus précisément, les deux systèmes implémentent deux types de raisonnement des Logiques de Description: le raisonnement sur les instances de classes et le raisonnement terminologique sur les classes elles-mêmes. Le raisonnement sur les instances de classes consiste essentiellement en l'implication, à partir de l'appartenance d'une instance à une classe, de l'appartenance de cette instance à toutes les classes ancêtres de cette classe dans l'ontologie. Le raisonnement sur les classes consiste en des inférences sur les relations de subsomption qui existent entre les classes.

4 Outils dédiés au e-Learning

Un acteur important dans le domaine de la recherche sur le e-Learning, est le groupe KMI (KMI, 2005) (Knowledge Media Institute) qui profite de l'expérience glanée au sein d'Open University (OU, 2005), organisme qui propose tout un panel de formations à distance. Dans le cadre de leur recherche un certain nombre d'outils ont été développés. Ils préfigurent certainement les futurs outils de partage et de communication indispensables au e-Learning. Nous présentons dans cette partie un survol de ces outils. Certains de ces outils sont utilisables gratuitement et parfois distribués en open source.

4.1 WebOnto

WebOnto est un service disponible sur le web qui permet de visualiser et d'éditer des modèles de connaissances (ontologies). Le langage utilisé pour exprimer les modèles est OCML (Operational Conceptual Modelling Language), un langage spécialement développé au sein de KMI, pour la construction de modèles de connaissances (Motta *et al.*, 2000). Le système est basé sur un serveur accessible au travers d'une applet Java (webOnto, 2005).

4.2 D3E

D3E est un outil permettant la discussion autour de documents. A partir d'un document HTML, l'outil permet de naviguer dans la structure du document et intègre un espace de discussion basé sur un forum : HyperNews.

Cet outil peut être utilisé par une équipe d'enseignants disséminés géographiquement et voulant produire un document de cours commun. Pour l'instant il a surtout été développé pour les discussions autour de la rédaction du journal électronique « Journal of interactive Media in Education » (Buckingham Shum & Sunner, 2001). D3E est distribué en open source (D3E, 2005).

4.3 Lyceum

Lyceum est un outil qui permet de réunir des étudiants et leur professeur autour d'une conférence vocale, à partir d'un accès internet par modem. Une interface permet d'interagir en même temps que l'on utilise le signal vocal (chat, vote, demande de parole). La voix peut passer dans les deux sens et permet de nombreuses interactions. L'interface permet d'afficher d'autres informations relatives au cours (figures par ex.). Cet outil (Lyceum, 2005) a été déployé pour la première fois au sein d'Open University fin 99 (Buckingham Shum *et al.*, 2001).

4.4 BuddySpace

BuddySpace est un outil de messagerie instantanée, écrit en Java et offrant l'opportunité de visualiser la position des correspondants sur une carte. Il utilise le protocole Jabber ce qui lui permet de dialoguer avec les systèmes de messagerie instantanée comme ICQ, MSN, Yahoo, etc... BuddySpace permet de définir précisément la notion de présence (avec plus de niveaux et de possibilités que les programmes classiques). L'apport principal de cet outil par rapport aux systèmes de messagerie instantanée classiques est de renforcer la sensation de « présence » des intervenant et ainsi de favoriser leur implication dans le processus pédagogique (Eisenstadt *et al.*, 2003). BuddySpace est distribué en open source (BuddySpace, 2005).

4.5 Claimaker

Claimaker est un système qui permet de modéliser et de visualiser des assertions et discussions faites par rapport à des documents de recherche, typiquement des articles scientifiques (Claimaker, 2005). Pour un document l'outil permet de visualiser le réseau d'idées à mesure qu'il est complété par les différents intervenants. Les assertions (claims) sont des triplets de la forme « concept – lien – concept », les liens font partie d'une ontologie du discours et les concepts se rapportent au document (Uren *et al.*, 2003).

Cet outil permet de rassembler des communautés d'idées, de formaliser les discussions, et de pouvoir faire le point sur un domaine, ou un cours par exemple.

4.6 eLPA

Dans (Kokkaros & Vlahava, 2002) est proposé le système eLPA, offrant un agent intelligent qui code sous forme de graphes conceptuels (GC) les métadonnées des « learning objects » auxquelles accède l'utilisateur, les stocke dans un répertoire personnel GC-LOM, puis utilise des connaissances du domaine exprimées sous forme de règles de graphes conceptuels, pour jouer auprès de l'utilisateur le rôle d'un assistant lui offrant une mémoire éducative personnalisée. Cet agent est implémenté au-dessus de Cogitant.

5 Conclusion

Nous avons présenté dans ce rapport des outils dédiés au Web sémantique et des outils de e-Learning. Des utilisations possibles des outils du Web sémantique sont la construction ou l'exploration d'ontologies dédiées au E-learning, l'annotation sémantique des ressources pédagogiques basée sur les ontologies construites, la recherche sémantique de ressources pédagogiques particulières (voire d'informations à l'intérieur de ces ressources), guidée par les ontologies construites, et mettant en oeuvre des mécanismes de raisonnement sur les annotations sémantiques des ressources et les ontologies.

D'autre part, les outils dédiés au E-learning proposant des espaces de discussion et supportant la collaboration entre enseignants et apprenants pourraient être adaptés pour prendre en compte des ontologies ou des annotations.

Références

- ARPIREZ J.C., CORCHO O., FERNANDEZ-LOPEZ M. AND GOMEZ-PEREZ A. (2003). WebODE in a Nutshell, *AI Magazine*, Vol 24(3).
- BROEKSTRA J., KAMPMAN A., VAN HARMELEN F. (2002). Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In Proceedings of the *International Semantic Web Conference (ISWC2002)*, LNCS 2342, pp. 54-68.
- BUCKINGHAM SHUM S. AND SUMNER T. (2001). JIME: An Interactive Journal for Interactive Media, *First Monday*, Vol 6 (2).
- BUCKINGHAM SHUM S., MARSHALL S., BRIER J. AND EVANS T. (2001). Lyceum: Internet Voice Groupware for Distance Learning. In Proceedings of *Euro-CSCL 2001: 1st European Conference on Computer-Supported Collaborative Learning*, Maastricht, The Netherlands.
- CAUSSANEL J., ZACKLAD M. AND CAHIER J.-P. (2002). Proposition d'un méta-modèle basé sur les Topic Map pour la structuration et la recherche d'information. *Journées scientifiques Web sémantique – CNRE-AS2W*, Paris.
- CORBY O., DIENG R., HEBERT C. (2000). A Conceptual Graph Model for W3C Resource Description Framework. *Proc. of the International Conference on Conceptual Structures (ICCS2000)*, Springer-Verlag LNCS 1867, p. 468-482, Darmstadt, Germany.
- CORBY O., DIENG-KUNTZ R. & FARON-ZUCKER C. (2004), Querying the Semantic Web with the Corese Search Engine, in Proceedings of the *European Conference on Artificial Intelligence (ECAI'2004)*, pp. 705-709, Valencia, Spain, 2004.
- CORCHO O., FERNANDEZ-LOPEZ M., GOMEZ-PEREZ A., VICENTE O. (2002). WebODE: An Integrated Workbench for Ontology Representation, Reasoning and Exchange. In Proceedings of the *International Conference on Knowledge Engineering and Knowledge Management (EKAW2002)*, LNAI 2473, pp. 138-153.
- EISENSTADT M., KOMZAK J. AND DZBOR M. (2003). Instant messaging + maps = powerful collaboration tools for distance learning. In Proceedings of *TelEduc03*, Havana, Cuba.
- HANDSCHUH S., STAAB S. AND MAEDCHE A. (2001). CREAM - Creating relational metadata with a component-based, ontology-driven annotation framework. In Proceedings of the *International Conference on Knowledge Capture (K-CAP 2001)*, pp. 76-83.
- KOKKORAS F., & VLAHAVAS I. (2002) "eLPA: An e-Learner's Personal Assistant", Proc. ICCS'02 Workshop on Applications of Conceptual Graphs, (electronic proceedings), Borovets, Bulgaria, 2002.

- KOPENA J.B., REGLI W.C. (2003). DAMLJessKB: A Tool for Reasoning with the Semantic Web. In Proceedings of the *International Semantic Web Conference (ISWC2003)*, LNCS 2870, pp. 628-643.
- KUNZE T., BRASE J. AND NEJDL W. (2002). Editing Learning Object Metadata: Schema Driven Input of RDF Metadata with the OLR3-Editor. In Proceedings of the *SAAKM Workshop at the European Conference on Artificial Intelligence*, Lyon, France.
- LAURENT D. (2002). De la création à la capitalisation des annotations dans un espace personnel d'informations. Thèse de doctorat de l'Université de Savoie.
- MILLER L., SEABORNE A., REGGIORI A. (2002). Three Implementations of SquishQL, a Simple RDF Query Language. In Proceedings of the *International Semantic Web Conference (ISWC2002)*, LNCS 2342, pp. 423-435.
- MOTTA E., BUCKINGHAM SHUM S., DOMINGUE J. (2000). Ontology-Driven Document Enrichment: Principles, Tools and Applications. *International Journal of Human-Computer Studies*, Vol 52(6), pp. 1071-1109.
- NOY N. F., SINTEK M., DECKER S., CRUBEZY M., FERGERSON R. W., MUSEN M. A. (2001). Creating Semantic Web Contents with Protege-2000. In *IEEE Intelligent Systems*, Vol 16(2), pp. 60-71.
- POPOV B., KIRYAKOV A., KIRILOV A., MANOV D., OGNANOFF D. AND GORANOV M. (2003). KIM – Semantic Annotation Platform. In Proceedings of the *International Semantic Web Conference (ISWC 2003)*, LNCS 2870, pp. 834-849.
- UREN V., BUCKINGHAM SHUM S., LI G., DOMINGUE J., MOTTA E. (2003). Scholarly Publishing and Argument in Hyperspace. In Proceedings of the *International World Wide Web Conference (WWW 2003)*, Budapest, Hungary.
- VARGAS-VERA M., MOTTA E., DOMINGUE J., LANZONI M., STUTT A., CIRAVEGNA F. (2002). MnM: Ontology Driven Semi-automatic and Automatic Support for Semantic Markup. In Proceedings of the *International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002)*, LNCS 2473, pp. 379-391.
- ANNOTEAL PROJECT (2005). <http://www.w3.org/2001/Annotea/>
- BUDDYSACE (2005). <http://kmi.open.ac.uk/projects/buddyspace/>
- CLAIMAKER (2005). <http://claimaker.open.ac.uk/>
- CORESE (2005) <http://www-sop.inria.fr/acacia/soft/corese/>
- D3E (2005). <http://d3e.sourceforge.net/>
- KIM (2005). <http://www.ontotext.com/kim/>
- KNOWLEDGE Media Institute (2005). <http://kmi.open.ac.uk/>
- LYCEUM (2005). <http://kmi.open.ac.uk/projects/lyceum/>
- IMARKUP (2005). <http://www.imarkup.com/>
- MNM (2005). <http://kmi.open.ac.uk/projects/akt/MnM/>
- ONT-O-MAT (2005). <http://annotation.semanticweb.org/tools/ontomat>
- OPEN UNIVERSITY (2005). <http://www.open.ac.uk/>
- PROTEGE (2005) <http://protege.stanford.edu/index.html>
- WEBONTO (2005). <http://kmi.open.ac.uk/projects/webonto/>