Université de Montréal

Une approche d'ingénierie ontologique pour l'acquisition et l'exploitation des connaissances à partir de documents textuels :

Vers des objets de connaissances et d'apprentissage

Par

Amal Zouaq

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures

en vue de l'obtention du grade de Philosophia Doctor (Ph.D.)

en informatique

Décembre 2007

Université de Montréal
Faculté des études supérieures

Cette thèse intitulée :

Une approche d'ingénierie ontologique pour l'acquisition et l'exploitation des connaissances à partir de documents textuels : Vers des objets de connaissances et d'apprentissage

Présentée par :
Amal Zouaq

A été évaluée par un jury composé des personnes suivantes :

Guy Lapalme, président-rapporteur
Claude Frasson, directeur de recherche
Roger Nkambou, co-directeur de recherche
Julie Vachon, membre du jury
Monique Grandbastien, examinateur externe
Jian-Yun Nie, représentant du doyen de la FES

Résumé

Les systèmes à base de connaissances doivent faire face à un problème d'acquisition des connaissances bien connu en intelligence artificielle. Notre approche vise à permettre une acquisition automatique de connaissances à partir de textes par la génération (semi) automatique d'une ontologie du domaine. Des techniques statistiques et des techniques du traitement du langage naturel sont utilisées à cet effet. Une méthodologie pour l'évaluation de la qualité de l'ontologie est également proposée.

Notre objectif ne vise pas seulement l'acquisition des connaissances mais également leur exploitation dans un contexte de formation par ordinateur. Deux communautés s'intéressent à un tel objectif : celle des systèmes tutoriels intelligents et celles du e-Learning. Bien qu'ayant des objectifs communs (la formation), les systèmes tutoriels intelligents et les systèmes traditionnels de e-Learning utilisent des moyens différents pour y parvenir. Nous pensons qu'une synergie de leurs points forts permettra de créer un pont entre les deux communautés, via un modèle de connaissance partagé : l'ontologie du domaine. Nous présentons une architecture ontologique qui se base sur ce modèle du domaine, ainsi que sur d'autres ontologies, pour générer dynamiquement des objets de connaissances et d'apprentissage ou LKO (*Learning Knowledge Objects*).

Les LKO se distinguent par un contenu basé sur une structure sémantique explicite et par leur capacité d'adaptation à un apprenant. Ils sont composés dynamiquement et au besoin, selon une compétence à atteindre et guidés par des théories pédagogiques explicites. Ces caractéristiques rendent possible l'exploitation des LKOs par des systèmes tutoriels intelligents. Nous proposons également des mécanismes de standardisation des LKOs afin de permettre aux environnements standards de formation en ligne d'en bénéficier.

Mots-clés : Acquisition des connaissances, formation, ontologies, mémoires, systèmes tutoriels intelligents, e-Learning, forage de données, traitement de la langue naturelle

Abstract

The Knowledge Acquisition Bottleneck is one of the oldest problems of knowledge-based systems in general and of computer-based training in particular. The aim of the this thesis is mainly to tackle the issue of knowledge acquisition and exploitation for computer training purposes by offering an integrated framework that enables knowledge extraction from texts and its dissemination for training purposes.

Knowledge acquisition is performed over plain text documents and aims at learning and populating a domain ontology from texts using statistics, machine learning and natural language processing. The learning methodology proposed here can be used for any domain and any application. However, one interesting aspect is that it provides intermediate structures in the form of concept maps that can be useful for training.

In fact, the second main objective of this thesis is to provide a bridge between the different computer-based training communities and especially the intelligent tutoring system community and the e-Learning community. Through a common knowledge base, and a common ontological framework, our system "The Knowledge Puzzle" is able to automatically compose new learning resources dubbed Learning Knowledge Objects (LKOs). These LKOs can be exploited by any training system and have some interesting characteristics: they are active, theory-aware, and content-aware and they have a set of services to act on their knowledge base. We also provide standardization mechanisms to enable the LKOs to be launched in standard e-Learning environments such as SCORM and IMS-LD.

Keywords: Knowledge Acquisition, ontologies, organizational memories, Intelligent Tutoring Systems, e-Learning, data mining, natural language processing.

Table des matières

Liste des tableaux

Liste des figures

A mes parents

Remerciements

Je tiens à remercier un ensemble de personnes qui m'ont accompagnée tout au long de cette thèse et qui m'ont permis de la réaliser.

Tout d'abord, M. Roger Nkambou pour son support inconditionnel, la qualité de son encadrement, sa disponibilité et son amitié. Merci également à ma famille qui a supporté les bons et moins bons moments durant cette thèse.

Un immense merci aux membres du jury qui m'ont prodigué des conseils avisés qui m'ont permis de grandement améliorer mon manuscrit. Tout particulièrement, j'aimerais remercier Mme Monique Grandbastien pour son rapport détaillé et extrêmement riche, ainsi que pour le temps qu'elle y a consacré. Enfin, mes remerciements vont également à M. Guy Lapalme (président-rapporteur) et Mme Julie Vachon (membre du jury) pour leurs recommandations.

1 Introduction

La difficulté de l'acquisition des connaissances est une problématique récurrente pour la communauté en intelligence artificielle (IA). Cela est également le cas pour la communauté des Environnements Informatiques pour l'Apprentissage Humain (EIAH). Par EIAH, nous entendons aussi bien les EIAH dits intelligents, mieux connus sous le vocable de systèmes tutoriels intelligents (STI), que les systèmes issus de la formation en ligne (e-Learning). Les STI nécessitent une modélisation de la connaissance du domaine, de la connaissance de l'apprenant et de l'expertise pédagogique. C'est entre autres, en raison de la difficulté de l'acquisition de ces connaissances, que ces derniers ont échoué à s'imposer dans la pratique. A l'opposé, le e-Learning a connu un véritable essor dû à la relative facilité de production et de déploiement des objets d'apprentissage, sortes de contenus agrégés de ressources d'apprentissage (pages web, images, etc.) ne nécessitant pas une réelle représentation de leur contenu.

Considéré à l'origine comme un avantage (rapidité et facilité de production de ressources ne nécessitant pas de réelle modélisation du domaine), ce manque de représentation des connaissances constitue une des limites du modèle e-Learning (Jovanovic, Gasevic, & Devedzic, 2006a) (Stojanovic, Staab, & Studer, 2001) (Ullrich, 2005) (Devedžić, 2004) (Zouaq, Nkambou, & Frasson, 2007a): les objets d'apprentissage demeurent des sortes de boites noires difficilement exploitables par des logiciels (agents, moteurs de recherche, STI, systèmes de e-Learning). Une meilleure représentation devrait permettre une recherche plus ciblée du contenu des objets d'apprentissage et donc une meilleure exploitation (pour la composition automatique de ressources, pour la recherche de portions de ressources à même de répondre à un besoin précis, etc.). Elle devrait également offrir des structures de connaissances enrichissantes pour les apprenants et utilisables pour la mise en place de services tutoriels (exploration des connaissances, service d'explication, etc.).

Dans ce contexte, il est nécessaire de trouver des modèles adéquats permettant la représentation des connaissances dans la formation en ligne. Par ailleurs, bien que plusieurs efforts aient été déployés pour la création de systèmes auteurs manuels pour les STI, ces

initiatives n'ont pas permis de faciliter suffisamment l'acquisition de la connaissance du domaine. Il importe donc également de doter les systèmes tutoriels intelligents de mécanismes d'extraction ou de création (semi) automatiques des connaissances qui leur sont nécessaires, notamment en ce qui concerne la connaissance du domaine. A terme, cette démarche vise à fédérer les communautés issus du e-Learning et des STI de manière à leur fournir des modèles communs, en dotant les systèmes de e-Learning de représentations explicites et sémantiques et en facilitant la création des ressources d'apprentissage utilisables par les deux communautés.

1.1 Problématiques et objectifs

Cette thèse vise donc à aborder diverses problématiques relatives aux EIAH. Elle vise à alléger la tâche des experts humains par la mise en place de mécanismes d'acquisition semi-automatiques des connaissances du domaine. Elle a également pour objectif de créer des ressources d'apprentissage disposant de représentations plus fines des connaissances (du domaine, de la pédagogie) au travers de la mise en place d'une architecture intégrée pour les communautés e-Learning et STI, de la proposition d'un nouveau modèle pour les objets d'apprentissage et enfin de la mise en place de services tutoriels communs pour les différents types de plateformes.

1.1.1 Acquisition semi-automatique des connaissances du domaine

L'acquisition d'un modèle de connaissances du domaine repose généralement sur des experts humains et sur un processus d'explicitation de leurs connaissances. Outre le fait que cette pratique est ardue, elle nécessite de recommencer l'explicitation pour chaque domaine et est difficile à mettre à jour de manière à refléter les évolutions du domaine. Il serait donc souhaitable de mettre en place un processus d'acquisition (semi) automatique à partir des sources de connaissances du domaine.

Les documents textuels représentent une source de connaissances potentielle pour ce processus. La multiplication des écrits et leur foisonnement sur le web, dans les

communautés de pratique et dans les universités, etc. ainsi que le nombre croissant d'objets d'apprentissage nécessitent de les prendre en compte comme matériaux premiers des connaissances explicites et implicites exprimées dans un domaine (Uren, et al., 2006) (Hammouda & Kamel, 2005) (LT4eL, 2008).

Afin de comprendre comment cela peut être fait, il importe de décrire l'interaction de deux disciplines : l'intelligence artificielle et la linguistique, et de revenir sur leurs historiques et héritages respectifs. Cette petite rétrospective est principalement tirée du passionnant article de Bernard (Bernard, 2000) :

La compréhension d'un texte par un ordinateur nécessite des modèles de compréhension de la langue, issus généralement de la linguistique. L'intelligence artificielle se base sur ces modèles pour représenter « adéquatement » la sémantique d'un texte. Ce concept de représentation sémantique est issu d'une longue tradition en IA, plus précisément depuis les travaux de McCarthy, de Minsky, de Newell et Simon. On assiste ainsi à la création des réseaux sémantiques (Quillian, 1968), des primitives de représentation (Schank, 1972), des schémas (Minsky, 1975), de nouveaux langages de représentation des connaissances (Brachman & Schmolze, 1985), des scénarios (Schank & Abelson, 1977) et des graphes conceptuels (Sowa, 1984).

Le passage d'un texte à une représentation utilisable et interprétable par des programmes d'IA s'effectue au moyen d'un analyseur (*parser*) de la langue naturelle. Cette représentation se traduit sous forme d'une analyse syntaxique au travers des grammaires de constituants. Un autre formalisme, inspiré des travaux de Tesnière (Tesnière, 1959), voit ensuite le jour : les grammaires de dépendances qui représentent les liens de dominance syntaxique-sémantique. Beaucoup de travaux ont ensuite exploité les grammaires de dépendances dans l'extraction de connaissances. Toutefois, une grande partie de ces projets s'est appuyée sur des primitives sémantiques typées et préalablement définies. Le problème est que le choix de ces primitives reste arbitraire et ne peut représenter un langage dans son ensemble, mais seulement des mondes ou domaines réduits. Trouver un moyen

d'extraire des concepts et des relations sémantiques à partir de textes sans toutefois recourir à des primitives sémantiques préalablement typées est donc, à notre sens, primordial. Cela est d'autant plus vrai dans le domaine de la formation par ordinateur qui vise de multiples domaines de connaissances.

Par conséquent, et en raison de son application à la formation, l'acquisition des connaissances du domaine ne doit pas se baser sur un monde fermé ou sur une sémantique prédéfinie mais doit être applicable à de nombreux domaines. Elle doit également permettre de modéliser le contenu des objets d'apprentissage et de préserver la structure de ce contenu. Notre premier objectif consiste donc à proposer une méthodologie et un outil à même de remplir ces conditions à travers la mise en place d'une nouvelle méthodologie pour acquérir un modèle du domaine de manière (semi) automatique. Cette méthodologie permet de générer des cartes de concepts à partir de textes et de les transformer ensuite en ontologie du domaine.

1.1.2 Mise en place d'une architecture intégrée pour les EIAH

En sus de la création d'un modèle du domaine, cette thèse vise la création d'une architecture intégrée pour les EIAH permettant de produire des ressources d'apprentissage qui soient exploitables par la multitude de plateformes disponibles pour l'apprentissage : les systèmes tutoriels intelligents (STI), les systèmes hypermédia adaptatifs et les systèmes de *e-Learning* standards (ce qui n'est pas le cas actuellement). Autrement dit, l'idée est de créer une base de connaissance unifiée qui doit non seulement bénéficier aux différentes communautés des EIAH, mais également offrir des alternatives aux technologies et méthodologies actuelles. Cette base de connaissance s'appuie sur des ontologies pour représenter les différents modèles d'un système tutoriel intelligent, ce qui a pour intérêt de les rendre partageables, interopérables et réutilisables, des caractéristiques jusque là faisant cruellement défaut aux STI. Pour le e-Learning, elle offre une alternative aux entrepôts d'objets d'apprentissage statiques et permet d'envisager les objets d'apprentissage sous une nouvelle forme. C'est ce qui est développé dans ce qui suit.

1.1.3 Proposition d'un nouveau modèle pour les objets d'apprentissage

La formation en ligne s'appuie complètement sur la notion d'objet d'apprentissage, popularisée par Wiley (Wiley, 2000). Un réexamen de ce qu'implique cette notion d'objet est à notre avis nécessaire : à quoi servent-ils et sont-ils capables de répondre aux défis de l'apprentissage dans le cadre du Web sémantique ?

Nous avons déjà évoqué l'aspect « boite noire » des objets d'apprentissage, qui ne disposent d'aucune représentation de leur contenu. A cette limite, il faut ajouter l'imbrication de la connaissance pédagogique et de la connaissance du domaine dans l'objet même. En effet, chaque concepteur crée une ressource d'apprentissage selon une expertise pédagogique qui lui est propre (Dehors, Faron-Zucker, Giboin, & Stromboni, 2005) (Ullrich, 2004). Cela pose deux sortes de problèmes : d'une part, cette expertise n'est pas accessible à un logiciel. D'autre part, il n'est pas possible de réutiliser la connaissance contenue dans cet objet d'apprentissage sans la pédagogie qui l'accompagne, limitant ainsi sa portée. Cela implique que les créations d'objets d'apprentissage restent tributaires de l'expert humain et ne sont pas à même de bénéficier de librairies encodant le savoir pédagogique (théories, stratégies, etc.).

Nous proposons donc une nouvelle vision des objets d'apprentissage qui permette de représenter explicitement leur contenu par une indexation utilisant l'ontologie du domaine préalablement générée. Nous proposons également de composer des objets d'apprentissage à la volée et au besoin, en choisissant au moment de la composition la théorie pédagogique adéquate. Cette composition doit également s'appuyer sur des portions granulaires d'objets d'apprentissage, à même de remplir un rôle pédagogique précis dans l'objet. Ces rôles pédagogiques doivent également pouvoir être réutilisés.

En les dotant de représentations du domaine et de représentations pédagogiques, ces objets d'apprentissage, nommés « *Learning Knowledge Objects* » disposent ainsi de certaines caractéristiques réservées jusque là aux STI, à savoir la représentation des

connaissances du domaine et des connaissances pédagogiques, l'adaptabilité (à des apprenants particuliers) et l'autonomie dans la mise en œuvre d'un enseignement.

1.1.4 Mise en place de services tutoriels

Enfin, la mise en place de services tutoriels nous semble incontournable dans le cadre d'une vision plus large d'objets d'apprentissage reposant non pas sur des objets statiques et entreposés, mais sur la base de connaissances préalablement introduite.

Différent services tutoriels sont possibles dans ce cadre. Nous avons déjà évoqué un service de **composition automatique** de LKOs à la demande. En effet, une telle composition représente une problématique importante du e-Learning. Cette composition doit non seulement s'adapter à un domaine de connaissance et à une stratégie pédagogique, mais elle doit également s'adapter à un profil d'apprenant particulier. Par ailleurs, les LKOs doivent être exploitables aussi bien par des STI que par des systèmes de e-Learning. L'architecture que nous proposons permet de soutenir un **service de déploiement** des objets d'apprentissage sur de multiples plateformes. Elle offre également un enrichissement des sessions d'apprentissage à travers **l'exploitation des structures de connaissances** établies notamment les cartes de concepts et l'ontologie du domaine. Enfin, un **service de standardisation** visant à garantir la portabilité des LKO sur des plateformes e-Learning standards est également proposé.

1.2 Organisation de la thèse

Pour résumer, l'objectif global de cette thèse est de présenter une architecture d'acquisition et d'exploitation des connaissances dans le domaine de la formation. Cette architecture a pour particularité d'extraire ses connaissances d'un corpus de documents textuels. Elle s'appuie sur des structures ontologiques afin de garantir des capacités de réutilisation, d'interopérabilité et d'inférence. Elle s'appuie également sur la notion de service visant ainsi à fédérer les différentes communautés des EIAH et à fournir une

solution à certaines problématiques auxquelles doit faire face l'apprentissage par ordinateur.

La thèse se scinde en deux parties distinctes (acquisition des connaissances et exploitation des connaissances), organisées en 7 chapitres :

Dans la première partie, le chapitre 2 brosse un état de l'art sur les techniques d'acquisition d'ontologies du domaine à partir de textes. Dans le chapitre 3, nous présentons un outil nommé « TEXCOMON » dont le but est de concrétiser la méthodologie en deux étapes visant tout d'abord à l'extraction de cartes de concepts et ensuite à la conversion de ces cartes de concepts en une ontologie du domaine. Le chapitre 4 est consacré à l'évaluation de la qualité de l'ontologie générée par la présentation d'une méthodologie d'évaluation originale à trois dimensions : une évaluation structurelle, sémantique et comparative. L'analyse structurelle s'inspire des métriques sur les graphes et considère les caractéristiques dites structurelles de l'ontologie. L'évaluation sémantique repose sur le jugement des experts. Enfin, l'analyse comparative est effectuée en comparant les résultats de notre outil TEXCOMON avec ceux générés par TEXT-TO-ONTO (Maedche & Staab, 2000c), un des outils les plus connus de ces dernières années.

La deuxième partie se consacre à la présentation de la vision globale du projet. Elle présente un état de l'art des EIAH et développe les diverses problématiques à traiter notamment dans le cadre du Web sémantique éducationnel (Chapitre 5). Une architecture globale d'acquisition et d'exploitation des connaissances dans le cadre d'un EIAH est ensuite présentée (chapitre 6). Concrétisée dans le système « *The Knowledge Puzzle* », cette architecture se présente comme une solution originale représentant un pont entre les différentes communautés STI et e-Learning et permettant d'envisager des réponses communes aux problématiques d'indexation, de personnalisation et de composition de ressources d'apprentissage. Ce chapitre présente l'architecture globale du projet, et explique des concepts importants tels que la mémoire organisationnelle, les ontologies qui constituent cette mémoire et les services tutoriels qui l'exploitent. La mémoire

organisationnelle est notre réponse aux limites des objets d'apprentissage textuels, expliquées tout au long de cette thèse. Elle sert également de base à la génération d'un nouveau type d'objets d'apprentissage, que nous appelons LKOs, dotés de caractéristiques jusque là réservées aux STI.

La thèse se termine par une discussion sur ses contributions et sur les travaux futurs envisagés.

2 Acquisition d'une ontologie du domaine à partir de textes

En intelligence artificielle (IA), l'acquisition des connaissances est le processus qui permet de produire une information formalisée, qui de ce fait, pourra être traitée par un programme logiciel. Avec l'augmentation des informations disponibles dans les réseaux d'organisations, dans les universités et sur le web, il devient crucial de trouver un moyen de transformer ces informations en connaissances utilisables par des machines, c'est-à-dire en entités structurées, sémantiquement annotées et de ce fait, facilement localisables, réutilisables et partageables. Les sources de connaissances sont bien souvent des humains (on parle d'explicitation des connaissances des experts), des documents semi-structurés, ou des bases de données.

Dans cette thèse, nous nous intéressons à une autre source importante de connaissances : les documents textuels non structurés (exprimés en langage naturel). Étant donné qu'une bonne partie des connaissances est véhiculée par de tels documents, cette approche est certainement une des plus prometteuses, même si elle doit faire face aux difficultés inhérentes au traitement de la langue naturelle. Cette approche peut être clairement distinguée de celles où les experts doivent exprimer leurs connaissances selon un langage logique artificiel ou sous une certaine forme (interviews, analyse de protocoles), ce qui présente l'avantage d'un traitement plus aisé par les machines ou les programmes d'IA, mais qui contraint les experts à un exercice intellectuel difficile. Il n'est pas certain que la connaissance puisse être transmise via cette forme imposée. Un tel processus souffre également d'autres limites telles que le temps que prend cette acquisition de connaissances, l'habileté d'un expert à exprimer ses connaissances selon la forme imposée, la volonté de l'expert de les exprimer, ou encore l'interprétation donnée aux propos de l'expert par les ingénieurs de connaissances (Potter, 2001).

On le voit, un processus d'acquisition de connaissances formalisées n'est pas exempt de problèmes. Il importe donc de trouver des moyens d'expliciter la connaissance en utilisant ce qui la véhicule de manière naturelle et quotidienne : les documents écrits. Cette explicitation doit se doter de représentations de connaissances adéquates, et de techniques d'extraction de connaissances à partir de textes. De telles représentations peuvent être concrétisées par des ontologies.

2.1 Définition de la notion d'ontologie

Diverses définitions de la notion d'ontologie existent et diverses disciplines s'y intéressent, notamment la philosophie et l'informatique. Le web sémantique est une des applications informatiques modernes de la notion d'ontologie. Dans ce cadre, l'ontologie désigne un système conceptuel qui définit la sémantique d'un contenu digital. En fait, la notion d'ontologie existait, en informatique, bien avant l'avènement du Web sémantique mais ce dernier l'a en quelque sorte popularisée.

Les définitions informatiques relatives aux ontologies foisonnent. Pour Tim Berners-Lee «*Ontology is a document that formally defines the relations among terms. The most typical kind of ontology for the Web has taxonomy and a set of inference rules*" (Berners-Lee, Hendler, & Lassila, 2001). Ici, une ontologie réfère donc à une formalisation des relations entre termes dans un document et à l'utilisation de règles d'inférences pour raisonner sur ces termes et relations. Depuis cette définition, la notion d'ontologie a évolué vers un système conceptuel à part entière, servant à définir la sémantique de métadonnées. Selon Mizoguchi, une ontologie « *définit la signification des métadonnées ... et est utilisée principalement pour réaliser l'interopérabilité sémantique entre les ressources informationnelles grâce aux métadonnées* » (Mizoguchi, 2004). De manière peut-être plus concrète, une ontologie est constituée d'un ensemble de classes, d'instances, de relations et d'attributs. C'est un système conceptuel, qui, exprimé via un langage du Web sémantique, permet de ce fait d'être réutilisable, partageable et permet d'effectuer des inférences.

Enfin, l'un des aspects importants d'une ontologie est qu'elle s'appuie sur la notion de consensus. Une ontologie n'est réellement utile que dans la mesure où elle est réutilisée par une communauté et dans la mesure où elle résulte d'un consensus dans cette communauté.

Il existe différents types d'ontologies : les ontologies de haut niveau ou génériques, les ontologies du domaine, les ontologies de tâches et les ontologies d'applications. Dans cette thèse, nous nous intéressons en particulier aux ontologies du domaine. Ces ontologies permettent de conceptualiser un domaine de connaissance donné. Ceci dit, la question importante est maintenant de se demander : comment construire une telle ontologie ?

2.2 Postulats de départs

La construction d'une ontologie est une démarche difficile et coûteuse. Comme tous les systèmes à base de connaissances, elle se heurte à la difficulté d'acquérir la connaissance, le fameux « *Knowledge Acquisition Bottleneck* » bien connu de la communauté en Intelligence Artificielle. Cette construction s'accompagne de nombreuses questions fondamentales notamment comment identifier et définir les concepts du domaine ?

En effet, en plus des obstacles techniques inhérents à la difficulté de l'extraction des connaissances à partir de textes, la construction ontologique se heurte à un conflit philosophique bien ancien entre le signifiant et le signifié : si un concept est une construction de l'esprit, existe-t-il une désignation spécifique à cette construction ? En d'autres termes, peut-il y avoir un couplage direct entre un terme et un concept ?

2.3 Lien entre terminologie et concept

Le langage étant le véhicule de la pensée, il est très difficile, lorsque l'on en vient à concevoir un système conceptuel de le dissocier de ses dénominations. C'est pourtant là que réside toute la différence entre la notion de terme et la notion de concept. Selon

Mizoguchi (Mizoguchi, 2004), bien que les deux soient dotés d'une étiquette (label), la différence fondamentale entre un terme et un concept est que pour le terme, cette étiquette est essentielle alors pour le concept, elle ne l'est absolument pas : « Un concept est indépendant de la façon dont on le nomme ». Cette affirmation est d'ailleurs concrétisée par des méthodes d'extraction statistiques (que nous introduisons en section 2.5.1) dans leur définition des concepts, puisque ceux-ci peuvent émerger d'un algorithme de catégorisation par exemple sans toutefois être dotés d'un nom particulier.

Le domaine de la sémantique lexicale aborde toutefois le problème avec une optique différente : seuls les termes dénotant des concepts sont considérés. Des réflexions très intéressantes sont présentées dans (Roche, 2006a) (Roche, 2006b) à propos de ce lien entre terminologie et concept. Pour Roche, la terminologie, en tant que discipline scientifique, ne s'intéresse aux termes qu'en tant que vecteurs de concepts : « *La terminologie, en tant que discipline scientifique, ne s'intéresse donc aux mots que dans la mesure où ils désignent des concepts (notions) permettant d'appréhender les objets du monde. Le terme, ou plus exactement, l'«unité terminologique» est une entité à double face, combinaison indissociable d'un concept et d'une « désignation », si l'on veut insister selon une sémantique référentielle sur les objets désignés par le terme, ou « dénomination » si l'on veut insister sur l'approche onomasiologique.* » (Roche, 2006b).

C'est dans ce lien entre termes et concepts, entre langage et pensée, qu'on en est venu à considérer la possibilité d'extraire des ontologies à partir de textes. Le postulat de base est qu'un ensemble de textes ou corpus véhiculent la connaissance d'un domaine donné. Bien que d'aucuns considèrent un tel corpus comme une sorte de monde fermé dont va pouvoir émerger une ontologie pertinente, nous rejoignons, quant-à nous, les chercheurs qui considèrent de tels corpus comme un moyen d'obtenir une sorte de squelette ontologique (Bourigault & Aussenac-Gilles, 2003) (Gargouri, Lefebvre, & Meunier, 2004), une construction de départ pouvant ensuite être complétée, validée et modifiée par un

humain. Roche a très bien nommé et décrit ce processus de rétro-ingénierie de la connaissance à partir de textes (Figure 1).

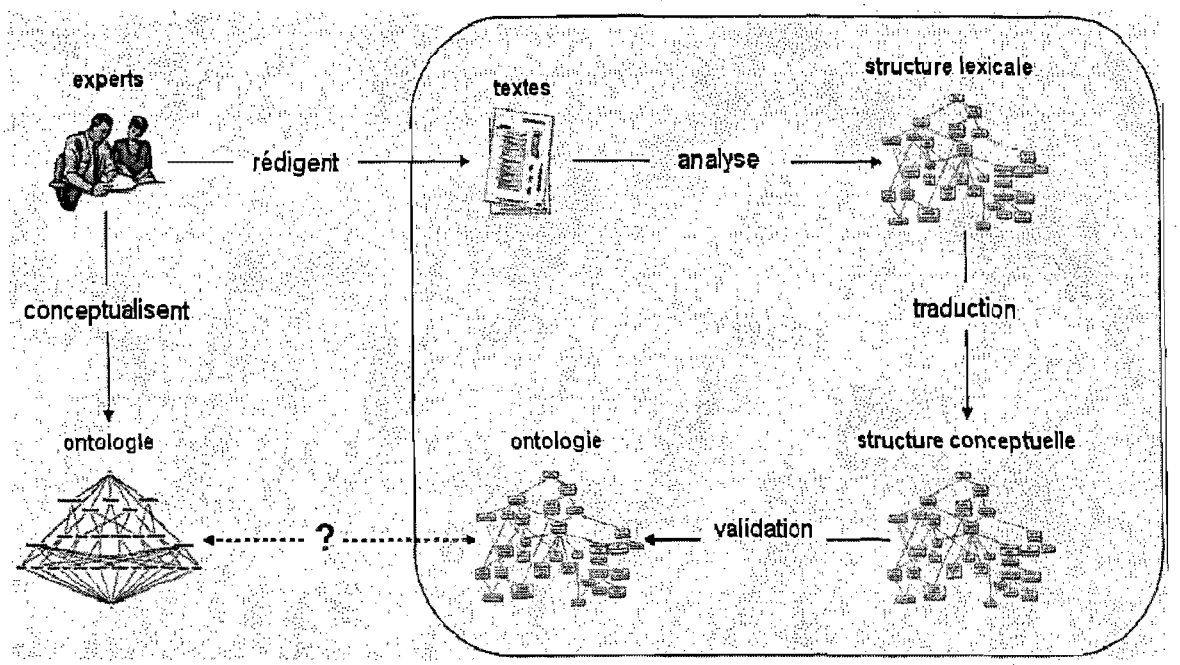


Figure 1. Rétro ingénierie des connaissances à partir des textes (Roche, 2006b)

Nous désignons le processus représenté ci-dessus par le vocable « apprentissage d'ontologie à partir de textes ». Dans la figure 1, l'analyse est appliquée sur des textes pour en déduire une structure lexicale. Toutefois, la démarche entreprise dans cette thèse ne s'appuie pas sur un processus d'analyse de tout le corpus. L'apprentissage d'ontologie est initié par la découverte de certains termes importants dans chaque document à analyser. Dans la prochaine section, nous abordons les algorithmes permettant la découverte de ces mots-clés.

2.4 Initier le processus d'apprentissage d'ontologie : les algorithmes de détection de mots-clés

Un processus d'apprentissage d'une ontologie à partir de textes commence donc généralement par l'obtention d'un ensemble de mots-clés représentant le domaine. Ces mots-clés servent d'éléments déclencheurs de l'apprentissage : ils sont recherchés dans le corpus, des liens sont créés avec d'autres termes au moyen de techniques linguistiques et/ou statistiques (catégorisation, cooccurrence, etc.). Ces nouveaux termes, une fois validés par un expert du domaine, servent ensuite à recommencer le processus. Dans le cadre de cette thèse, l'intérêt de commencer par l'extraction des mots-clés d'un document relève aussi d'une autre préoccupation : la nécessité de minimiser la taille du corpus à examiner, notamment dans le cadre d'une analyse linguistique coûteuse en temps et en ressources. Nous reviendrons sur ce point ultérieurement mais nous voulons souligner dès à présent que seules les phrases contenant les mots-clés seront utilisées dans une analyse subséquente.

Différents algorithmes pour l'extraction de mots-clés à partir de textes ont été proposés dans la littérature notamment l'algorithme KEA (Frank, Paynter, Witten, Gutwin, & Nevill-Manning, 1999) et l'algorithme GenEx (Turney, 2000). D'après les expérimentations menées (Turney, 2000), les deux algorithmes ont, statistiquement parlant, des performances équivalentes. Toutefois, KEA a en plus d'avoir été intégré dans l'outil de traitement de la langue naturelle GATE (*General Architecture for Text Engineering*) (Cunningham, Maynard, Bontcheva, & Tablan, 2002) et de continuer à être amélioré (Medelyan & Witten, 2006) (Witten, Paynter, Frank, Gutwin, & Nevill-Manning, 2005). C'est la raison pour laquelle nous avons opté pour KEA (Frank, Paynter, Witten, Gutwin, & Nevill-Manning, 1999).

KEA (Frank, Paynter, Witten, Gutwin, & Nevill-Manning, 1999) est un algorithme d'apprentissage machine supervisé pour la détection de mots-clés. Deux versions de

l'algorithme sont disponibles : KEA-3.0 et KEA-4.0. L'indexation de certains mots du texte comme mots-clés est effectuée de manière libre dans KEA-3.0 (sans aucune ressource pour guider l'indexation) alors qu'elle est guidée par un thésaurus relié au domaine dans KEA-4.0.

La figure suivante (Figure 2) présente le fonctionnement de la version KEA-4.0 contenant un thesaurus. Si on ignore la partie « Thesaurus », on peut aussi voir le fonctionnement de KEA-3.0 (expliqué par ailleurs ci-dessous).

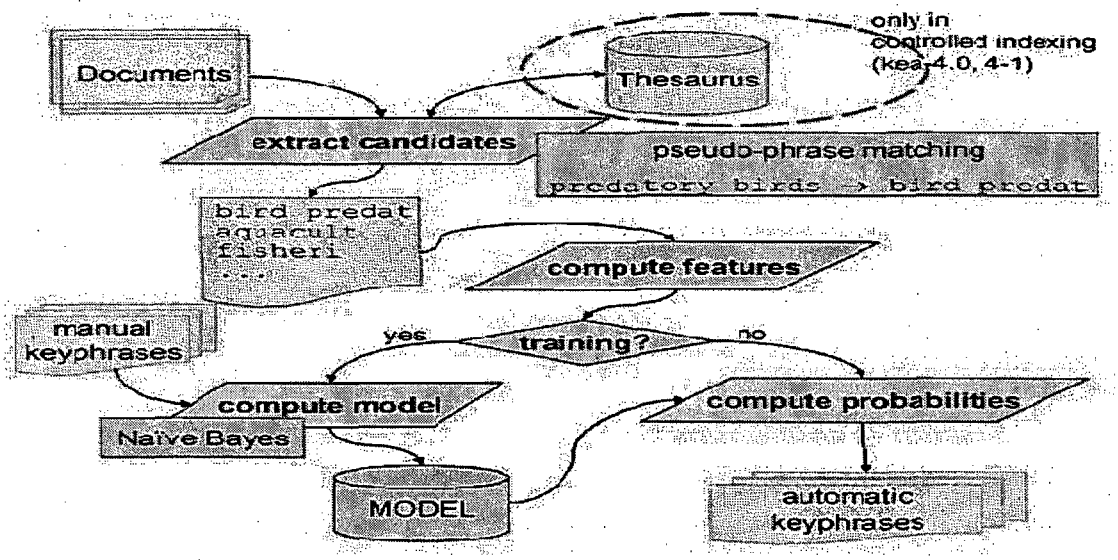


Figure 2. Fonctionnement de l'algorithme KEA 4.0 (KEA, 2007)

KEA 3.0 se base sur un modèle bayésien (*Naive Bayes*) extrait à partir d'un corpus exemple constitué de documents et de leurs mots-clés (fournis par des experts humains). Lors de la phase d'entraînement, l'algorithme considère que les mots-clés indiqués par l'expert humain sont des exemples positifs et il les repère dans le texte. Tout le reste du document est considéré comme un exemple négatif. Après analyse de métriques (présentées ci-dessous) calculées pour chaque mot-clé candidat, un modèle d'apprentissage est calculé. Ce modèle reflète la distribution des valeurs calculées pour chaque mot-clé.

Dans sa phase d'extraction, à partir d'un ensemble de documents de type « txt », l'algorithme KEA 3.0 se charge d'extraire des n-grams de longueur prédéfinie. Ces n-grams ne doivent pas contenir de mots outils (*stop words*) et représentent des mots-clés candidats. Des caractéristiques statistiques sont calculées pour chacun des mots-clés candidats :

- la *métrique TF*IDF* qui décrit la spécificité d'un terme pour un document donné par rapport à l'ensemble des documents du corpus. Un TF*IDF élevé indique une plus grande probabilité pour le candidat d'être un mot-clé représentatif.
- la *métrique de « première occurrence »* qui calcule le pourcentage du document qui précède la première occurrence du terme candidat recherché dans le document. Cette métrique découle du postulat que des termes qui apparaissent au début ou à la fin d'un document ont une plus grande probabilité d'être des mots-clés.
- la *fréquence du mot* dans l'ensemble des mots-clés détectés dans le corpus ayant servi à entraîner le modèle.

Lors de la phase d'extraction, les mots candidats dotés des plus hautes probabilités sont retenus comme mots-clés.

Ce qui différencie KEA 4.0 de la précédente version, c'est donc essentiellement l'utilisation d'un thésaurus du domaine et l'utilisation de métriques supplémentaires pour le choix des mots-clés. Selon (Medelyan & Witten, 2006), KEA-4.0 donne de meilleurs résultats que la version précédente. Néanmoins, ce n'est pas cette version que nous avons utilisée mais bien KEA-3.0 : disposer d'un thésaurus suppose que l'on connaît le domaine de connaissance dans lequel se situent les documents du corpus. Or notre objectif est justement de préserver notre indépendance du domaine et la possibilité de traiter des documents relatifs à plusieurs disciplines. KEA-3.0 a donc été choisi puisqu'il s'appuie essentiellement sur des caractéristiques statistiques non liées à un domaine particulier.

Suite à l'extraction de mots-clés, des techniques de fouilles de données permettent de transformer les structures lexicales ou linguistiques (voir figure 1) en structures conceptuelles. Ce sont ces techniques que nous abordons dans la section suivante.

2.5 Techniques d'extraction automatique d'ontologies du domaine

Les techniques d'acquisition de la connaissance à partir de texte se basent généralement sur des méthodes linguistiques, des méthodes statistiques ou sur une combinaison des deux. Largement utilisées dans l'extraction d'informations et dans la linguistique computationnelle, ces techniques sont généralement complémentaires dans leurs forces et faiblesses. Dans ce qui suit, nous présentons les deux grandes classes de méthodes pour le forage de textes, à savoir les méthodes statistiques et d'apprentissage machine et les méthodes linguistiques. Nous examinerons ensuite en détail les différents composants d'une ontologie (concepts, attributs, instances, rôle, etc.) et la manière de les extraire.

2.5.1 Les méthodes statistiques et l'apprentissage machine

Les techniques de traitement de la langue naturelle dites statistiques se basent sur des approches quantitatives. Globalement, ces techniques produisent des informations sur le nombre d'occurrences d'un terme, le nombre de cooccurrences de plusieurs termes, la fréquence d'apparition d'un terme dans un document ou un corpus. Elles permettent également de déterminer des "concepts" et relations statistiques.

Bien souvent, les statistiques obtenues sont utilisées comme paramètres ou entrées dans des algorithmes d'apprentissage machine pour détecter des connaissances jusque là non perceptibles. Étant donné que les techniques d'apprentissage machine ont été créées dans le but de capturer de la connaissance implicite, l'idée d'utiliser des algorithmes d'apprentissage machine pour l'extraction de connaissances à partir de textes semble

fondée et les méthodes d'apprentissage machine se sont d'ailleurs révélées importantes de par leurs résultats. Toutefois, leur mise en œuvre peut être ardue. En effet, les algorithmes d'apprentissage ne peuvent être appliqués sur des textes écrits directement et il n'est pas toujours simple de transformer ces textes en attributs tels que requis par les algorithmes d'apprentissage machine.

Différentes méthodes statistiques et d'apprentissage machine peuvent être utilisées pour l'extraction de connaissances (concepts et relations) à partir de textes : la métrique TF*IDF (Salton & McGill, 1983), le modèle vectoriel, les techniques découlant de l'hypothèse distributionnelle de Harris (Harris, 1968), l'analyse sémantique latente, ou encore les techniques de catégorisation et de classification.

Un des plus grands avantages de l'approche statistique est sa facilité de mise en œuvre et le peu de ressources nécessaires. L'approche statistique est indépendante du langage des textes sur lesquels elle s'applique. Sa seule exigence est la nécessité d'avoir un corpus documentaire de taille significative. Toutefois, la pertinence du traitement et surtout des résultats est très difficile à prévoir et à contrôler, surtout si on la compare à celle résultant de techniques linguistiques. Lorsqu'ils s'appuient sur des méthodes de classification, dont les résultats sont plus faciles à prévoir, les modèles de langage statistiques nécessitent souvent des données d'entraînement considérables afin d'être suffisamment efficaces et de représenter adéquatement la sémantique des phrases. Or ces données ne sont pas toujours disponibles et requièrent des efforts importants en termes d'annotation. Par ailleurs, la seule parade, lorsque les résultats ne sont pas satisfaisants, consiste à entraîner le système avec de nouveaux exemples.

Enfin, les méthodes statistiques occultent certaines connaissances lorsqu'elles ne sont pas statistiquement repérées. Elles ne font pas la différence, en ne tenant pas compte de la structure des phrases, entre différentes représentations utilisant les mêmes mots mais pas dans le même ordre. Par ailleurs, elles ne permettent pas de s'adapter à la spécificité

d'un domaine donné ou d'un contexte donné : tous les textes sont traités de manière similaire qu'il s'agisse de textes juridiques, médicaux, techniques ou scientifiques.

Les méthodes linguistiques permettent de pallier à certains de ces inconvénients en fournissant des résultats plus précis et surtout plus prévisibles.

2.5.2 Les méthodes linguistiques

Lorsque l'on parle de méthodes linguistiques, on se réfère à des techniques d'analyse de la langue naturelle, effectuée généralement en linguistique computationnelle. Certaines méthodes utilisent des techniques d'analyses peu approfondies comme la recherche d'expressions régulières, tandis que d'autres s'appuient sur des analyses plus complexes. Bien souvent, les méthodes linguistiques s'appuient sur des bases de connaissances pour mener à bien leur tâche. Ces bases de connaissances incluent des bases de patrons, des listes, des thésaurus ou glossaires ou encore des ontologies.

De manière générale, les méthodes linguistiques offrent des résultats plus probants ou pertinents que les méthodes statistiques : les résultats des algorithmes sont plus prévisibles. Lorsque cela est nécessaire, elles permettent également d'exploiter les connaissances d'un domaine donné de manière à personnaliser et raffiner le processus d'extraction. Elles se basent également sur des besoins qui sont clairement identifiés. Toutefois, les coûts de mise en œuvre de telles méthodes sont élevés car elles nécessitent des connaissances préalables telles que des thésaurus, des lexiques, des dictionnaires, des bases de patrons, etc. Par ailleurs, le fait de procéder à une analyse linguistique profonde peut être lourd en termes de temps de traitement, bien que cette dernière limite puisse être contournée avec la puissance de calculs des nouveaux ordinateurs et de ceux à venir. Enfin, généralement, les outils de traitement de la langue naturelle sont dépendants d'une langue et il est donc nécessaire d'avoir autant d'outils qu'il y a de langages dans les corpus à traiter.

Étant donné les avantages et inconvénients des méthodes linguistiques et statistiques, la majorité des projets d'analyse de textes tentent de trouver une combinaison appropriée de ces techniques. Ainsi, certains projets utilisent une analyse statistique sur les résultats d'une analyse linguistique, tandis que d'autres emploient les statistiques et l'apprentissage machine sur certaines tâches et la linguistique sur d'autres tâches. Par exemple, dans cette thèse, nous avons appliqué une mesure statistique et d'apprentissage machine pour retrouver les mots-clés d'un document et nous avons ensuite effectué une analyse linguistique des phrases clés retenues (Zouaq, Nkambou, & Frasson, 2007c).

2.5.3 Choix d'une grammaire de représentation

Ceci dit, une analyse linguistique suppose d'abord de faire le choix d'un outil de traitement de la langue naturelle qui soit le plus performant possible et qui nous permette d'obtenir des structures grammaticales correctes. Une analyse linguistique nécessite également d'opter pour une grammaire apte à représenter la structure des phrases.

Deux grammaires permettent essentiellement de représenter la structure d'une phrase en langage naturel (Covington, 2001) :

- **la grammaire de constituants** (*constituency grammar*) qui permet de diviser une phrase en constituants imbriqués. Cette grammaire est à la base de la théorie formelle du langage utilisée en linguistique computationnelle. L'idée fondamentale derrière la notion de constituants est qu'un ensemble de mots peuvent former une unité, comme une phrase nominale par exemple.
- **la grammaire de dépendances** (*dependency grammar*) qui crée des liens grammaticaux binaires entre les différents mots d'une phrase. Lorsque deux mots sont reliés par une relation de dépendance, on dit que l'un représente le gouvernant ou la tête (*head*) et l'autre le dépendant (*dependent*). En général, la tête guide le comportement de la relation et le dépendant représente un modifiant, un objet ou un complément. Entre les deux se trouve la relation, schématisée par un arc entre la tête et le dépendant. Un

graphe de dépendance est un arbre (un graphe direct acyclique DAG) dont la racine est le verbe principal de la phrase (Covington, 2001). La grammaire de dépendances permet également de retrouver les différents constituants de la grammaire précédente : chaque mot et ses dépendants (de manière imbriquée) représentent un constituant.

Lorsque l'on essaie d'extraire une représentation des connaissances d'un texte en langage naturel, la grammaire de dépendances et l'analyse selon ces dépendances présentent de nombreux avantages (Covington, 2001) :

- un analyseur par dépendances semble plus intuitif et plus semblable au fonctionnement d'un cerveau humain puisque les mots sont reliés par des relations grammaticales dès qu'ils sont rencontrés, sans aucune hypothèse préalable et sans attendre de compléter la phrase ;
- les liens de dépendances sont intuitivement plus proches des relations sémantiques que l'on veut obtenir dans une analyse sémantique subséquente ;

C'est essentiellement, ce dernier point qui nous a incités à opter pour la représentation par dépendances.

Il existe plusieurs analyseurs qui permettent d'effectuer une analyse par dépendances, parmi les plus importants, on peut citer *MINIPAR* (Lin, 1998), *Link Grammar* (Sleator & Temperley, 1993), ou encore *Stanford Parser* (Klein & Manning, 2003) (De Marneffe, MacCartney, & Manning, 2006).

Il importe d'avoir un analyseur le plus performant possible, c'est-à-dire le plus apte à fournir des liens de dépendance exacts. Les résultats présentés dans (Stevenson & Greenwood, 2006) suggèrent que l'analyseur de l'université Stanford (Klein & Manning, 2003) est capable de générer des analyses pour la majeure partie des phrases rencontrées. Par ailleurs, les travaux en traitement du langage naturel à l'Université Stanford (The Stanford NLP Group, 2007) sont à la fine pointe de ce qui se fait actuellement dans le domaine. L'analyseur de l'Université de Stanford est un analyseur probabiliste i.e. qu'il fait

appel à des modèles du langage. Issus des méthodes statistiques et d'apprentissage machine, ces modèles sont construits à partir de l'entraînement de l'analyseur sur des phrases préalablement annotées par un expert humain. De tels analyseurs sont généralement plus rapides et sont dotées d'une bonne performance. Il est aussi possible de les ré-entraîner sur d'autres corpus lorsque cela s'avère nécessaire pour améliorer leurs résultats.

La représentation en dépendances typées (*Typed Dependencies*) s'appuie sur une hiérarchie de relations grammaticales décrites dans (De Marneffe, MacCartney, & Manning, 2006) et représentée dans la figure 3. Cette hiérarchie contient 48 relations grammaticales et commence par la relation la plus générique qui est « *dep* » et qui indique simplement une dépendance entre le gouvernant et le dépendant. D'autres travaux ont déjà décrit des taxonomies de relations grammaticales similaires, notamment (Carrol, Minnen, & Briscoe, 1999), mais le travail de (De Marneffe, MacCartney, & Manning, 2006) les enrichit de nouvelles relations comme la détection des appositions « *appos* » (*appositive modifier*), les noms composés « nn » (*noun compound*), les relations numériques « num » (*numeric modifier*), les nombres « number » (*element of compound number*) et les abréviations « abbrev » (*abbreviation*). On peut retrouver une explication de chaque relation qui compose la hiérarchie en annexe C.

dep - dependent
  aux - auxiliary
    auxpass - passive auxiliary
    cop - copula
  conj - conjunct
  cc - coordination
  arg - argument
  subj - subject
    nsubj - nominal subject
      nsubjpass - passive nominal subject
    csubj - clausal subject
  comp - complement
    obj - object
      dobj - direct object
      iobj - indirect object
      pobj - object of preposition
    attr - attributive

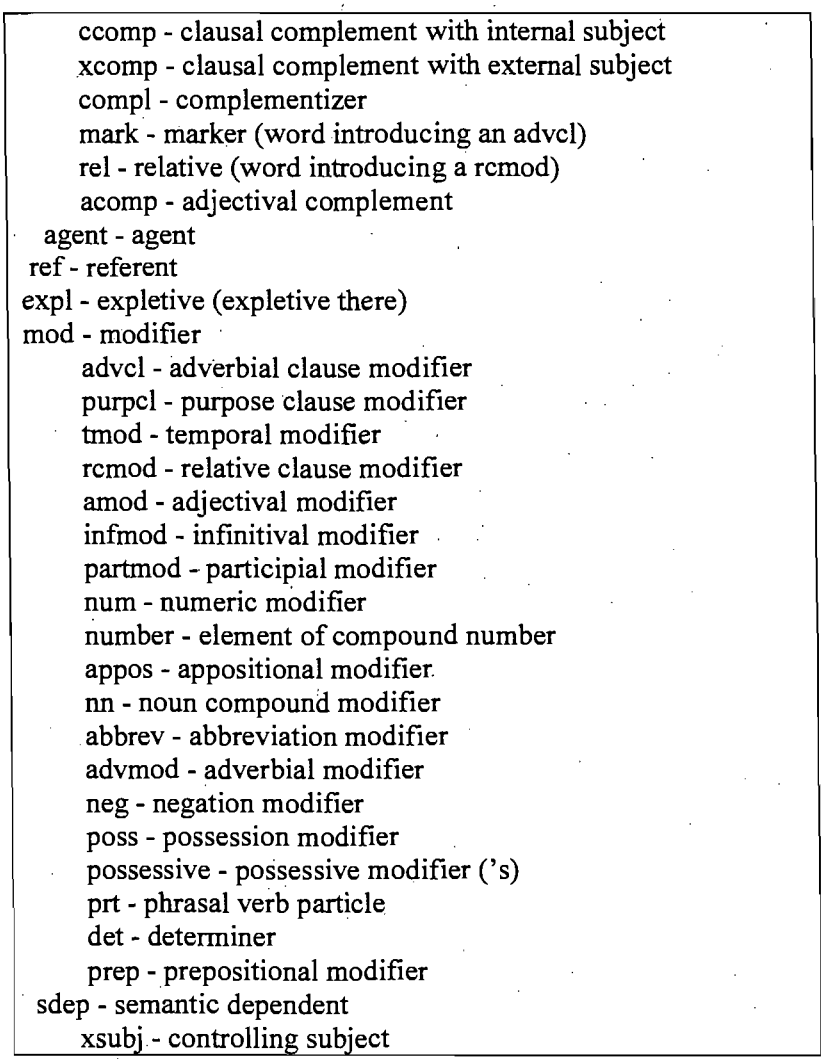


Figure 3. Hiérarchie des relations grammaticales (De Marneffe, MacCartney, & Manning, 2006).

Jusqu'à maintenant, nous avons justifié l'utilisation d'un outil (l'analyseur de Stanford) et d'un formalisme (la grammaire des dépendances) dans notre thèse. Nous avons également présenté un algorithme d'extraction de mots-clés comme moyen d'initier le processus d'apprentissage d'ontologies. La prochaine section explique le processus d'extraction d'ontologies à partir de textes en détail et effectue un état de l'art des différentes techniques utilisables à chaque étape du processus.

2.6 Décomposition du processus d'extraction d'ontologies à partir de textes

Le processus d'extraction d'une ontologie du domaine peut être décomposé en un ensemble d'étapes, résumées par (Buitelaar, Cimiano, Grobelnik, & Sintek, 2005). La figure suivante illustre ces étapes.

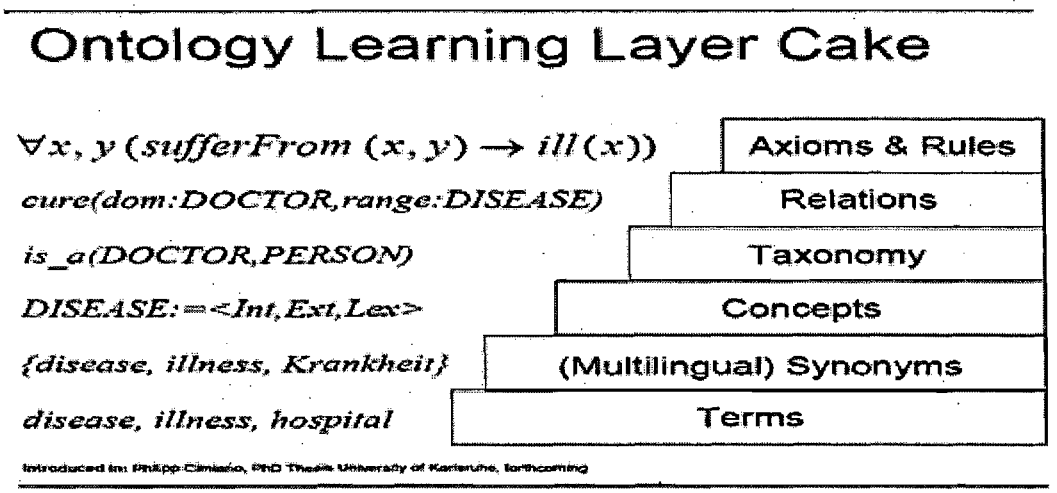


Figure 4. Étapes d'extraction d'une ontologie du domaine (Buitelaar, Cimiano, Grobelnik, & Sintek, 2005)

Notons que des travaux intéressants ont présenté un état de l'art des techniques et projets dédiés à l'apprentissage d'ontologies (*Ontology learning*), notamment (Zavitsanos, Paliouras, & Vouros, 2006) (Shamsfard & Barforoush, 2003) (Maedche & Staab, 2001) (Buitelaar, Cimiano, Grobelnik, & Sintek, 2005).

2.6.1 Extraction de termes

La première étape à effectuer est d'extraire les termes du domaine. Un terme est une unité sémantique de base et peut être simple ou complexe. Les termes sont extraits au moyen de différentes techniques incluant l'analyse statistique (Dekang, 1998) (Rinaldi, et

al., 2002), l'utilisation de patrons (expressions régulières), l'analyse linguistique (repérage des phrases nominales et prépositionnelles par exemple), la désambigüisation des termes (Véronis, 2004), l'interprétation des termes composés (comme ce qui a été fait par (Navigli & Velardi, 2004) en utilisant Wordnet) ou encore sur une combinaison de ces techniques.

Par exemple, pour déterminer les termes les plus à mêmes de représenter un domaine, on peut utiliser :

- **des méthodes statistiques,** qui se basent principalement sur l'analyse des cooccurrences des termes ainsi que d'autres paramètres tels que la fréquence absolue d'un terme, la fréquence d'un terme relative à un domaine donné, etc. En droite ligne des techniques héritées de l'hypothèse de Harris (Harris, 1968), ces méthodes déterminent un score représentant le lien qui existe entre deux termes et retiennent ceux dont le score est supérieur ou égal à un seuil donné. Par exemple, La combinaison de la mesure TF*IDF avec d'autres méthodes comme l'analyse sémantique latente peut être utilisée pour retrouver les concepts du domaine (Fortuna, Mladevic, & Grobelnik, 2005). Il est à noter que ces méthodes occultent les termes statistiquement non significatifs.

- **des méthodes linguistiques** : Après une analyse linguistique du texte, des règles ou patrons sont utilisés pour extraire certaines catégories grammaticales ou certaines combinaisons de catégories. D'autres règles peuvent être utilisées pour ignorer certaines catégories telles que les déterminants ou certains termes précis (mots-outils). De manière générale, on s'appuie sur des indices structurels (informations lexicales, morphologiques, syntaxiques ou sémantiques) pour détecter des termes d'intérêt (Claveau, 2003). L'analyse morphologique peut être intéressante pour la sélection de termes du domaine. Après avoir appliqué une analyse morphologique, qui permet entre autres de retrouver la racine des termes, il est possible d'essayer de détecter certains termes finissant par certains suffixes. Cette technique a l'avantage de ne pas requérir

une analyse de texte profonde (pas de *POS tagging*) mais seulement de surface. Par ailleurs elle peut constituer un remède à l'inconvénient évoqué dans les mesures d'extraction de termes statistiques : certains termes non fréquents peuvent avoir un intérêt pour le domaine et peuvent donc être extraits par l'examen de certains suffixes (Cohen, 1995).

- **des méthodes hybrides** utilisent des règles linguistiques pour extraire des termes candidats et des statistiques ou de l'apprentissage machine pour filtrer ces termes. Par exemple, les syntagmes nominaux sont extraits comme candidats potentiels et des techniques d'apprentissage machine sont utilisées pour en retenir certains et en éliminer d'autres.

2.6.2 Extraction de synonymes

La seconde étape consiste à repérer les synonymes parmi les termes extraits, cela permet d'associer différents termes au même concept que ce soit dans une même langue ou dans différentes langues (Buitelaar, Cimiano, Grobelnik, & Sintek, 2005). L'extraction de synonymes s'effectue généralement de deux façons :

- par des techniques de classification, qui permettent de catégoriser des termes selon des classes préexistantes. On utilise souvent Wordnet à cet effet où les « *Synsets* » sont considérés comme les classes à étendre (Hearst, 1992) (Kavalec & Svatek, 2005).

- par des techniques de catégorisation qui permettent de regrouper des termes situés dans un même contexte (par exemple, les cooccurrences de termes (Baroni & Bisi, 2004)).

2.6.3 Extraction de concepts

Il importe ensuite de déterminer, parmi les termes existants, ceux qui sont des **concepts**. Selon Buitelaar et al. (Buitelaar, Cimiano, Grobelnik, & Sintek, 2005), un terme peut représenter un concept si on peut définir :

- **son intention**, c'est-à-dire la définition, formelle ou non, de l'ensemble des objets que le concept décrit. Par exemple, un félin est un mammifère carnassier qui ....

- **son extension**, c'est-à-dire l'ensemble des objets ou instances que la définition du concept décrit. Par exemple : lion, panthère, tigre.

- **ses réalisations lexicales**, c'est-à-dire un ensemble des synonymes dans différentes langues. Par exemple : Lion (fr, en), León, etc.

Extraire un concept revient donc à en découvrir l'intention et/ou l'extension. Différentes techniques sont utilisées à cet effet. De manière générale, on peut dire que des concepts peuvent être créés à partir de l'extraction de termes disponibles dans les textes (méthode linguistique ou mixte), ou de manière statistique, où un concept émerge à partir de la définition d'attributs ou de fonctionnalités. Dans ce dernier cas, il n'est pas impossible que le concept ne figure aucunement dans les textes dont il est issu (Shamsfard & Barforoush, 2003).

L'hypothèse de Harris a également donné lieu à différentes techniques de **catégorisation** permettant de construire des concepts et des hiérarchies de concepts sans aucune connaissance préalable, les concepts étant créés en regroupant des objets selon certaines caractéristiques. Il s'agit essentiellement de techniques non supervisées. Il existe en général trois types de catégorisation (Zavitsanos, Paliouras, & Vouros, 2006) (Buitelaar, Cimiano, Grobelnik, & Sintek, 2005) (Cimiano, Hotho, & Staab, 2004b) (Memmi, 2000) :

- une **catégorisation hiérarchique** (*Agglomerative Clustering*) qui au départ assigne chaque terme à un « *cluster* ». Ensuite, des clusters plus larges sont générés au fur et à mesure de manière à contenir des termes ayant certaines similarités. Ces similarités peuvent être basées sur des propriétés lexicales (formes racines, lemmatisation, etc.) ou contextuelles.

- une **catégorisation non hiérarchique** (*Partitional Clustering*) : à l'inverse de la première technique, tous les termes sont regroupés au départ dans un même cluster. Ensuite, des sous-clusters sont créés en se basant sur certaines caractéristiques communes aux termes. Un exemple d'une telle technique est l'algorithme des k-moyennes (*K-Means*) ou les méthodes basées sur les réseaux neuronaux. Les méthodes non-hiérarchiques donnent directement une classification à un seul niveau de partition, et sont plus simples (Memmi, 2000).

- une **catégorisation conceptuelle** qui regroupe des concepts en fonction de la distance sémantique qui existe entre ces concepts et essaie d'en découvrir les caractéristiques, c'est-à-dire les propriétés (Faure & Poibeau, 2000). L'analyse formelle de concepts (AFC) (Ganter & Wille, 1999) est un exemple de catégorisation conceptuelle. L'AFC spécifie en entrée une matrice constituée d'un ensemble d'objets et de propriétés et trouve tous les clusters d'objets (un cluster étant constitué des objets partageant un ensemble de propriétés) et les clusters de propriétés (constitué de l'ensemble des propriétés relatives à un même ensemble d'objets). Dans l'AFC, un concept est défini comme une paire (ensemble de propriétés, ensemble d'objets partageant ces propriétés).

En sus de la catégorisation, des techniques d'apprentissage machine basées sur la **classification** peuvent également être utilisées. Contrairement aux techniques de catégorisation précédemment citées, des données d'entraînement sont fournies aux algorithmes pour qu'ils puissent en sortir un modèle. Dans les techniques de forage de textes, les attributs peuvent être reliés à l'orthographe des termes (Collier, Nobata, & Tsujii, 2000), à des préfixes ou à des suffixes (Shen, Zhang, Zhou, Su, & Tan, 2003) ou à des catégories syntaxiques (*Part-Of-Speech*).

Les techniques de classification sont généralement utilisées lorsque l'on dispose de classes de départ, comme une taxonomie par exemple, et que l'on veut assigner des termes

à ces classes. A partir d'un modèle créé à partir d'un corpus d'entrainement, ces algorithmes peuvent ajouter de nouveaux termes sous forme de sous-classes. Des exemples de tels algorithmes sont les algorithmes kNN (*k Nearest Neighbours*), HMM (*Hidden Markov Models*) et SVM (*Support Vector Machines*).

2.6.4 Extraction d'une taxonomie

L'extraction d'une taxonomie revient à trouver les liens « *is-a* », autrement dit les classes et sous-classes ou les hyponymes. On parle de relations taxonomiques. «*Taxonomies are widely used to organize ontological knowledge using generalization/specialization relationship through which simple/multiple inheritance could be applied.* » (Corcho & Gómez-Pérez, 2000). L'extraction de ces relations peut intervenir dans le corpus même mais également sur le Web (Etzioni, et al., 2004) ou en exploitant Wordnet (Cimiano, Pivk, Schmidt-Thieme, & Staab, 2004).

(Cimiano, Pivk, Schmidt-Thieme, & Staab, 2004) ont d'ailleurs effectué une bonne récapitulation des techniques utilisée pour l'extraction de relations taxonomiques.

2.6.4.1 Utilisation de méthodes statistiques

L'hypothèse de similarité distributionnelle couplée à des mesures de distance peut être utilisée pour enrichir une taxonomie existante avec de nouveaux concepts. Ces mesures peuvent être utilisées en comparant des vecteurs représentant le contexte d'un mot avec ceux déjà disponibles dans l'ontologie.

2.6.4.2 Utilisation de patrons lexico-syntaxiques

Une des méthodes les plus connues pour l'extraction de relations taxonomiques a été popularisée par les travaux de Hearst (Hearst, 1992). Concrètement, un ensemble de patrons linguistiques d'hyponymie sont recherchés dans les textes et permettent de déduire, avec relativement de succès, l'existence d'une telle relation (Tableau I.)

Patrons d'hyponymie
NP such as NP, NP, ... and NP
Such NP as NP, NP, ... or NP
NP, NP, ... and other NP
NP, especially NP, NP,... and NP
NP is a NP.

Tableau I. Patrons d'hyponymie

Les patrons de Hearst ont été étendus par d'autres patrons dans (Iwanska, Mata, & Kruger, 2000) ou été enrichis par des recherches sur le Web (Etzioni, et al., 2004). Des méthodes d'apprentissage machine ont également été utilisées pour générer de nouveaux patrons de manière automatique (Banko, Cafarella, Soderland, Broadhead, & Etzioni, 2007) (Stevenson & Greenwood, 2006). Ces nouveaux patrons peuvent ensuite être recherchés pour extraire de nouvelles connaissances ou pour confirmer la fiabilité d'autres patrons et d'autres relations précédemment extraits.

2.6.4.3 Utilisation d'heuristiques linguistiques

Un peu comme dans le cas des patrons linguistiques, les heuristiques linguistiques exploitent la structure interne des phrases nominales et essaient par exemple de détecter les phrases nominales et les adjectifs qui leur sont accolées de manière à créer une relation « *is-a* ». En effet, les modifiants tels que les adjectifs ou les noms restreignent la portée ou la sémantique du nom qu'ils modifient. Par exemple, dans le terme « *Intelligent Tutoring System* » où l'adjectif est « *intelligent* », on peut déduire : *is-a (intelligent tutoring system, tutoring system)*. Ce type d'heuristique a été largement utilisé dans des projets tels que ONTOLEARN (Navigli & Velardi, 2004), ONTOLT (Buitelaar, Olejnik, & Sintek, 2004) ou TEXT-TO-ONTO (Maedche & Staab, 2000c).

2.6.4.4 Utilisation de techniques de catégorisation

Les relations taxonomiques peuvent être également extraites via des techniques de catégorisation. Ces techniques sont basées sur des mesures de similarité géométriques (cosinus, Euclidienne or distance Manhattan), ou autre (entropie relative, information mutuelle), etc.

Certains travaux ont proposé la catégorisation de deux termes qui ont un hyperonyme en commun (Cimiano & Staab, 2005), ce qui permet de labéliser les clusters avec l'hyperonyme en question. En effet, un des grands problèmes des approches non supervisées est la difficulté d'affecter un label aux clusters. D'autres recherches utilisent l'analyse formelle de concepts pour l'acquisition d'une taxonomie.

Des techniques de catégorisation ont été utilisées dans les travaux de (Cederberg & Widdows, 2003) (Bisson, Nedellec, & Canamero, 2000) (Cimiano, Hotho, & Staab, 2004b) (Iwanska, Mata, & Kruger, 2000). Là encore, la majorité de ces travaux se basent sur l'hypothèse de Harris.

2.6.5 Extraction de relations sémantiques

L'extraction de toute relation qui ne soit pas taxonomique rentre dans ce cadre. En effet, d'autres types de relations caractérisées peuvent être recherchés dans les textes. Par relation sémantique, on entend non seulement les relations dénotées dans le texte par des prédicats implicites ou explicites (relations syntagmatiques), mais également les relations dites paradigmatiques (synonymie, antonymie, méronymie, hyperonymie, hyponymie, ...) (Claveau, 2003) (Cruze, 1986), qui structurent l'espace sémantique. Nous avons exploré quelques exemples de ces relations paradigmatiques dans les sections précédentes (extraction d'une taxonomie, extraction de synonymes).

Les relations syntagmatiques ou prédicatives reposent sur l'idée qu'un prédicat peut être détecté par le repérage de certaines formes syntaxiques. Robison (Robison, 1970) a

identifié 8000 patrons associés à un mot déclencheur (patron primaire) et reliés à des structures syntaxiques définies (patrons secondaires) (Morin, 1999). Le tableau suivant donne quelques exemples de relations sémantiques explorées dans la littérature.

Les relations de composition (Part-whole relationships) : (Berland & Charniak, 1999) (Girju, Badulescu, & Moldovan, 2003) (Girju, Badulescu, & Moldovan, 2006) (Van Hage, Kolb, & Schreiber, 2006)
Les relations causales : (Girju, 2003) (Girju & Moldovan, 2002)
Les relations de "Qualia" : (Yamada & Baldwin, 2004), (Cimiano & Wenderoth, 2007)
Les attributs : (Poesio & Almuhareb, 2005)

Tableau II. Exemples de relations sémantiques caractérisées

D'autres relations peuvent être découvertes également sans être reliées a priori à une catégorie sémantique. La découverte de relations, qu'elles soient sémantiquement caractérisées ou non, s'effectue là encore par des méthodes linguistiques ou statistiques : des techniques de sous-catégorisation ou d'acquisition de structures d'arguments sont utilisées pour la découverte des relations syntagmatiques (Resnik, 1993) (Faure & Nedellec, 1998). Certains marqueurs syntaxiques tels que les compléments appositifs permettent d'extraire des relations labélisées (Byrd & Ravin, 1999).

L'analyse statistique seule sert également à retrouver les relations syntagmatiques. Elle utilise pour cela la même méthodologie que pour la détection de termes mais en choisissant des fenêtres de cooccurrence plus grandes (Claveau, 2003). Comme exemple de technique statistique d'apprentissage de relations, les algorithmes d'apprentissage **de règles d'association** essaient de découvrir les éléments qui co-occurrent fréquemment et de déduire des règles régissant cette cooccurrence telles que l'implication ou la corrélation

(Maedche & Staab, 2000a) (Maedche & Staab, 2000b). Ce type d'algorithmes, en général basé sur le travail de (Srikant & Agrawal, 1997), a été beaucoup utilisé pour l'acquisition de relations taxonomiques destinées à l'enrichissement d'ontologies existantes (Etzioni, et al., 2004). Les règles d'association produisent toutefois des relations non labélisées, ce qui est un inconvénient de taille. Une modification de l'algorithme a été proposée par (Kavalec & Svatek, 2005) pour tenter d'y remédier.

De nombreux travaux combinent généralement de l'analyse statistique avec des niveaux plus ou moins complexes d'analyse linguistique, par exemple en exploitant la structure syntaxique et les dépendances provenant de l'analyse linguistique. D'autres techniques s'appuient sur l'analyse des cooccurrences des termes pour déduire l'existence ou non d'une relation (Grefenstette, 1993). Ces relations peuvent être dotées d'un label (Sánchez & Moreno, 2006) ou pas (Maedche & Staab, 2000a), et ce label peut être extrait du texte ou pas. Lorsque les relations n'ont pas de label, la relation n'existe que pour indiquer un certain lien entre les concepts qu'elle relie. D'autres techniques regroupent les termes en fonction des verbes avec lesquels ils co-occurrent, comme dans le système ASIUM par exemple (Faure & Nedellec, 1998).

2.6.6 Extraction d'axiomes et de règles

L'extraction d'axiomes constitue le dernier niveau du processus d'apprentissage d'ontologies et également le plus difficile. Jusqu'à maintenant, peu de projets se sont attaqués à la découverte d'axiomes et de règles à partir de textes. Dans (Volker, Hitzler, & Cimiano, 2007) (Volker, Vrandecic, Sure, & Hotho, 2007), on retrouve toutefois des tentatives de découverte d'axiomes dans l'équipe qui a réalisé TEXT-TO-ONTO / Text-2-Onto (Cimiano & Völker, 2005). Pour le moment, notre opinion est de considérer cette partie comme étant l'apanage de l'expert du domaine, étant donné la difficulté qu'il y a déjà à extraire les autres composants de l'ontologie.

2.6.7 Enrichissement d'ontologies existantes

Enfin, plutôt que de créer des ontologies sans aucune base de départ, certaines approches tentent d'enrichir des ontologies existantes. Nous en avons parlé dans certaines techniques de classification supervisées. Étant donné un corpus de documents et une ontologie de départ, des techniques statistiques, d'apprentissage machine et linguistiques peuvent être utilisées.

Ces ontologies peuvent être des ontologies de haut niveau (génériques, de tâches) et guider le processus d'enrichissement. En combinaison avec Wordnet (Navigli & Velardi, 2004), avec un dictionnaire ou thésaurus (Park, 2004) ou en exploitant le Web (Banko, Cafarella, Soderland, Broadhead, & Etzioni, 2007) (Agirre, Ansa, Hovy, & Martinez, 2000), de nouveaux concepts et relations peuvent être identifiés via les mêmes techniques présentées jusqu'ici.

La section suivante présente un ensemble de projets œuvrant à l'acquisition d'ontologies du domaine à partir de textes. Le but de cet état de l'art global est d'offrir un panorama sur ce qui a été réalisé en la matière, et enfin de comparer et de critiquer ces projets.

2.7 Projets d'extraction automatique d'ontologies du domaine

L'utilisation de techniques d'acquisition des connaissances à partir de textes n'est pas une problématique nouvelle. Les techniques provenant du traitement de la langue naturelle et du forage de textes non plus. Toutefois, on entend de plus en plus parler de techniques d'acquisition d'ontologies à partir de textes, c'est pourquoi nous nous focaliserons surtout, dans cet état de l'art, sur les projets les plus récents reliés à l'acquisition et à la population d'ontologies à partir de textes.

2.7.1 Les projets basés essentiellement sur des méthodes linguistiques

Parmi les projets basés essentiellement sur des méthodes linguistiques, on peut notamment citer SYNDICATE, ONTOLT et MO'K.

Le projet SYNDICATE (Hahn & Romacker, 2000) permet de transformer des textes en base de connaissances textuelles. Le projet ne s'attaque pas seulement à l'analyse linguistique des phrases, mais tient compte de la structure cohésive du discours. Le système requiert toutefois une connaissance du domaine préalable représentée en KL-ONE (Brachman & Schmolze, 1985) et sert donc plutôt à la population d'une ontologie qu'à la représentation de leur structure.

Le projet ONTOLT (Buitelaar, Olejnik, & Sintek, 2004) met en œuvre des méthodes surtout linguistiques pour l'extraction de connaissances. Il utilise des patrons linguistiques et sémantiques qui font le lien entre des structures linguistiques complexes et des concepts et relations sémantiques. ONTOLT se présente comme un plugin de l'environnement Protégé, ce qui a l'avantage de permettre l'intégration d'ontologies provenant de méthodes manuelles et de méthodes automatiques. Les concepts sont extraits sous forme de classes dans Protégé et les relations sous forme de « *slots* ». ONTOLT nécessite en entrée des documents préalablement annotés selon le format MM (Vintar, Buitelaar, Ripplinger, Sacaleanu, Raileanu, & Prescher, 2002) et présentés sous forme XML.

Le projet MO'K (Bisson, Nedellec, & Canamero, 2000) est un outil de construction d'ontologies spécialisé dans les méthodes de catégorisation conceptuelle. Il offre des outils pour l'évaluation et la comparaison de différentes représentations et permet de changer certains paramètres comme les mesures de distance utilisées de manière à trouver un résultat optimal. MO'K met en œuvre l'hypothèse de Harris en considérant que les relations syntaxiques entre les mots peuvent être utilisées pour dériver des relations sémantiques. L'algorithme reçoit un corpus du domaine étiqueté en entrée et vise à obtenir une

taxonomie de concepts. Il utilise des techniques de traitement de la langue naturelle pour extraire des triplets composés d'une structure : « verbe, mot et le rôle syntaxique du mot dans la phrase». MO'K calcule ensuite le nombre d'occurrences de chaque triplet. Ceux qui sont trop rares ou trop fréquents sont ignorés. Enfin, la distance sémantique entre les triplets est calculée pour former des catégories conceptuelles.

2.7.2 Les projets basés essentiellement sur des méthodes d'apprentissage machine

Le projet ONTOLEARN (Navigli & Velardi, 2004) utilise majoritairement des méthodes statistiques et des méthodes d'apprentissage machine pour identifier les concepts et les relations sémantiques. Il commence par l'extraction d'une terminologie domaine à partir de sites Web et de documents spécialisés. Pour ce faire, le système effectue une analyse comparative en utilisant des techniques statistiques sur différents domaines ou corpus. Cette analyse lui permet de filtrer les termes du domaine et de les séparer de ceux qui ne représentent pas le domaine en question. Les termes sont ensuite interprétés sémantiquement en utilisant Wordnet (Fellbaum, 1998) et SemCor (SemCor, 2002) et des relations taxonomiques sont créées, ce qui permet de générer une forêt de concepts (*Domain Concept Forest*). Cette forêt est ensuite intégrée à Wordnet pour créer une ontologie domaine, qui doit être validée par un expert.

KNOWITALL (Etzioni, et al., 2004) (Popescu, Yates, & Etzioni, 2004) permet d'extraire des concepts, des relations et des faits à partir du Web. Il est utilisé pour étendre une ontologie existante et contient un ensemble de règles génériques de départ qui lui servent à extraire de nouvelles règles pour chaque classe et relation de l'ontologie. KNOWITALL est indépendant du domaine et du langage utilisé. Pour ses recherches sur le Web, KNOWITALL formule automatiquement des requêtes en utilisant les nouvelles règles qu'il a extraites. Chaque règle est associée à une requête composée des mots-clés contenus dans la règle. Les pages relatives à chaque requête sont téléchargées et leurs phrases sont analysées à la recherche de nouvelles relations. La fiabilité de chaque

connaissance extraite est fonction des statistiques fournies par les moteurs de recherche. KNOWITALL utilise une mesure nommée « *pointwise mutual information (PMI)* » (Turney, 2001) entre les mots-clés recherchés et les phrases qui les contiennent. Si cette mesure est élevée, cela indique probablement que la relation est correcte. Dans (Etzioni, et al., 2004), les auteurs fournissent l'exemple suivant : Si l'extracteur de KNOWITALL propose « Liège » comme étant une « Ville » et si la mesure PMI entre « Liège » et des phrases comme « ville de Liège » est élevée, alors on peut conclure que « Liège » est bien une instance valide de «Ville».

ONTOGEN (Fortuna, Grobelnik, & Mladenič, 2006b) (Fortuna, Grobelnik, & Mladenič, 2006a) est un outil d'aide à la construction d'ontologies, qui suggère des concepts à l'utilisateur à partir d'une large collection de documents. ONTOGEN offre également des outils de visualisation afin d'assister l'usager dans le processus de construction. Pour l'extraction des concepts, ONTOGEN utilise de nombreuses techniques statistiques et d'apprentissage machine telles que *TF*IDF, Support Vector Machines, Latent Semantic Indexing* et *k-means clustering*.

L'extraction de connaissances dans le projet SNOWBALL (Agichtein, 2005) (Agichtein & Gravano, 2000) commence avec un ensemble d'instances exemples représentant des relations recherchées et des entités nommées. Des occurrences de ces concepts et relations sont recherchées et des patrons d'extraction sont générés à partir de ces exemples constitués de combinaisons de certaines entités et relations.

Dans SNOWBALL, un patron est représenté par un centroïde de catégorie (*cluster centroid*). Un vecteur de termes est généré pour chaque portion de texte (contexte) où une relation intéressante est repérée et les différents vecteurs sont ensuite regroupés au moyen d'un algorithme de catégorisation. Une évaluation des patrons est effectuée et les plus fiables sont retenus comme patrons d'extraction valides à rechercher. La fiabilité d'un patron provient d'un ensemble de poids que SNOWBALL assigne aux patrons qu'il génère.

Ces patrons sont ensuite utilisés pour extraire de nouvelles instances dans le corpus et ainsi de suite.

TEXTRUNNER (Banko, Cafarella, Soderland, Broadhead, & Etzioni, 2007) extrait des tuples de corpus hétérogènes sans requérir une intervention humaine. Selon ses auteurs, TEXTRUNNER facilite la mise à échelle nécessaire dans un environnement tel que le Web. Il se base sur un module, qui à partir d'un corpus échantillon non annoté, extrait un classificateur (*classifier*) qui décide si les candidats extraits sont fiables ou pas. TEXTRUNNER n'utilise pas d'analyseur linguistique. A partir d'une seule itération sur le corpus, Il génère des tuples pour chaque phrase rencontrée, introduit ces tuples dans le classificateur et retient seulement ceux considérés comme fiables. Cette fiabilité est mesurée par un modèle probabiliste introduit par (Downey, Etzioni, & Soderland, 2005).

2.7.3 Les projets utilisant des méthodes hybrides

Le projet ASIUM (Faure & Nedellec, 1998) a été cité à de multiples reprises dans la littérature pour l'apprentissage de schémas verbaux (*verb subcategorization frames*). ASIUM considère les noms de même catégorie syntaxique apparaissant avec un même verbe comme étant sémantiquement reliés. Il crée ainsi des sortes de « *clusters* » sémantiques composés de termes apparaissant de manière concomitante dans au moins deux contextes différents. Une technique de catégorisation est ensuite appliquée pour former des concepts plus génériques et donc des relations taxonomiques. Cela permet également de généraliser les schémas verbaux. Une intervention humaine est toutefois requise pour vérifier les généralisations apprises. Le projet ASIUM rentre dans le cadre de l'extraction de relations syntagmatiques.

Le projet TEXT-TO-ONTO (Maedche & Staab, 2000c) est un des projets qui a le plus vulgarisé la notion d'acquisition d'ontologies à partir de textes. TEXT-TO-ONTO est une suite logicielle qui s'appuie sur l'atelier KAON (Bozsak, et al., 2002) et est composée d'un ensemble de techniques de forage de textes, principalement basées sur l'apprentissage

machine mais faisant également appel à quelques techniques linguistiques telles que l'appariement de patrons ou «*pattern matching*». Les techniques statistiques et d'apprentissage machine sont utilisées aussi bien pour la découverte de concepts (Maedche & Staab, 2001) que pour la découverte de relations taxonomiques (Cimiano, Hotho, & Staab, 2004a) (Cimiano, Hotho, & Staab, 2004b) (Cimiano, Pivk, Schmidt-Thieme, & Staab, 2004) et non taxonomiques (Maedche & Staab, 2000a) (Maedche & Staab, 2000b).

TEXT2ONTO (Cimiano & Völker, 2005) est le successeur de TEXT-TO-ONTO. Les changements majeurs introduits par TEXT2ONTO relèvent du rattachement de probabilités aux connaissances extraites, représentées par des méta-modèles. Par ailleurs, ces probabilités sont mises à jour automatiquement lorsque des modifications de corpus sont effectuées.

Les projets cités ci-dessus sont parmi ceux qui ont eu le plus d'impact dans la dernière décennie, du moins en termes de citations dans la littérature. Toutefois, d'autres projets comme ONTOBASIS (Reinberger & Daelemans, 2004), RELEXT (Schutz & Buitelaar, 2005), ou TOKO (Anjewierden & Efimova, 2006) sont d'autres exemples d'outils permettant la génération d'ontologies et sont dignes d'intérêt.

Cet ensemble de projets constitue un état de l'art des travaux réalisés durant la dernière décennie. Il n'est pas facile de faire la synthèse des divers travaux réalisés. On peut toutefois constater que plusieurs projets utilisent maintenant les techniques d'apprentissage machine pour généraliser, classifier ou apprendre de nouveaux patrons d'extraction (KNOWITALL, TEXTRUNNER, ONTOGEN, SNOWBALL, ONTOLEARN) sans forcément recourir à de l'analyse linguistique. Certains ont combiné les deux approches. C'est le cas de TEXT-TO-ONTO, ASIUM (plus spécialisé dans les sous-catégorisations verbales), ou encore MO'K (spécialisé dans l'apprentissage de catégories conceptuelles). Les deux derniers systèmes s'attachent plus à une sous-partie de l'apprentissage d'ontologies qu'au processus dans son ensemble. Seul TEXT-TO-ONTO

nous semble prendre en charge l'ensemble du processus. C'est la raison pour laquelle nous effectuons une analyse comparative en utilisant cet outil (voir chapitre 4).

Le tableau suivant récapitule certains traits caractéristiques des projets évoqués.

Projet	*Objectif*	*Techniques d'extraction*	*Source et ressources nécessaires*
KNOWITALL	Extraire des instances (entités)	Patrons (Règles), Patrons inspirés de Hearst, appariement de chaines de caractères	HTML
TEXT-TO-ONTO / TEXT2ONTO	Extraire une ontologie	Patrons – Apprentissage statistique de concepts – Règles d'association – Analyse formelle de concepts	Textes (structurés et non structurés) Dictionnaires Ontologies (Wordnet)
TEXTRUNNER	Extraire des relations sous forme de triplets	Patrons	HTML
ONTOGEN	Extraire une ontologie	Analyse statistique-Catégorisation	Textes
ONTOLEARN	Enrichir une ontologie	Analyse linguistique Apprentissage machine Statistiques	Textes Dictionnaires Wordnet
SNOWBALL	Extraire des instances	Patrons prédéfinis - Génération de nouveaux patrons	HTML, WEB
MO'K	Extraire des classes et relations taxonomiques	Catégorisation conceptuelle	Texte étiqueté
ASIUM	Extraire des relations conceptuelles	Catégorisation conceptuelle	Texte étiqueté

	et taxonomiques		
ONTOLT	Extraire des classes et des rôles	Règles d'appariement linguistiques Statistiques	Texte étiqueté
SYNDICATE	Extraire des instances	Analyse linguistique	Textes (allemand) Lexiques et ontologies (génériques et domaine)

Tableau III. Comparaison des projets d'extraction d'ontologies à partir de textes

Un second tableau sert à illustrer l'accomplissement, par les différents projets, des étapes d'extraction d'une ontologie du domaine (présentées dans le chapitre 2). Notons que le signe « ×~ avec label » concernant TEXT-TO-ONTO indique que c'est une fonctionnalité sensée être implantée dans le projet mais qui a fourni des résultats décevants dans nos expérimentations (voir chapitre 4). Le terme « clusters » apparaissant dans la colonne « Concepts » indique que le projet a défini un concept en regroupant un ensemble de termes (avec ou sans label pour le concept en question).

Projet	*Termes*	*Synonymes*	*Concepts (classes primitives)*	*Classes définies*	*Relations taxonomiques*	*Relations conceptuelles / triplets*	*Instances*
KNOWITALL						×	×
TEXT-TO-ONTO / TEXT2ONTO	×	×	×		×	×~ avec label × sans label	×
TEXTRUNNER	×	×				×	×

ONTOGEN	×		×		×	×	
ONTOLEARN	×	×	×		×	×	
SNOWBALL	×					×	×
MO'K	×		clusters		×		
ASIUM	×		clusters		×	×	
ONTOLT	×		×		×	×	
SYNDICATE	×					×	×

Tableau IV. Comparaison des projets basée sur les différentes étapes d'extraction

Dans ces projets, certaines faiblesses sont à relever :

- Le manque de préservation de la structure des phrases.
- La difficulté d'extraire des concepts sans guider le processus d'extraction par des structures relatives au domaine telles que les thésaurus par exemple.
- La difficulté de produire des relations labélisées entre concepts.

2.8 En résumé

L'extraction de connaissances à partir de textes peut représenter une alternative intéressante à la construction manuelle d'ontologies du domaine. L'utilisation de méthodes automatiques pour l'extraction d'ontologies à partir de textes devrait permettre leur adoption à grande échelle et devrait contribuer au développement du Web sémantique.

Nous avons présenté un ensemble de méthodes permettant aussi bien l'apprentissage que la population d'ontologies. Ces méthodes se scindent en trois grands blocs : les méthodes statistiques et d'apprentissage machine, les méthodes linguistiques et les méthodes hybrides. La majorité des approches intéressantes ont utilisé les techniques hybrides. D'ailleurs, (Aussenac-Gilles, Biébow, & Szulman, 2000) et (Cimiano, Pivk,

Schmidt-Thieme, & Staab, 2004) ont conseillé l'adoption d'une suite logicielle combinant toutes les techniques développées, offrant ainsi un choix de méthodes à l'ingénieur d'ontologies, lui permettant d'expérimenter diverses alternatives et de comparer leurs résultats.

Nous avons souligné certains inconvénients reliés aux différentes techniques d'extraction. La majorité des recherches qui utilisent les méthodes linguistiques ont une connaissance a priori de l'information à extraire. C'est le cas de la détection d'entités nommées, de la reconnaissance de rôles sémantiques ou des patrons sémantiques liés au domaine d'application. Seules les structures lexico-syntaxiques (Hearst, 1992) provenant de l'analyse du langage restent indépendantes du domaine d'application et ne nécessitent pas, au préalable, une base de connaissance du domaine. Les méthodes statistiques seules, quant-à elles, ne peuvent garantir à l'avance les résultats qu'elles produisent puisque par essence, ils ne sont pas prévisibles. Ces méthodes ne permettent pas de couvrir toutes les significations du langage. Par exemple : les méthodes "*Bag of words*" ignorent la structure d'une phrase alors que cette structure est particulièrement importante dans le domaine de la formation puisqu'elle véhicule les connaissances à apprendre. Les modèles de langage statistiques ne sont pas à même de capturer les représentations sémantiques des phrases : différentes manières de combiner les mêmes mots aboutissent à des représentations sémantiques différentes, et cette différence n'est pas perçue par les modèles statistiques comme l'analyse sémantique latente par exemple.

Cette thèse vise donc une approche qui soit indépendante du domaine (elle doit pouvoir s'appliquer à divers domaines), qui soit non supervisée (on ne sait pas à l'avance quelles connaissances sont contenues dans les textes et on ignore donc ce que l'on est sensé rechercher) et qui permette la génération de structures à même de conserver la signification des phrases. En effet, le domaine qui nous intéresse, celui de la formation, requiert de repérer précisément le contenu des objets d'apprentissage pour y référer l'apprenant. Le contenu doit pouvoir être présenté à différents degrés de granularité : référer à un document

entier, à un passage, à une phrase particulière, etc. Les structures conceptuelles qui émergent de ces documents doivent donc être à même de conserver ces structures.

Par conséquent, nous postulons que les techniques d'analyse linguistique sont primordiales pour l'extraction d'ontologies à même de représenter la sémantique des objets d'apprentissage textuels. Cette analyse nous permet de préserver la structure des phrases sous forme de cartes conceptuelles, ce que nous détaillons au prochain chapitre. Toutefois nous adoptons également une approche hybride puisque les méthodes statistiques nous servent à focaliser l'analyse linguistique sur les parties importantes des documents (via la détection de mots-clés et de phrases-clés) et à formaliser ensuite certaines parties des cartes conceptuelles en un langage plus formel de façon à créer une ontologie du domaine.

A notre connaissance, dans tous les projets reliés à l'acquisition d'ontologies à partir de textes, aucun n'a été destiné au domaine de la formation en particulier. Dans ce contexte, une approche particulière d'extraction utilisant des cartes conceptuelles comme structures intermédiaires offre d'une part la possibilité de préserver la structure du discours des textes, et génère, d'autre part, des structures utiles à la formation.

3 L'acquisition de l'ontologie du domaine dans le projet « The Knowledge Puzzle » : l'outil TEXCOMON

3.1 Introduction

Les EIAH souffrent de la difficulté d'acquisition des connaissances qui leur sont nécessaires. Parmi ces connaissances, l'acquisition d'un modèle du domaine est un processus très lourd. L'acquisition semi-automatique d'un tel modèle permet donc d'alléger la tâche de l'expert humain.

De manière générale, nous avons pu constater que les systèmes d'extraction de connaissances à partir de textes se devaient de combiner les approches statistiques et d'apprentissage machine ainsi que les approches linguistiques. C'est en nous appuyant sur cette optique et forts des limites observées en matière d'acquisition des ontologies à partir de textes que nous proposons l'outil TEXCOMON. Cet outil est destiné à générer semi-automatiquement une ontologie du domaine à partir de textes et plus particulièrement à partir d'objets d'apprentissage textuels provenant de la formation à distance. Le fait de créer une ontologie destinée à la formation impose une démarche différente incluant la production de structures intermédiaires qui préservent la structure du discours. Ces structures permettent aussi l'indexation précise de portions de documents par des concepts et des relations entre ces concepts. Notre objectif dans ce chapitre est de présenter une méthodologie d'acquisition des connaissances à partir de textes permettant de concrétiser cette démarche.

3.2 Processus général

Le processus d'extraction de l'ontologie du domaine suit de manière générale les étapes indiquées à la section 2.6. Différents types de connaissances, notamment les connaissances déclaratives et procédurales, peuvent être visés par un processus

d'extraction. Dans cette thèse, seules les phrases de type assertif, qui donnent une information déclarative sur le monde (Searle, 1975) sont traitées. Compte tenu de l'origine de nos documents, qui proviennent de la formation à distance, les textes sources ou les objets d'apprentissage transmettent une connaissance du domaine bien souvent déclarative. Par ailleurs, établir un modèle du domaine pour un système de formation revient à conceptualiser les connaissances déclaratives de ce domaine, ce qui est notre but. Les connaissances procédurales figurent généralement dans un modèle à part (le modèle expert) bien souvent sous forme de règles. Le traitement des connaissances procédurales nécessiterait donc l'utilisation de techniques d'extraction différentes de celles qui sont employées actuellement.

Le nom TEXCOMON, acronyme de l'expression « TEXt- COncept Map-ONtology», illustre une des spécificités de notre approche, à savoir :

- la transformation de textes en cartes de concepts
- la formalisation de cartes de concepts sous forme d'ontologies du domaine.

TEXCOMON a pour autre spécificité d'utiliser une approche d'extraction non supervisée et indépendante du domaine. Par ailleurs, même si nous utilisons OWL (OWL Web Ontology Language Overview, 2004) comme langage de l'ontologie cible, différents formalismes d'exportation sont par ailleurs possibles à partir des cartes de concepts (OWL, RDFS, F-Logic, graphes conceptuels,...)

La Figure 5 illustre ce processus et les étapes suivies pour extraire les cartes de concepts et l'ontologie du domaine.

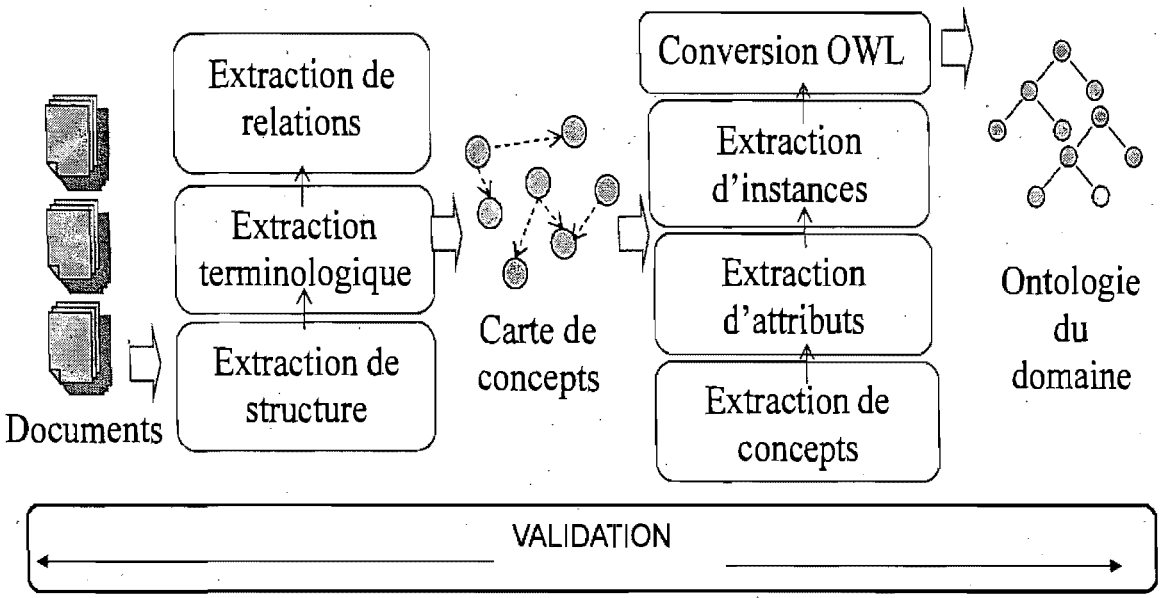


Figure 5. Processus d'extraction de cartes de concepts et d'ontologies du domaine.

Ces différentes étapes sont illustrées dans les sections suivantes. Tout au long de ce chapitre, nous présenterons des exemples issus d'un corpus sur le standard e-Learning SCORM. Ce corpus, constitué de documents en anglais, est présenté de manière plus détaillée dans le chapitre 4.

3.3 Extraction automatique de la structure des documents

Avant de pouvoir analyser un document, il est nécessaire d'en analyser la structure. En effet, cette dernière peut être significative : les titres peuvent apporter des mots-clés pour l'analyse des sections qu'ils précédent, des structures telles que « *RTE : est un environnement...* » indiquent souvent des définitions, etc.

Par ailleurs, l'unité de base à analyser dans un document lors d'une analyse linguistique est la phrase. Pour retrouver les phrases, on utilise des techniques de segmentation de textes généralement basées sur la ponctuation. Les points, les points d'interrogation, d'exclamation, de suspension marquent souvent les limites d'une phrase. Pareillement, la segmentation d'une phrase en mots est effectuée en utilisant les caractères espace et permet, lorsqu'elle est effectuée de manière conjointe à la détection de phrases, de

déjouer certaines ambiguïtés reliées aux points par exemple, dans les abréviations comme «*Inc.*» ou les appellations comme « *Mr.* ».

Pour identifier les types de composants structurels, nous avons construit une ontologie de structure de document qui vise à recenser les composants structurels et à établir des liens entre eux. Par exemple, un paragraphe dans un document « txt » est équivalent à la notion de paragraphe « <p> » dans un document HTML. Cette ontologie doit permettre l'analyse de différents types de documents (txt, doc, RTF, PDF, html) et de différentes composants (figures, tables, titres, etc.). Dans les faits, nous avons seulement traité les documents de type « txt » où les paragraphes et les phrases sont extraits de manière automatique en utilisant l'architecture d'IBM UIMA (*Unstructured Information Management Architecture*) (UIMA Java Framework, 2007).

Nous avons développé deux annotateurs pour repérer les phrases et les paragraphes d'un format donné. Une portion de texte est considérée comme un paragraphe si ce paragraphe contient au moins une phrase et que l'on atteint une ligne blanche ou la fin du fichier. Les phrases sont déterminées en fonction des signes de ponctuation et en utilisant la classe « *java.text.BreakIterator* ».

Bien évidemment, ces annotateurs ne permettent de détecter que certains types de paragraphes et il serait nécessaire d'enrichir le système, ainsi que précédemment évoqué, par d'autres formats de documents et par d'autres composants structurels. La figure suivante montre l'interface de l'outil TEXCOMON.

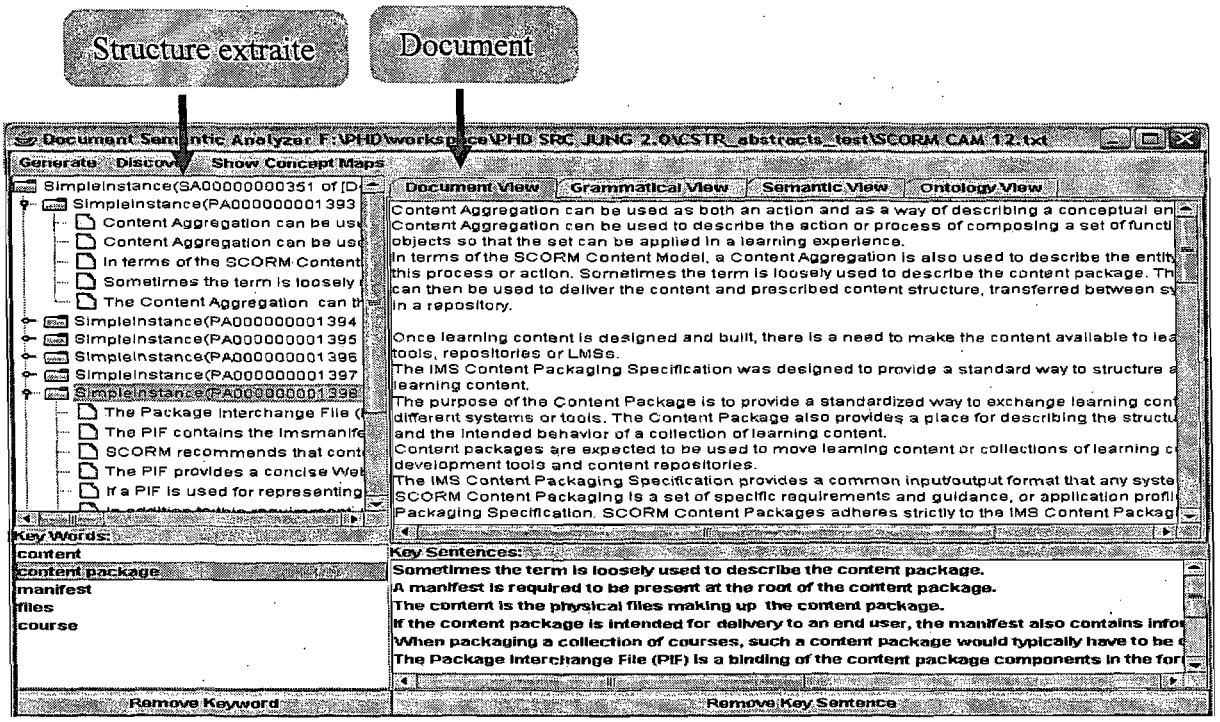


Figure 6. Extraction automatique de la structure des documents

Nous avons préalablement indiqué que notre démarche d'apprentissage d'ontologie est initiée par l'extraction de mots-clés. C'est ce que nous présentons à la section suivante.

3.4 Extraction automatique des mots-clés des documents

La majorité des systèmes s'appuient sur des experts humains pour leur fournir les mots-clés initiaux. Dans le cas de TEXCOMON, nous avons eu recours à un algorithme de détection de mots-clés présenté dans le chapitre 2 : l'algorithme KEA-3.0 (Frank, Paynter, Witten, Gutwin, & Nevill-Manning, 1999). Cet algorithme, entrainé sur un corpus de documents informatiques, permet d'obtenir des résultats intéressants (notre corpus est également relié au domaine informatique). Il est intégré dans l'outil TEXCOMON et est exécuté pour chaque nouveau document (Figure 7). Par exemple, à partir du petit texte suivant, TEXCOMON fournit les mots-clés : *asset, text, images, web, object :*

« An asset is a content object that will not use the SCORM API but that can still be used for an activity. For example, it might be a text document or an image.

Assets are electronic representations of media such as text, images, sound, web pages, assessment objects, or other pieces of data that can be delivered to a Web client.

An Asset can be described with asset metadata to allow for search and discovery within online repositories, thereby enhancing opportunities for reuse."

La figure suivante illustre là encore l'interface de TEXCOMON et l'affichage des mots-clés.

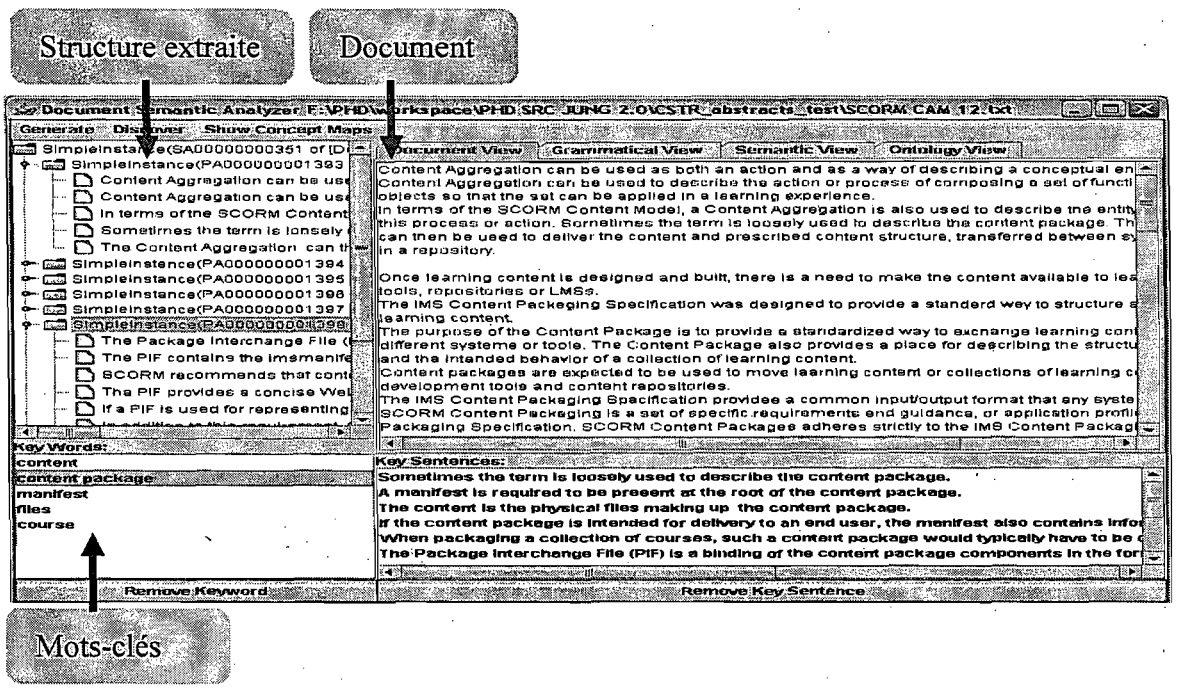


Figure 7. Extraction des mots-clés d'un document

3.5 Analyse linguistique des phrases-clés des documents

Les mots-clés servent à enclencher le processus d'extraction. Ils doivent permettre de repérer non seulement les concepts intéressants, mais également les relations sémantiques qui les relient. C'est à ce niveau qu'interviennent les méthodes linguistiques et

les analyseurs de la langue naturelle. TEXCOMON, recherche dans chaque document les phrases contenant les mots-clés. Ces phrases sont ensuite considérées comme des phrases-clés devant passer par le processus d'analyse linguistique.

L'outil TEXCOMON effectue l'analyse des phrases-clés retenues en dépendances typées en incorporant l'analyseur de Stanford. TEXCOMON produit, pour chacune des phrases clé ce que nous avons appelé une carte grammaticale de concepts(CGC). Nous avons implanté un éditeur graphique de cartes de concepts qui utilise la librairie JUNG (Java Universal Network/Graph Framework, 2007) pour la visualisation des graphes. JUNG offre diverses possibilités au niveau de l'affichage des graphes (*layouts*) et des possibilités de zoom sur le graphe (figure 8.)

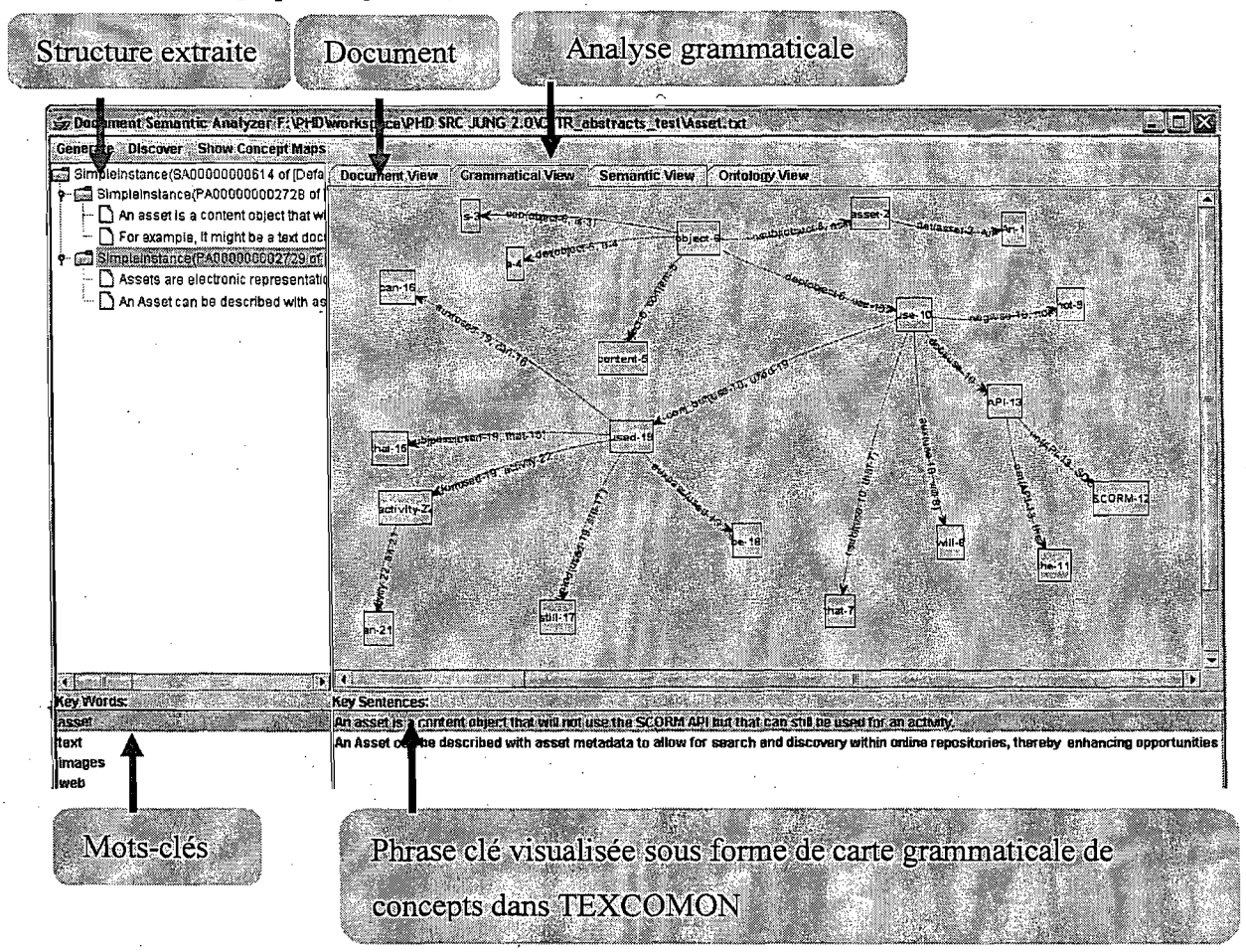


Figure 8.Vue d'une CGC dans TEXCOMON

Dans un souci de lisibilité, les relations présentées dans la CGC de la figure 8 sont reprises ci-dessous :

det(asset-2, An-1)
nsubj(object-6, asset-2)
cop(object-6, is-3)
det(object-6, a-4)
nn(object-6, content-5)
nsubj(use-10, that-7)
aux(use-10, will-8)
neg(use-10, not-9)
dep(object-6, use-10)
det(API-13, the-11)
nn(API-13, SCORM-12)
dobj(use-10, API-13)
nsubjpass(used-19, that-15)
aux(used-19, can-16)
advmod(used-19, still-17)
auxpass(used-19, be-18)
conj_but(use-10, used-19)
det(activity-22, an-21)
prep_for(used-19, activity-22)

3.6 Génération de cartes de concepts sémantiques

Les cartes grammaticales de concepts servent de base à la génération de cartes de concepts sémantiques. Dans cette génération, nous nous appuyons sur le ***principe de compositionalité*** (Godard, 2006). Ce principe est basé sur l'idée que l'ordre dans lequel apparaissent les mots d'une phrase et les relations entre ces mots déterminent la sémantique d'une phrase, en d'autres termes, que la sémantique de la phrase découle de sa structure syntaxique (Jurafsky & Martin, 2000). Les langages humains ont d'ailleurs une sémantique structurée sous forme d'arguments et de prédicats qui relient les différents concepts d'une phrase. Notre approche sémantique doit donc permettre de recréer et de capturer cette structure de prédicats, lorsque cela est possible.

L'approche utilisée par TEXCOMON pour le passage de cartes grammaticales à une représentation sémantique se base sur la recherche automatique de patrons dans les cartes grammaticales.

3.6.1 Les patrons utilisés dans TEXCOMON

Une des approches les plus souvent utilisées dans l'extraction d'information est la notion de patrons. La détection de patrons revient à rechercher leurs constituants dans le texte. Un patron est généralement constitué d'une séquence ou d'une structure en arbre.

Les patrons sous forme de séquences sont souvent exprimés à l'aide **d'expressions régulières** (Sipser, 2005). L'un des problèmes avec cette approche, c'est qu'elle implique que l'on sait ce que l'on recherche et que l'on sait comment la connaissance va être exprimée, particulièrement lorsque les expressions régulières renvoient à une certaine sémantique du domaine (comme rechercher, par exemple, les patrons reliés à une entité « Organisation » particulière). Toute connaissance exprimée autrement échappe au processus d'extraction. Pour y remédier (partiellement), un sous-ensemble de patrons, les patrons lexico-syntaxiques, sont largement utilisés en extraction d'information. Ils s'appuient sur des attributs syntaxiques pour détecter des portions de textes intéressantes.

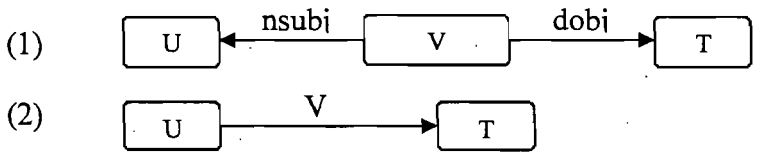
Dans cette thèse, nous nous appuyons sur des patrons lexico-syntaxiques issus de la grammaire de dépendances. TEXCOMON ne tient compte que des dépendances grammaticales pour produire une analyse sémantique. On parle d'analyse sémantique guidée par la syntaxe.

Différents modèles de patrons ont été proposés basés sur les représentations par dépendances (Stevenson & Greenwood, 2006). Ces modèles diffèrent par leur force d'expressivité et leur complexité. Dans le cadre de cette thèse, nous nous appuyons sur des patrons de type « sous-arbres » dans un arbre de dépendance. Ce type de patron considère que n'importe quel sous-ensemble de l'arbre représente un patron d'extraction. Toutefois, cette démarche correspond au besoin de rechercher automatiquement des patrons possibles

à partir de textes, ce qui n'est pas notre objectif dans cette thèse. Aussi, contrairement au modèle « Sous-Arbres », nous ne considérons pas tous les sous-graphes (sous-arbres) possibles comme patrons. Étant donné le but de l'extraction, qui est d'arriver à une carte de concepts intelligible par l'apprenant humain, nous ne pouvons utiliser que des sous-arbres dont nous « connaissons » l'analyse sémantique. Autrement dit, nous avons constitué une base de patrons lexico-syntaxiques basés sur les liens de dépendances après avoir analysé **manuellement** un ensemble d'arbres de dépendances générés à partir des phrases du corpus. Ces patrons sont génériques et ne sont pas dépendants du domaine. Nous avons ensuite étudié les transformations possibles de ces patrons grammaticaux en une représentation sémantique. Pour chacun de ces patrons, nous avons donc codé une fonction java chargée d'effectuer la transformation une fois le patron détecté. C'est cette transformation que nous appelons « **analyse sémantique** » et qui permet de transformer un sous-graphe grammatical en graphe sémantique. Quand nous parlons ici de sémantique, nous différons du sens généralement utilisé en linguistique computationnelle : la sémantique n'est pas destinée à la machine ou au système informatique (du moins à ce stade de représentation), mais bien à un utilisateur humain. La représentation résultante doit donc présenter à ce dernier, de manière claire et graphique, le contenu des documents analysés.

Dans TEXCOMON, un **patron** est défini comme un sous-arbre SG constitué d'une racine t avec des liens entrants et des liens sortants sous forme de relations grammaticales. Étant donné une carte grammaticale de concepts (CGC), chaque nœud t de la CGC est stocké comme un sous-arbre constitué de sa racine t et de deux tables de hachage représentant les liens entrants et sortants de t. Chaque lien est représenté par une structure «*Relation*» qui indique les extrémités du lien sous forme de «*DomainTerm*». La détection d'une certaine configuration déclenche la transformation de la structure grammaticale. C'est à ce niveau que servent les structures «*DomainTerm*» qui sont réutilisés pour créer des liens «sémantiques».

Par exemple, à partir de ce sous-graphe grammatical présenté en (1), TEXCOMON obtient la représentation (2). Ici nous avons trois objets de type « DomainTerm », soit U, V et T. Deux relations relient ces trois objets, soit « *nsubj* » et « *dobj* »



Plus concrètement, l'analyse sémantique de la phrase: *An asset is a content object that will not use the SCORM API but that can still be used for an activity.* » produit le résultat en figure 9.

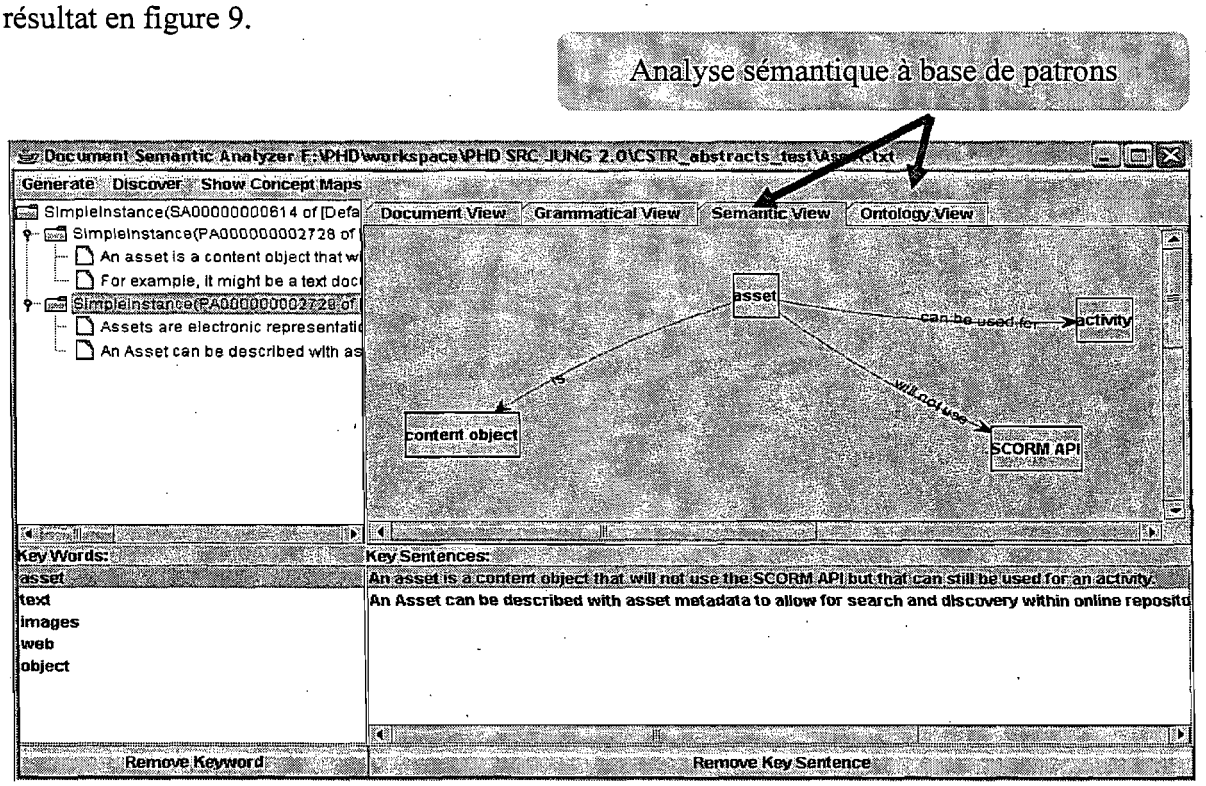


Figure 9. Analyse sémantique d'une phrase clé

La vue « *Ontology View* » permet simplement d'effectuer des transformations sur le graphe obtenu (suppression de relations erronées, modification du label d'un concept ou d'une relation, etc.).

Les différentes transformations appliquées aux CGC sont expliquées de manière détaillée dans les prochaines sections et sont regroupées selon leurs rôles syntaxiques.

3.6.2 Les patrons reliés aux groupes nominaux

Nous avons préalablement indiqué qu'un des premiers traitements à mettre en œuvre consiste à détecter les termes et les termes composés dans les phrases, autrement dit, les groupes nominaux. La détection d'adjectifs qualificatifs est également traitée à ce niveau.

Le tableau suivant résume les patrons utilisés pour la détection des phrases nominales. On parle de patrons terminologiques, car ils permettent de déterminer les termes simples et composés. Dans le tableau, t représente le terme pivot à partir duquel l'algorithme tente de détecter une des configurations listées, soit un lien sortant de type adjectif qualificatif (*amod*) ou de type nom composé (*nn*).

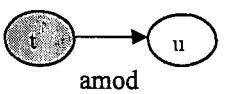
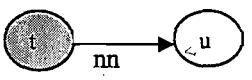
Liens entrants (t)	Liens sortants (t)	Patron	Méthode
-	Adjectif qualificatif (amod) avec u comme destination.		Créer un nouveau terme en agrégeant t et u.
-	Nom composé (nn) avec u comme destination.		Créer un nouveau terme en agrégeant t et u.

Tableau V. Des patrons terminologiques et leur méthode d'agrégation

La figure 10 présente un exemple graphique de patrons terminologiques.

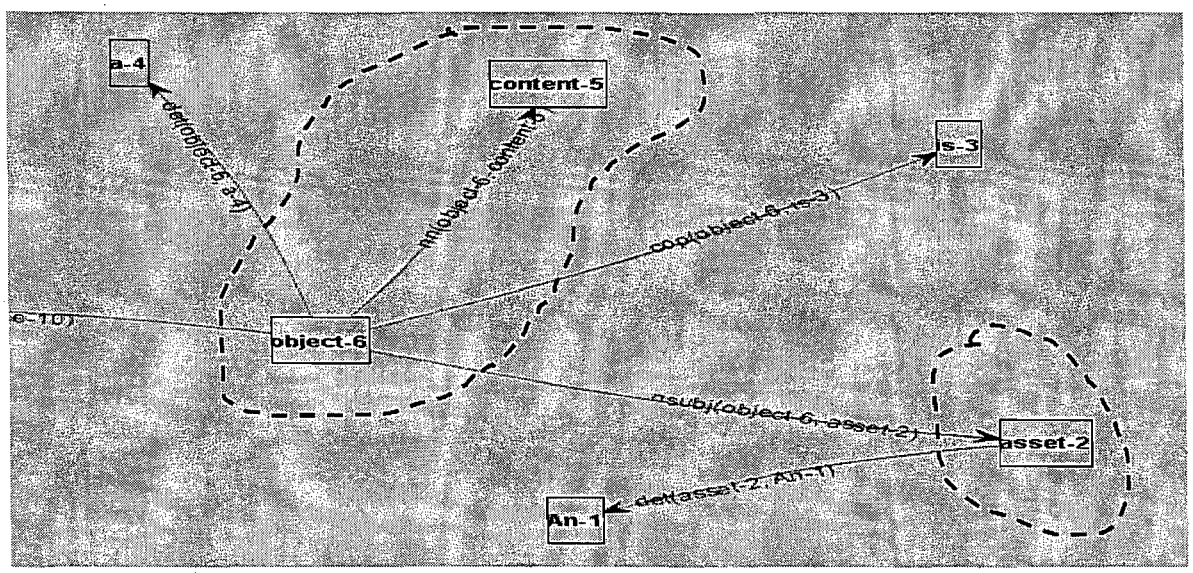


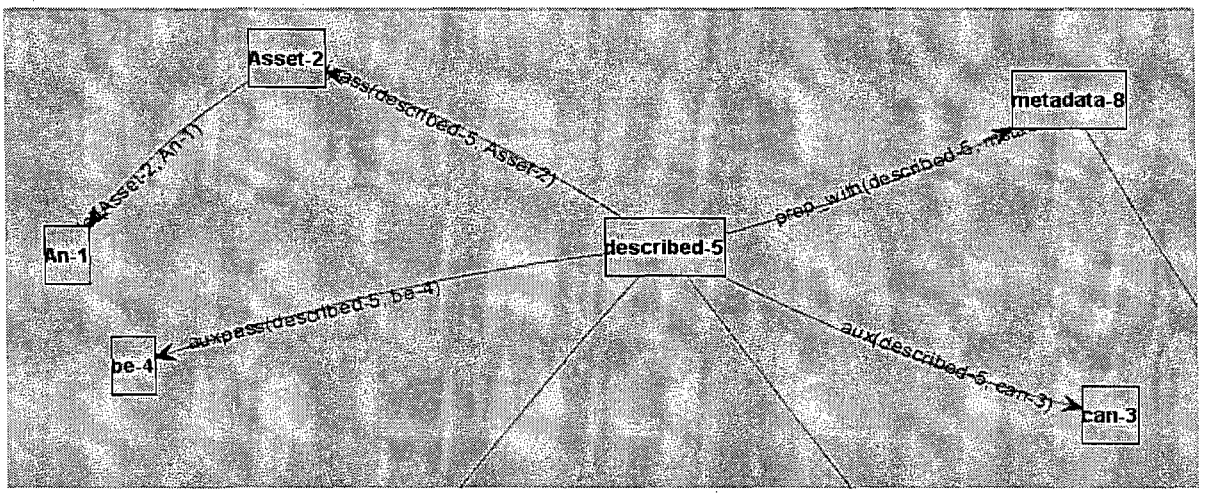
Figure 10. Des liens grammaticaux à agréger

A part les patrons qui déclenchent une agrégation des termes simples en termes composés (*nn et amod*), différents liens grammaticaux entrants (dans un terme pivot t) indiquent un groupe nominal : tous les liens désignant un sujet (Liens *subj* et leurs sous-liens), tous les liens de compléments d'objets (*obj* et leurs sous-liens), les conjonctions de coordination, les prépositions, les liens de type agent, les liens d'apposition (*appos*), et d'abréviation (*abbrev*) sont autant de liens indiquant un groupe nominal (cf. Figure 3. Hiérarchie des relations grammaticales .).

3.6.3 Les patrons reliés aux groupes verbaux

Les groupes verbaux peuvent être aussi simples que dans le modèle Sujet-Verbe-Objet où ils relient un sujet et un complément d'objet direct, mais ils peuvent aussi être plus complexes et apparaître reliés à des auxiliaires actifs ou passifs ou à des particules «*prt - phrasal verb particle* ». Par exemple, ils peuvent renvoyer à des expressions comme « *is made up* » ou «*to figure out*».

De manière générale, TEXCOMON définit des patrons pour détecter des groupes verbaux et agréger les verbes, leurs auxiliaires, leur négation et leur particule de manière à former une seule et même relation. De plus, lorsqu'il existe une préposition après le groupe verbal, elle est généralement agrégée avec ce groupe en tant que composante de la relation verbale. Par exemple, La figure 11 illustre une partie de l'analyse de la phrase « *an asset can be described with asset metadata* » ainsi que le résultat de la transformation en représentation « sémantique ».




```
det(Asset-2, An-1)
nsubjpass(described-5, Asset-2)
aux(described-5, can-3)
auxpass(described-5, be-4)
nn(metadata-8, asset-7)
prep_with(described-5, metadata-8)
```

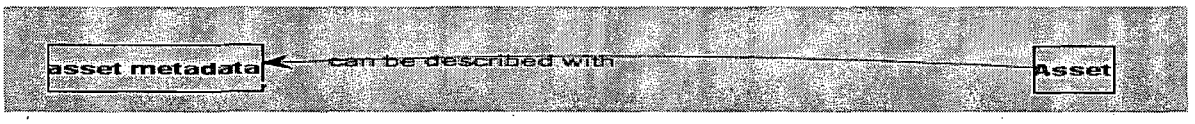


Figure 11. Extraction de relation verbale avec une préposition

Pour obtenir le résultat affiché dans la figure ci-dessus, TEXCOMON a détecté deux patrons de groupes nominaux, à savoir :

nsubjpass(described-5, Asset-2) : Asset est considéré comme un terme du domaine.

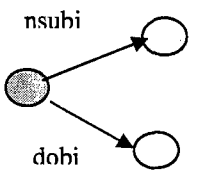
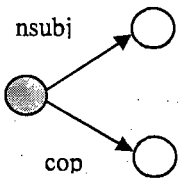
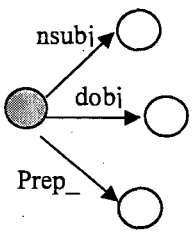
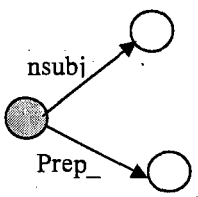
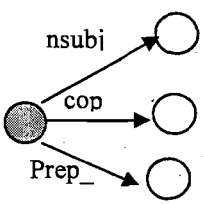
nn(metadata-8, asset-7): ce patron mène à l'agrégation des deux termes reliés par la relation nn

Un patron de groupe verbal a également été détecté à partir des relations grammaticales suivantes :

- aux(described-5, can-3)
- auxpass(described-5, be-4)
- prep_with(described-5, metadata-8)

Ce patron enclenche la transformation automatique de ces 3 triplets en une seule relation : *"can be described with"* qui relie le concept destination de la relation « *nsubjpass* » (soit Asset) et le concept destination de la relation « *prep_with* » (soit « asset metadata »). C'est une fonction java associée à ce patron qui permet l'agrégation des auxiliaires (aux, auxpass) et du verbe auquel ils sont reliés ainsi que l'agrégation de la structure obtenue (auxiliaire + verbe) à la préposition « with » reliée également au verbe. L'ordonnancement de l'agrégation utilise les numéros d'ordre attribués aux différents termes lors de l'analyse grammaticale. Autrement dit, on obtient bien : « *can-3 be-4 described-5*» dans cet ordre.

Nous avons répertorié un ensemble de patrons pour la détection des groupes verbaux, illustrés dans le tableau suivant. A partir d'un terme pivot t, l'algorithme parcourt les patrons pour déterminer si les liens entrants et sortants du terme représentent une configuration de patron connue. Si c'est le cas, l'algorithme déclenche alors l'exécution de la fonction liée au patron. Dans le tableau, la colonne « *Liens sortants* » désigne les relations grammaticales en sortie à partir d'un terme pivot t. La colonne « *patron* » illustre le patron de manière graphique (le rond grisé représente le terme pivot t). Enfin la colonne « *exemple* » indique une phrase du corpus où un tel patron a été retrouvé de même que les relations grammaticales provenant de l'analyse grammaticale de la phrase. Par exemple, la première ligne indique que tout terme t ayant 2 liens sortants de type « *nsubj* » et « *dobj* » indique une structure reconnue comme un patron.

Liens sortants	*Patron*	*Exemple*
"nsubj","dobj"		*An asset might be a text document* nsubj(might-5 be-6, asset-4) dobj(might-5 be-6, text-8 document-9)
"nsubj","cop"		*An asset is a content object* nsubj(content-5 object-6, asset-2) cop(content-5 object-6, is-3)
"nsubj","dobj","prep_"		*A SCO must establish a communication session with the runtime environment* nsubj(must-3 establish-4, SCO-2) dobj(must-3 establish-4, communication-6 session-7) prep_with(must-3 establish-4, runtime-10 environment-11)
"nsubj","prep_"		*SCORM specifies in detail...* nsubj(specifies-3, SCORM-2) prep_in(specifies-3, detail-5)
"nsubj","cop","prep_"		*A shareable content object is a kind of object* nsubj(kind-12, Shareable-2 Content-3 Object-4) cop(kind-12, is-9) prep_of(kind-12, object-15)

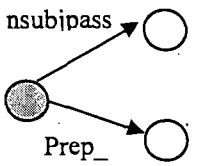
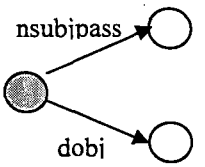
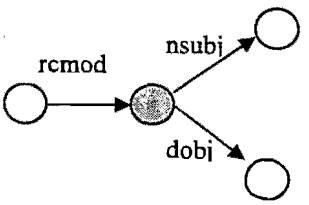
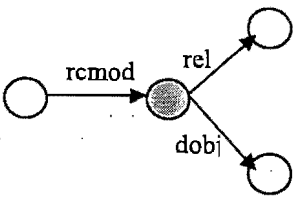
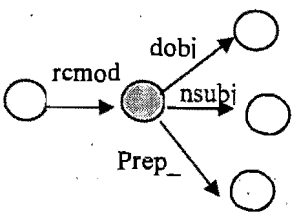
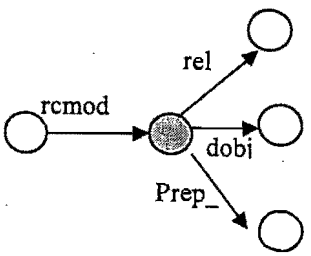
"nsubjpass","prep_"		*An asset can be described with asset metadata* nsubjpass(can-3 be-4 described-5, Asset-2) prep_with(can-3 be-4 described-5, asset-7 metadata-8)
"nsubjpass","dobj"		*Content objects are called shareable content objects* nsubjpass(are-14 called-15, content-2 objects-3) dobj(are-14 called-15, Shareable-16 Content-17 Objects-18)

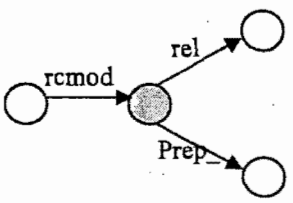
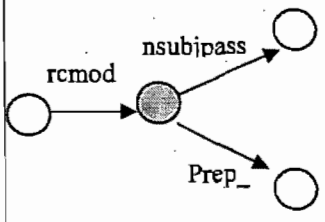
Tableau VI. Quelques patrons de relations verbales avec des liens entrants=null

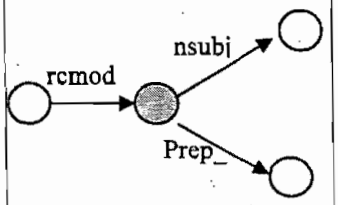
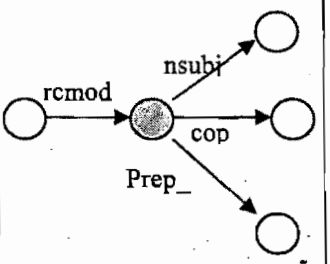
Pour le moment, nous ne considérons que les verbes transitifs. D'autres patrons reliés au traitement des groupes verbaux, constitués cette fois de liens entrants et sortants, ont été définis. Seuls ceux qui ont été validés et aboutissant toujours à des représentations sémantiques correctes ont été retenus. Ce sont ces patrons qui sont introduits dans la section suivante.

3.6.4 Les patrons reliés aux pronoms relatifs (*Relative Clause Modifiers*)

Les patrons de ce type se distinguent par un lien entrant de type rcmod (*Relative Clause Modifier*) qui est introduit par un pronom relatif, à savoir : *that, which, whichever, who, whoever, whom, whomever, et whose* (voir le tableau ci-dessous). Ces pronoms servent à désigner un nom qui les précède.

Liens entrants	*Liens sortants*	*Patron*	*Exemple*
"rcmod"	"nsubj","dobj" Ou "rel", "dobj"	 	*The prescription specifies the activities that use the content objects in the package* nsubj(specifies-3, prescription-2) dobj(specifies-3, activities-5) **nsubj**(use-7, that-6) **rcmod**(activities-5, use-7) **dobj**(use-7, objects-10) nn(objects-10, content-9) prep_in(objects-10, package-13)
"rcmod"	"nsubj", "dobj", "prep_" Ou "rel","dobj","p rep_"	 	*The content objects that can exchange data with the SCORM Runtime Environment are called sharable content objects* nn(objects-3, content-2) **rel**(exchange-6, that-4) aux(exchange-6, can-5) **rcmod**(objects-3, exchange-6) **dobj**(exchange-6, data-7) nn(Environment-12, SCORM-10) nn(Environment-12, Runtime-11) **prep**_with(exchange-6, Environment-12) auxpass(called-14, are-13) amod(objects-17, sharable-15) nn(objects-17, content-16) dobj(called-14, objects-17) nsubjpass(called-14, objects-3)

"rcmod"	"rel", "prep_"		*CBT authoring tools typically provided custom sequencing and navigation features that were encoded in proprietary data formats* **nn**(tools-3, CBT-1) **nn**(tools-3, authoring-2) nsubj(provided-5, tools-3) advmod(provided-5, typically-4) **nn**(sequencing-7, custom-6) dobj(provided-5, sequencing-7) nn(features-10, navigation-9) conj_and(sequencing-7, features-10) auxpass(encoded-13, were-12) **rel**(encoded-13, that-11) **rcmod**(sequencing-7, encoded-13) **prep_in**(encoded-13, formats-17) amod(formats-17, proprietary-15) nn(formats-17, data-16)
"rcmod"	"nsubjpass", "prep_"		*A SCO is a small portable web site that can be copied from place to place* nsubj(site-8, SCO-2) cop(site-8, is-3) amod(site-8, small-5) amod(site-8, portable-6) nn(site-8, web-7) **rcmod**(site-8, copied-12) **nsubjpass**(copied-12, that-9) **prep_**from(copied-12, place-14) aux(copied-12, can-10) auxpass(copied-12, be-11) prep_to(copied-12, place-16)

"rcmod"	"nsubj", "prep_"		*The current version of the IMS Content Packaging specification only defines one form of content organization, which is in the shape of a tree or hierarchy* amod(version-3, current-2) nsubj(defines-11, version-3) nn(specification-9, IMS-6) nn(specification-9, Content-7) nn(specification-9, Packaging-8) prep_of(version-3, specification-9) dobj(defines-11, form-13) nn(organization-16, content-15) prep_of(form-13, organization-16) **nsubj**(is-19, which-18) **rcmod**(organization-16, is-19) **prep_in**(is-19, shape-22) prep_of(shape-22, tree-25) conj_or(tree-25, hierarchy-27)
"rcmod"	"nsubj", "prep_","cop" ;		*Traditional distance training programs emphasize synchronous training technologies that are valuable in providing distance education and training in which students are physically separated from instructors* amod(programs-4, Traditional-1) nn(programs-4, distance-2) nn(programs-4, training-3) nsubj(emphasize-5, programs-4) amod(technologies-8, synchronous-6) nn(technologies-8, training-7) dobj(emphasize-5, technologies-8) nsubj(valuable-11, that-9) cop(valuable-11, are-10) rcmod(technologies-8, valuable-11)

			dep(valuable-11, in-12) dep(in-12, providing-13) nn(education-15, distance-14) dobj(providing-13, education-15) conj_and(education-15, training-17) rel(separated-23, in-18) dep(in-18, which-19) **nsubj**(separated-23, students-20) **cop**(separated-23, are-21) **rcmod**(education-15, separated-23) **prep_from**(separated-23, instructors-25)

Tableau VII. Les patrons constitués de liens entrants de type RCMOD

Certaines contraintes sont appliquées sur les patrons de ce type car les liens sortants de type « rel » ou « nsubj » ne peuvent avoir comme données que des pronoms relatifs (that, which, etc.).

Dans ces patrons, il est également nécessaire de détecter des sujets ou des objets non contigus, et cela fait donc partie du problème de résolution de coréférences.

3.6.5 La résolution des coréférences

La résolution de coréférences indique le processus de détermination de l'entité à laquelle réfèrent deux ou plusieurs expressions distinctes. Nous effectuons une résolution des coréférences dans certains cas, détaillés ci-dessous.

L'extraction des racines de mots permet de relier à un même concept des termes ayant la même racine mais exprimés sous différentes formes. Ces différentes formes peuvent varier en termes de temps, d'accord de genre et de nombre, de synonymes, d'abréviations et d'acronymes, de contractions (exemple *wrt : with respect to*) sans parler des erreurs d'orthographe. Par exemple : «*Stemming*» et «*Stemmer*» ont la même racine «*stem*». Les deux termes renverront à un seul et même concept. Les cartes de concepts sont

sauvegardées avec leurs racines, extraites au moyen d'un algorithme de « *stemming* » : *The Porter Stemmer* (Porter, 1980). L'intérêt de ces racines, pour les concepts et les relations, est de reconnaître les formes dérivées du concept ou de la relation.

TEXCOMON détecte les liens d'abréviation, reconnus par l'analyseur de Stanford sous la forme «*abbrev*» comme désignant une même entité. Pareillement, les liens appositifs (*appositives*) sont reconnus comme désignant la même entité que leur terme source.

La résolution de coréférences permet également, à partir de pronoms relatifs comme « *that* » et « *which* » de retrouver les parties du discours ou entités auxquelles ces mots réfèrent. Il est ainsi possible de retrouver le sujet d'un verbe même lorsque ces derniers ne sont pas contigus dans la phrase. Par exemple : dans la phrase « *An asset is a content object that will not use the SCORM API but can still be used for an activity* », TEXCOMON crée trois relations à partir de cette phrase :

- *is (Asset, Content Object)*
- *can be used for (Asset, Activity)*
- *will not use (Asset, SCORM API)*

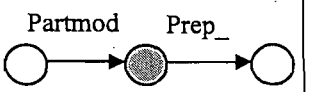
Ce mécanisme peut également être activé lorsqu'un verbe et son objet ne sont pas contigus.

Toutefois, nous ne prenons pas en charge la résolution des pronoms personnels, ni la détection d'anaphores, ce qui est une faiblesse du système actuel. Par exemple, on ne peut traiter la phrase « *It is an essential component of the architecture* ». L'humain peut soit remplacer préalablement ces pronoms dans les phrases (solution pas très pratique, nous en convenons !), soit enrichir manuellement les cartes de concepts résultant de l'analyse.

3.6.6 Les patrons reliés aux participes (*participial modifiers*) et autres structures grammaticales

Pour le moment, un seul patron relatif aux participes (*participial modifier*) a été détecté avec la certitude de donner une représentation sémantique correcte :

Étant donné un terme t.

Liens entrants ***(t)***	***Liens sortants*** ***(t)***	***Patron***	***Exemple***
Partmod	Prep_		*The user can navigate from SCO to SCO through controls provided in the user interface* nsubj(navigate-4, user-2) aux(navigate-4, can-3) prep_from(navigate-4, SCO-6) aux(SCO-8, to-7) xcomp(navigate-4, SCO-8) prep_through(SCO-8, controls-10) **partmod**(controls-10, provided-11) nn(interface-15, user-14) **prep_in**(provided-11, interface-15)

Ce patron est composé d'un lien entrant indiquant un participe et un lien sortant indiquant une préposition (n'importe laquelle). Ce participe est transformé en une relation verbale qui agrège le participe et la préposition. Par exemple, dans la phrase: « *The user can navigate from SCO to SCO through controls provided in the user interface*», TEXCOMON extrait une relation verbale « *controls **provided in** user interface* ».

Un autre patron, constitué d'un lien de dépendance en entrée et de deux liens de sujet et d'objet direct en sortie nous permet de créer une relation verbale, lorsque le sujet

est un pronom tel que « *that* ». TEXCOMON recherche alors le sujet associé au pronom relatif et crée une relation verbale entre ce sujet et le complément d'objet direct. Cette relation a pour label le terme t. Par exemple, dans la phrase : « *An asset is a content object that will not use the SCORM API but can still be used for an activity* », on peut constater que le terme « use » possède cette configuration de liens entrants et sortants. TEXCOMON aboutit à une relation verbale entre asset et SCORM API : «*asset will not use SCORM API*».

Enfin, un dernier patron, avec un lien de préposition en entrée « *prep : prepositional modifier* » et un lien de dépendance en sortie « *dep* » permet à TEXCOMON de créer des relations verbales selon différents cas de figures :

Dans la phrase : « *A SCO can be launched in a web browser by using a url* », TEXCOMON extrait la relation « *by using* » et retrouve l'objet ou le groupe nominal : ici «*web browser*» : « *web browser by using url* ». Notons que cela permet de conserver la structure de la phrase dans la représentation sémantique. Toutefois, il serait probablement plus judicieux de créer deux relations en plus : « *A SCO can be launched in a web browser*» et «*A SCO can be launched by using a url* », ce qui n'est pas fait actuellement.

D'autres patrons ont pu être identifiés par rapport aux clauses de compléments (*ccomp, xcomp*), aux compléments d'adverbes (*advcl*), et aux *purpose clause modifiers* (*purpcl*), mais leur résultat n'a pu être validé sur toutes les phrases testées. Davantage de recherches sont donc requises pour déterminer la représentation sémantique de ces patrons.

3.6.7 Les patrons reliés aux prépositions

Les prépositions peuvent être scindées en prépositions de temps, prépositions après un verbe, après un adjectif et autres. La préposition se place devant le groupe nominal qu'elle introduit (nom, pronom, verbe en –ing).

La majorité des prépositions, comme les prépositions de temps, les prépositions « *with* » et « *for* », etc. sont reprises telles quelles en supprimant le préfixe « *prep _* » et

constituent ainsi une relation à part entière. Par exemple, dans la phrase « *metadata allows for search within online repositories* », la préposition « *within* » sera conservée telle quelle dans la représentation sémantique.

Dans le cas de prépositions après un verbe, celles-ci sont normalement intégrées à la relation constituée de la phrase verbale, comme par exemple : «*assets* ***can be described with*** *metadata*» et sont traitées dans les patrons relatifs aux groupes verbaux.

3.6.8 Les patrons reliés aux conjonctions de coordination

Comme pour les prépositions, les conjonctions sont reprises telles quelles en supprimant le préfixe « *conj _*». Dans le cas de certaines de ces conjonctions, notamment le « *and* » et le « *or* », il est nécessaire de créer des liens qui ne se trouvent pas initialement dans l'arbre de dépendances. Ces nouveaux liens permettent de dédoubler certains liens de dépendances de manière à créer plus d'une relation sémantique. Par exemple, pour la phrase : « *The prescription specifies activities and sub-activities that use the content objects in the package* », on obtient la représentation suivante (Figure 12) :

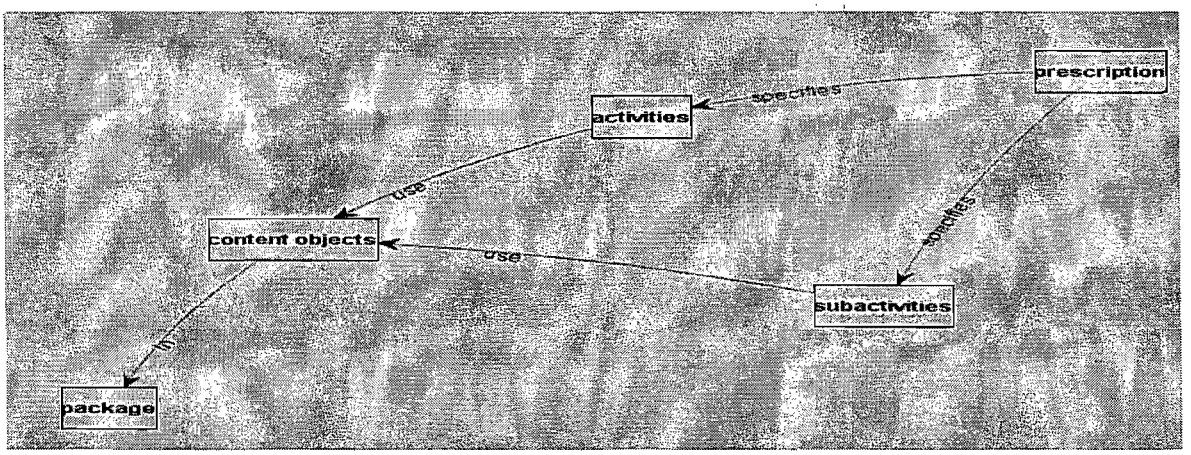


Figure 12. Une représentation avec duplication du verbe lorsqu'une conjonction de coordination "AND" est rencontrée

Dans cette représentation, le lien « and » a été supprimé, deux relations verbales ont été créées : « *prescription specifies activities* » et « *prescription specifies subactivities* » à

la place d'une seule dans l'arbre de dépendances. De la même façon, la relation verbale «*use*» a également été dédoublée.

Si on prend une autre phrase en exemple : « *The runtime environment is typically provided by a LMS, a performance support system or a competency management system* », on obtient la représentation sémantique montrée en (Figure 13). Cela montre le dédoublement résultant d'une conjonction « or ».

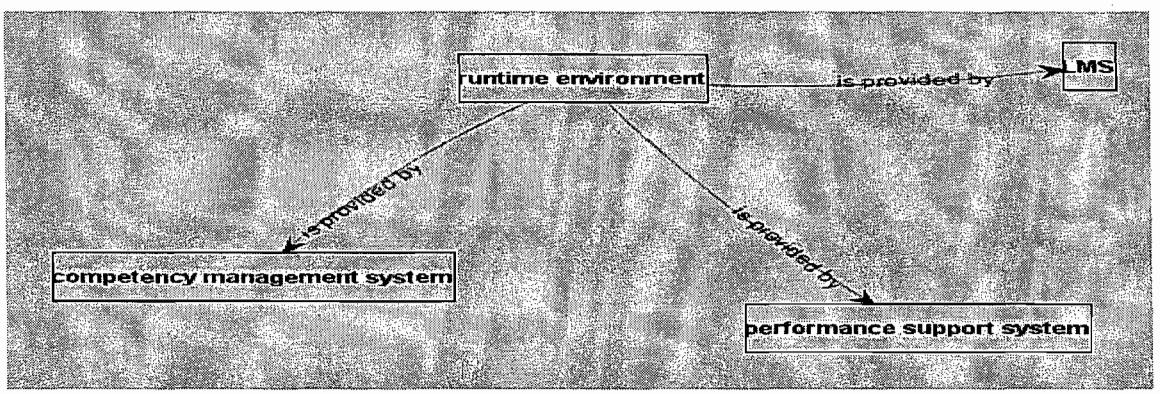


Figure 13. Une représentation avec duplication du verbe lorsqu'une conjonction de coordination "OR" est rencontrée

D'autres types de dédoublements sont pris en charge notamment dans les patrons d'hyponymie tels que :

NP such as NP, NP, ... and NP
Such NP as NP, NP, ... or NP
NP, NP, ... and other NP
NP, especially NP, NP,... and NP

Par exemple, dans la phrase « *media such as text and images* », le lien « *such as* » est dédoublé en sortie du terme « *media*».

3.6.9 Les patrons reliés aux relations spécifiques

De manière générale, certaines relations ont une catégorie clairement déterminée dans la littérature. Il s'agit notamment des relations de composition, de causalité, des liens

définis dans la structure des *qualia* du lexique génératif (Pustejovsky, 1995), d'attributs ou encore des relations hiérarchiques. Certains patrons lexico-syntaxiques indiquent ces relations comme les patrons d'hyperonymie de Hearst (Hearst, 1992). Dans TEXCOMON, ces relations sont traitées au départ comme les autres, c'est-à-dire qu'elles sont détectées par des patrons syntaxiques et ne sont pas catégorisées. Toutefois, il est possible de rajouter un « Analyseur de relations » qui à partir de leurs labels, les catégorisent dans certaines classes prédéfinies. Cela aussi peut être modélisé sous forme de patrons.

3.6.9.1 Les relations de composition

Les relations de composition (*meronyms*) peuvent être indiquées par des relations explicites telles que « *consist of* », « *made of* » ou « *part of* » ou par des expressions plus implicites telles que « *asset's label* », « *label of the asset* », «*the asset has metadata* ». Le problème avec ces constructions, c'est qu'elles peuvent indiquer aussi la possession ou des états. TEXCOMON se charge de convertir de telles constructions en relations verbales, soit en les adjoignant à la préposition comme dans l'expression « *consist of* », soit en convertissant les dépendances résultant d'expressions telles que «*label of the asset*» ou «*asset's label* » par une relation « *has* ».

3.6.9.2 Les relations d'attributs

Les attributs sont des caractéristiques ou propriétés des objets auxquels ils sont associés. TEXCOMON utilise là encore des patrons pour les détecter, notamment des patrons indépendants du domaine, tels que :

- « **X of Y** » => the label of the asset … → The asset *has* label
- « **X' Y** » => assets' metadata→ assets *have* metadata
- « **X's Y** » => Asset's id » → Asset *has* id

Ces patrons sont transformés en une relation « *has* » entre X et Y. Là encore, il est possible d'enrichir ces patrons d'une couche domaine, à savoir des structures linguistiques qui indiquent des attributs liés à un domaine donné. Notons toutefois que de nombreux travaux

ont montré la complexité et les pièges de la modélisation de telles structures en relations de possession ou de composition (*part-whole relationships*) (Artale, Franconi, & Guarino, 1996) (Winston, Chaffin, & Herrmann, 1987). Cette confusion provient de la diversité des types de composition et de la possibilité de les confondre. Dans notre cas, le problème ne se pose pas vraiment puisque là encore, la représentation initiale est destinée à l'humain qui est capable de faire la différence entre les différents types de possession.

3.6.9.3 Les relations causales

Les relations de causes à effets ont été largement étudiées. Pour le moment, nous détectons simplement celles qui utilisent un verbe causal comme « *cause* » ou « *generate* » (Girju & Moldovan, 2002) (Girju, 2003), donc qui sont explicitement indiquées dans le texte.

3.6.10 La détection de sous-classes et d'instances

La détection de sous-classes repose principalement sur les patrons lexico-syntaxiques définis par Hearst précédemment évoqués. La détection d'instances, aussi appelée population d'ontologie, est prise en charge par la détection de patrons d'hyperonymie quand ceux-ci concernent des instances plutôt que des sous-classes (nous opérons une distinction entre la notion de sous-classe et d'instance que nous expliquons en 3.6.14). Par exemple, dans la phrase « ... *media such as text, sound, images, ...* », si *text, sound* et *images* ne sont pas considérés comme des concepts, alors ils sont stockés sous forme d'instances du concept « *Media* ».

Par ailleurs, les adjectifs qualificatifs permettent non seulement de créer des termes composés comme dans «*Intelligent Tutoring System*», mais également de créer une relation taxonomique « *is-a* » entre la phrase nominale et son adjectif. Par exemple « *an Intelligent Tutoring System is-a Tutoring System* ». Un adjectif peut également indiquer un attribut ou une caractéristique comme dans la phrase : « *The system is intelligent* ».

Les différents patrons sont pris en charge par un algorithme de détection de patrons.

3.6.11 L'algorithme de détection de patrons

L'algorithme de détection de patrons est un algorithme de « *pattern matching* » et est exécuté pour chaque phrase clé de chaque document dans le corpus. Cet algorithme nécessiterait l'ajout d'heuristiques pour améliorer son efficacité. Chaque patron est associé à une méthode qui est exécutée à chaque fois que le patron est détecté.

Le pseudo-code de l'algorithme est le suivant. Nous détaillons ici quelques classes méthodes importantes lors de la recherche et de l'exécution des patrons :

- Chargement des patrons
  Pour tous les documents du corpus
  Retrouver les phrases clés
  Pour chaque phrase-clé, lancer l'analyse sémantique et la recherche de patrons

- Analyse sémantique : Effectue la recherche d'une occurrence des patrons dans la phrase courante et déclenche la transformation associée

  Rechercher les patrons terminologiques
  Rechercher les patrons relationnels
  Désagréger les conjonctions de coordination
  Agréger les prépositions
  Sauvegarder l'ensemble

- Recherche des patrons terminologiques : Commence par supprimer les déterminants, rechercher les noms simples et par agréger les noms composés et les groupes verbaux

- Recherche de patrons relationnels : A partir de chaque terme du domaine et des liens grammaticaux (entrants et sortants),

  Parcourir les patrons ;

  Si un patron correspond à la configuration <liens entrants, liens sortants> de terme courant

  Exécuter la méthode de transformation du patron ;

  Ajouter une nouvelle relation

- Désagréger les conjonctions de coordination :
  Rechercher les liens de type « conjonctions de coordination »
  Ex : X and Y
  Créer de nouvelles relations en associant à y les liens entrants et sortants de X et leurs descendants

On pourrait s'étonner de ne pas voir, au niveau de l'algorithme, de gestion des ambiguïtés (lexicales, syntaxiques, sémantiques). Les deux premiers niveaux d'ambiguïté sont gérés par l'analyseur de Stanford qui produit les dépendances typées. Quant-à l'analyse sémantique, elle repose sur des patrons dont la transformation est exempte d'ambiguïté. Cela implique, d'une part, que seules les représentations syntaxiques dont nous « connaissons » la transformation sont traitées et d'autre part, que ce qui n'est pas traité échappe à l'algorithme d'extraction.

La figure suivante (Figure 14) montre l'exécution de l'algorithme sur la phrase « *An asset is a content object that will not use the SCORM API but can still be used for an activity* »

det(asset-2, An-1)
nsubj(object-6, asset-2)
cop(object-6, is-3)
det(object-6, a-4)
nn(object-6, content-5)
nsubj(use-10, that-7)
aux(use-10, will-8)
neg(use-10, not-9)
dep(object-6, use-10)
det(API-13, the-11)
nn(API-13, SCORM-12)
dobj(use-10, API-13)
nsubjpass(used-19, that-15)
aux(used-19, can-16)
advmod(used-19, still-17)
auxpass(used-19, be-18)
conj_but(use-10, used-19)
det(activity-22, an-21)
prep_for(used-19, activity-22)

Etape 1: nsubj(object-6, asset-2) cop(object-6, is-3) nn(object-6, content-5) nsubj(use-10, that-7) aux(use-10, will-8) neg(use-10, not-9) dep(object-6, use-10) nn(API-13, SCORM-12) dobj(use-10, API-13) nsubjpass(used-19, that-15) aux(used-19, can-16) auxpass(used-19, be-18) conj_but(use-10, used-19) prep_for(used-19, activity-22)	**Étape 1** : Suppression des déterminants
Etape 2 : nsubj(content-5 object-6, asset-2) cop(content-5 object-6, is-3) nsubj(will-8 not-9 use-10, that-7) dep(content-5 object-6, will-8 not-9 use-10) dobj(will-8 not-9 use-10, SCORM-12 API-13) nsubjpass(can-16 be-18 used-19, that-15) conj_but(will-8 not-9 use-10, can-16 be-18 used-19) prep_for(can-16 be-18 used-19, activity-22)	**Étape 2** : Agrégation des termes complexes (patrons terminologiques) et des groupes verbaux

Etape 3 : is-3 (asset-2, content-5 object-6) will-8 not-9 use-10 (asset-2, SCORM-12 API-13) can-16 be-18 used-19 prep_for (asset-2, activity-22)	**Étape 3** : Détection de patrons lexico-syntaxiques nsubj(content-5 object-6, asset-2) cop(content-5 object-6, is-3) **→ is-3 (asset-2, content-5 object-6)** nsubj(content-5 object-6, asset-2) nsubj(will-8 not-9 use-10, that-7) dep(content-5 object-6, will-8 not-9 use-10) dobj(will-8 not-9 use-10, SCORM-12 API-13) **→ will-8 not-9 use-10 (asset-2, SCORM-12 API-13)** nsubjpass(can-16 be-18 used-19, that-15) conj_but(will-8 not-9 use-10, can-16 be-18 used-19) prep_for(can-16 be-18 used-19, activity-22) **→ can-16 be-18 used-19 prep_for (asset-2, activity-22)**
Résultat : Is (asset, content object) will not use (asset, SCORM API) can be used for (asset, activity)	**Étape 4** : Suppression des chiffres d'ordre et des préfixes de prépositions « prep_ »

Figure 14. Exécution de l'algorithme de détection de patrons et de construction d'une analyse sémantique

3.6.12 Vers des cartes de concepts

Une fois que l'ensemble des documents est traité, il est possible d'avoir une vue d'ensemble sur les connaissances extraites et de détecter les concepts du domaine. Les concepts sont définis par leur position dans le réseau sémantique (Quillian, 1968). Chaque document peut avoir contribué à la création d'une carte autour d'un concept donné. La création d'une carte sémantique autour d'un concept donné se fait par une simple opération

visant à rassembler les différentes relations extraites des documents du corpus concernant ce concept. En fait, cette opération se fait graduellement lors de l'analyse sémantique. A chaque fois qu'un concept est retrouvé et associé à une relation verbale, cette relation est enregistrée dans un champ « Relation » relié au concept en question. La génération d'une carte autour d'un concept donné se fait donc simplement par l'affichage des différentes relations autour du concept source. Elle permet également de visualiser les cooccurrences de termes et les liens spécifiques qui les relient.

Les figures suivantes montrent les cartes extraites autour des concepts « *runtime environment* » et « *asset* ».

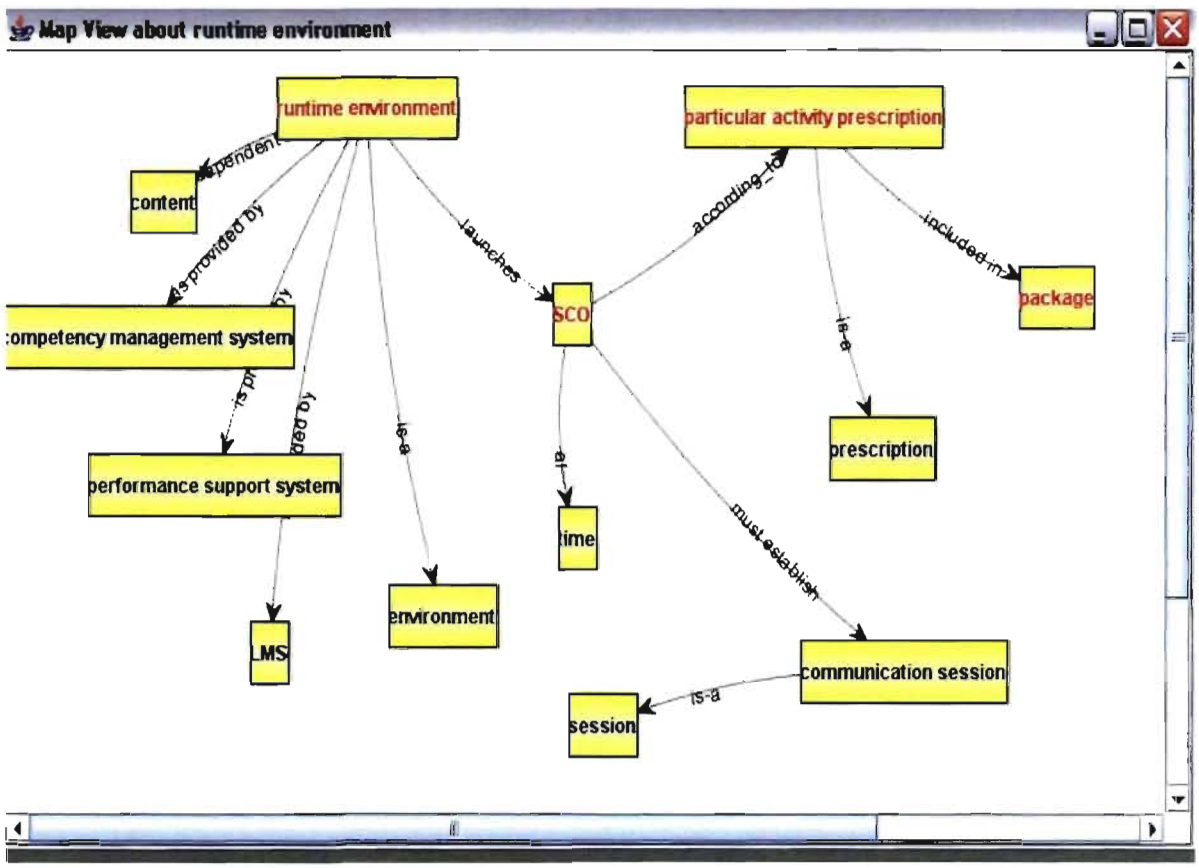


Figure 15. Carte de concepts autour du concept "*runtime environment*"

Ce qui est intéressant dans les cartes de concepts, c'est qu'elles ne se limitent pas à des relations binaires, mais permettent de modéliser des ***chemins d'information***. Ces chemins renvoient à des phrases particulières, à des paragraphes particuliers, et à des documents. Cela offre une structure imbriquée qui permet, à partir d'un concept, de rechercher les phrases, paragraphes et documents qui y réfèrent et à partir d'un document, d'un paragraphe ou d'une phrase, de connaître les concepts et relations sémantiques qu'ils contiennent. Dans la Figure 15, on peut voir un ensemble de nœuds en rouge : ils indiquent le chemin d'information et la manière de lire la carte de concept. Dans ce cas précis, on peut lire : « Runtime environment *launches* SCO *according to* particular activity prescription *included in* package ».

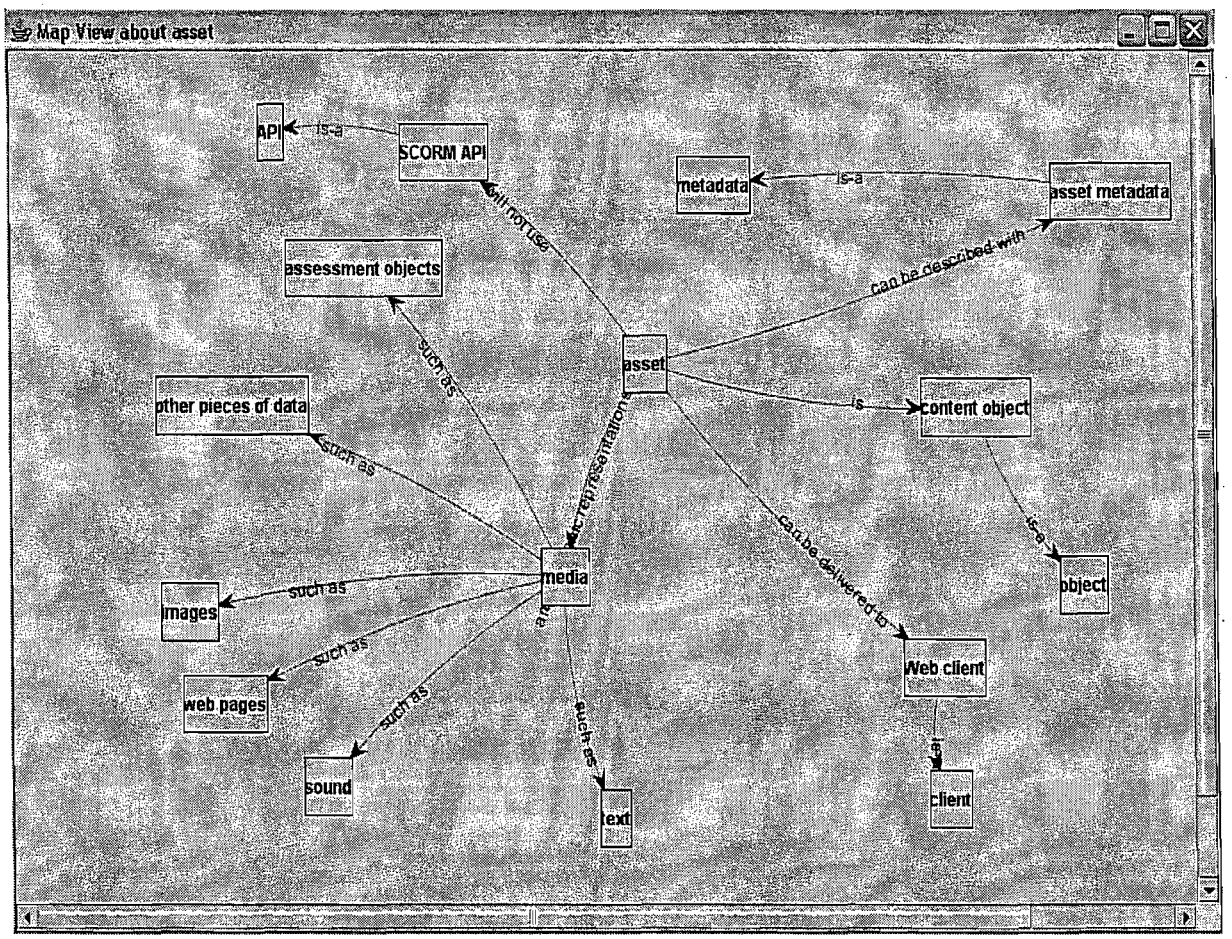


Figure 16. Carte de concepts autour du concept "*asset*"

3.7 La transformation des cartes de concepts en ontologie du domaine

Les cartes de concepts, en tant que telles, peuvent servir à exprimer le modèle du domaine dans un système de formation. Un apprenant peut apprendre simplement en découvrant la notion de proximité sémantique qui découle de la proximité spatiale des concepts et des liens qui les relient. Toutefois, les cartes de concepts n'offrent pas un modèle formel du domaine permettant des inférences et ne constituent pas un standard permettant leur échange et leur interopérabilité avec d'autres systèmes. C'est pourquoi il est intéressant de penser à une transition entre cartes de concepts et ontologie du domaine.

Cette transition n'est pas triviale. En effet, formaliser l'informel de manière automatique peut entraîner des ambiguïtés ou des connaissances erronées. Cette opération doit donc être étroitement surveillée par un expert humain. La démarche consiste concrètement en la détection des concepts et des relations qui ont une nature ontologique (Roche, 2006a) (Soderland & Mandhani, 2007). En d'autres termes, il s'agit de retrouver les concepts et relations ayant le plus de poids par rapport au domaine considéré. A cet effet, TEXCOMON utilise les techniques d'analyse de liens ou plus généralement de réseaux (*Network Analysis*) pour trouver les caractéristiques structurelles du graphe. Cela permet de déduire des informations importantes sur le graphe, telles que les parties du graphe les plus denses, la centralité de tel ou tel concept, etc. Dans le cas de cartes de concept représentant un domaine, il est possible de déduire les concepts et relations les plus représentatifs et d'effectuer certaines déductions pour retrouver des attributs de concepts, en se basant là encore sur des patrons linguistiques.

Dans notre cas, un concept de la carte de concepts est considéré comme ontologique s'il possède un degré sortant (*out-degree*) supérieur ou égal à une valeur paramétrable et définie par l'expert humain. Par exemple, on peut considérer qu'un concept ayant 4 relations sémantiques est suffisamment important pour être retenu dans l'ontologie finale

du domaine. Ce paramètre peut changer en fonction de la taille du corpus ou des desideratas de l'expert qui peut effectuer des tests pour trouver la valeur optimale du paramètre.

Les attributs des concepts sont retrouvés à partir de marqueurs linguistiques (Poesio & Almuhareb, 2005), préalablement détectés lors de l'analyse de patrons. Chaque concept a une carte conceptuelle dont il est le centre. Chacune des relations du concept ont un label qui permet parfois d'indiquer une relation spécifique telle une relation d'attribut par exemple, ou une relation taxonomique (via des liens « is », « are »). Des relations dont le label est « *has* », « *of* » ou « *possess* » peuvent indiquer des attributs lorsque les termes concernés ne sont pas déjà considérés comme des concepts. Par exemple ID dans la phrase « *the asset has an ID* » peut être considéré comme un attribut s'il n'a pas été retenu comme un concept. C'est le même mécanisme qui permet de différencier la notion de sous-classes et la notion d'instances, qui peuvent être toutes les deux représentées par une relation « *is-a* » ou « *is/are* » dans les cartes de concepts. Dans «*A is-a B*», si A n'a pas été considéré comme un concept alors il est converti sous forme d'instance de la classe B. Si A et B sont tous les deux des concepts alors A est considéré comme une sous-classe de B.

Enfin, à part les relations taxonomiques, une relation est considérée comme ontologique si elle relie deux concepts considérés comme ontologiques. Seules les relations verbales sont conservées dans l'ontologie du domaine, les phrases prépositionnelles n'ont pas de sens si elles sont prises hors du contexte de la phrase. Elles ne peuvent donc pas être conservées dans l'ontologie du domaine. Par exemple, la relation « metadata-*within*-online repositories » n'est pas conservée dans l'ontologie (elle l'est pas contre dans la carte de concepts).

Etant donné que le processus aboutit à une structure intégrée où les textes sont reliés aux cartes de concepts qui, à leur tour, sont associées à l'ontologie du domaine, il est possible de naviguer dans ces structures dans un sens ou dans l'autre. A partir de l'ontologie, il est facile de retrouver des cartes de concepts et des portions de textes et inversement.

L'un des langages du Web sémantique est le « *Web Ontology Language* » (OWL) et il permet la déclaration d'un modèle sous forme de logique de description (OWL-DL). C'est au travers de ce langage qu'est exprimée l'ontologie du domaine générée par TEXCOMON. Les concepts sont représentés par des classes OWL (*owl : Class*), les instances sont représentées sous forme d'individus (*OWLIndividual*), les relations verbales sont exportées sous forme de propriétés objets (*OWLObjectProperty*) et les attributs sous forme d'attributs de données (*OWLDataTypeProperty*). Notons que les classes OWL générées sont souvent des classes primitives c'est-à-dire des classes où il n'existe que des conditions nécessaires à l'inverse des classes définies (*Defined Classes*) qui disposent de conditions nécessaires et suffisantes (définition complète). Les seules classes définies qui sont générées par TEXCOMON sont issues de la déclaration d'une relation d'équivalence entre un concept du domaine et son abréviation (par exemple *Learning Management System* et *LMS* sont reconnus comme des classes équivalentes). Par ailleurs, TEXCOMON ne génère pas d'axiomes indiquant que des classes sont disjointes ou utilisant une combinaison booléenne de classes. C'est donc bien une ontologie du domaine incomplète que nous obtenons et que l'expert humain doit valider et enrichir avant de pouvoir l'utiliser comme ontologie du domaine à part entière. Cependant, si les capacités d'inférences ne sont pas vraiment requises et si l'ontologie du domaine doit servir seulement à indexer le contenu des textes, alors il n'est nécessaire que de valider l'ontologie du domaine afin de s'assurer de l'exactitude des concepts et relations générés.

Dans la section suivante, nous présentons une mini-ontologie du domaine générée à partir de trois petits textes issus de notre corpus.

3.8 Un exemple d'ontologie du domaine générée à l'aide de TEXCOMON

Soient les 3 petits textes suivants tirés du corpus sur le standard SCORM. Les mots en gras sont des mots qui ont été retenus comme concepts (ou instances) dans l'ontologie du domaine.

Texte 1:

*A **runtime environment** (**RTE**) must be used to launch the individual **content objects** in a SCORM conformant package. The runtime environment is typically provided by a LMS, a performance support system, or a competency management system. The **learner** interacts with the runtime environment and the web content through a **standard web browser** with JavaScript enabled.*

*The runtime environment is completely independent of the content. However, some parts of it must be constructed in a particular way so that some of the content objects will be able to exchange data with the runtime environment. Typically, the runtime environment is split across a network connection, with **parts of it** on a server and part of it running in the user's browser.*

Texte 2:

*An **asset** is a content object that will not use the SCORM API but that can still be used for an activity. For example, it might be a text document or an image. Assets are electronic representations of **media** such as **text, images, sound, web pages, assessment objects, or other pieces of data** that can be delivered to a Web client. An Asset can be described with **asset metadata** to allow for search and discovery within online repositories, thereby enhancing opportunities for reuse.*

Texte 3:

The content objects that can exchange data with a SCORM conformant runtime environment are called ***Shareable Content Objects (SCOs)****. The runtime environment launches the SCOs one at a time, according to* ***a particular activity prescription*** *included in the package. Unless the activity prescription forbids it, the user can also navigate from SCO to SCO through controls provided in the runtime environment's user interface.*

The SCORM specifies in detail how a SCO must behave within the runtime environment. The SCO must establish a ***communication session*** *with the runtime environment, and there is a standard set of data elements that the SCO can use during the communication session. This includes tracking data that allows the SCO to report success and progress, as well as other information about the status of content objectives, results of interactions, and so on.*

A Shareable Content Object, or SCO, is a special kind of content object that knows how to communicate with the runtime environment in which it is launched. A SCO is ***web content****, meaning that it can be launched in a web browser by using a URL. It may consist of a single HTML page, or it may be a large collection of web pages and include simulations, Flash assets, or other media rich content. A SCO is basically a* ***small portable web site*** *that can be copied from place to place by gathering all its files and capturing them in a SCORM package.*

To be portable, a SCO must be compatible with any generic web server. In other words, it cannot depend on special services that might exist on one web server but not on another.

Même si ces trois textes représentent un « corpus » extrêmement restreint, il est intéressant de disposer d'une ontologie du domaine générée à partir de ces trois textes seulement. Cela donne une idée de ce que permet actuellement TEXCOMON. La Figure 17. Une vue des classes de la mini ontologie générée par TEXCOMON.

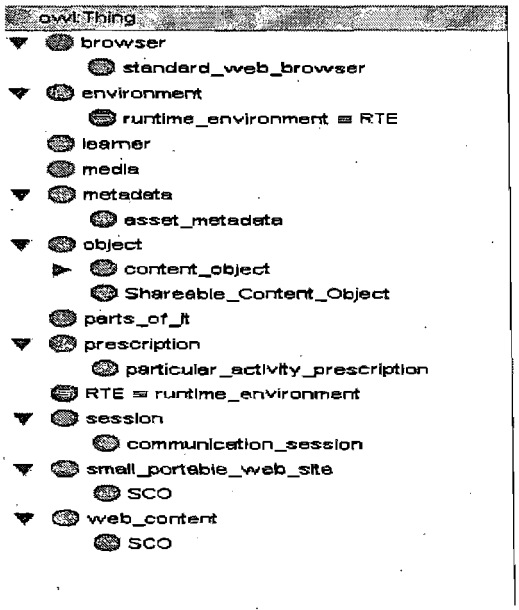


Figure 17. Une vue des classes de la mini ontologie générée par TEXCOMON

La figure suivante illustre les différents liens taxonomiques qui existent entre les classes de l'ontologie.

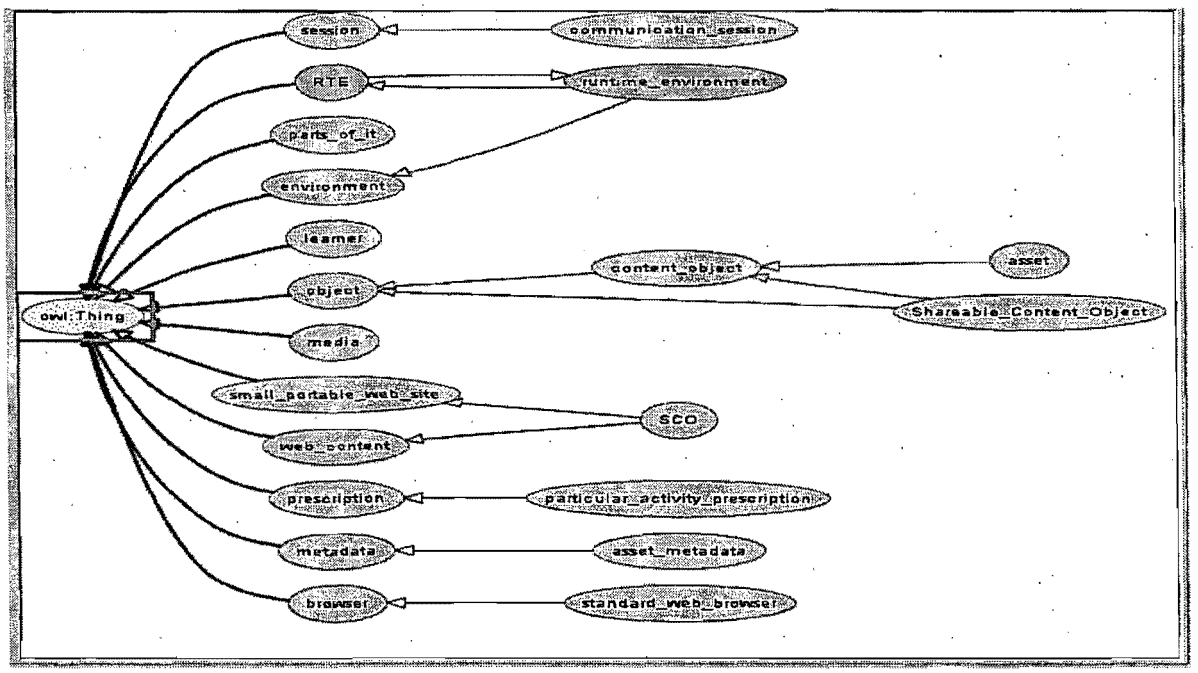


Figure 18. Vue des classes et sous-classes dans OWLVizTab

Enfin, la figure 19 montre l'ensemble des rôles entre concepts (liens en bleu), des relations taxonomiques (liens *isa* noirs) et les instances trouvées dans les textes (liens rouges).

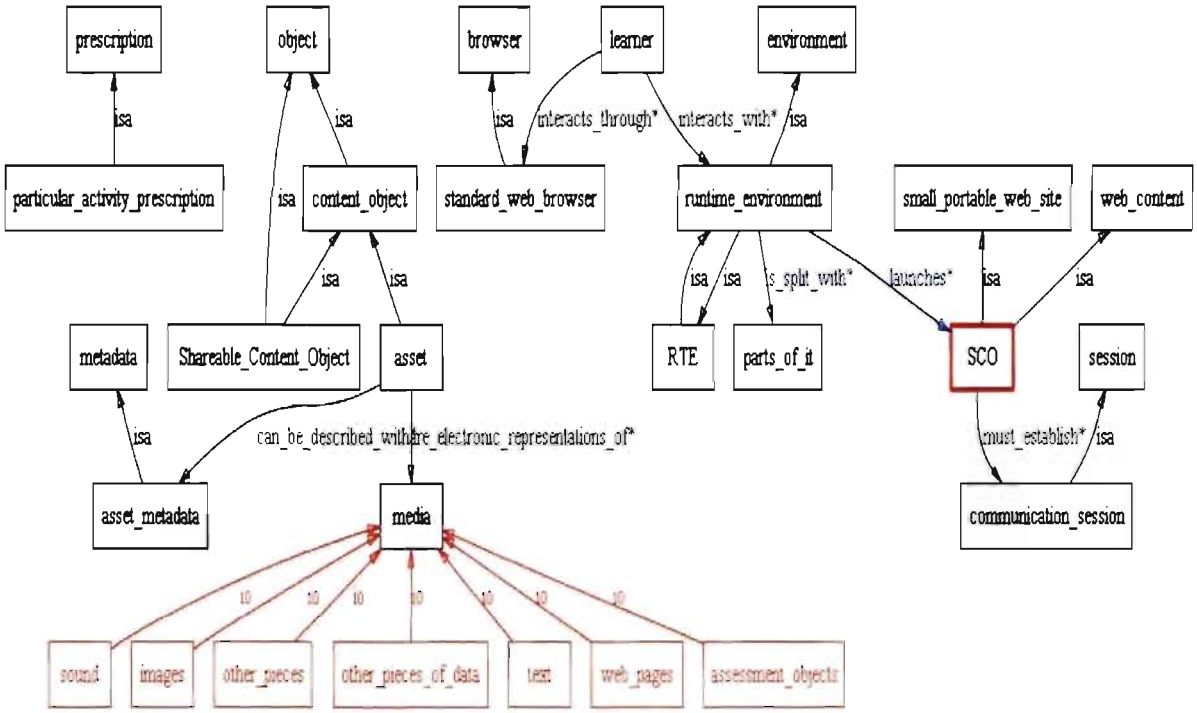


Figure 19. Vue graphique de l'ontologie dans Ontoviz

Bien que n'ayant été testé en profondeur que sur un seul corpus, TEXCOMON est capable de générer une ontologie du domaine à partir de tout corpus contenant les patrons lexico-syntaxiques identifiés.

3.9 En résumé

Nous avons présenté une méthodologie d'ingénierie pour l'acquisition d'une ontologie du domaine à partir de textes. Cette approche est concrétisée par l'outil TEXCOMON, une suite logicielle qui se base essentiellement sur des patrons linguistiques pour l'extraction de connaissances à partir de textes. Notre approche est différente de l'existant sur au moins deux points : elle utilise les structures syntaxiques du langage et les

dépendances typées pour l'extraction de cartes de concepts, ce qui n'a jamais été fait, à notre connaissance, de manière aussi complète, les travaux en extraction d'information se focalisant sur les relations verbales simples. Ensuite, notre modélisation de patrons associés à des fonctions permettant de transformer les cartes de concepts grammaticales en représentations sémantiques est réutilisable. Nous avons donc mis au point une base de connaissances de patrons qui peut être enrichie, par la suite, par des patrons plus complexes en partenariat avec des linguistes. Elle peut également bénéficier de techniques d'apprentissage machine pour la génération de patrons. Cela offre à notre méthode une certaine flexibilité puisqu'il « suffit » d'ajouter une fonction relative à un nouveau patron pour lui permettre de créer des représentations sémantiques. Par ailleurs, notre approche se caractérise par une méthode non supervisée et indépendante du domaine. Cette approche en couches n'empêche toutefois pas l'adjonction d'une couche de patrons reliés au domaine. La connaissance à extraire n'est pas connue a priori mais bien découverte au fur et à mesure de l'extraction. Ainsi, aucune définition préalable n'est requise (hormis celle réservée à la langue utilisée) à partir d'ontologies, de taxonomies, ou d'entités nommées par exemple.

De plus, en sus des méthodes utilisées, c'est plutôt dans la démarche du projet qu'il faut chercher son originalité : TEXCOMON génère d'abord des cartes de concepts, converties ensuite en ontologie du domaine. L'originalité de cette démarche se situe aussi dans sa finalité : son application au domaine de la formation lui permet d'utiliser les structures générées comme un tout intégré : les documents renvoient aux cartes de concepts qui sont à leur tour raffinées en ontologie du domaine et inversement, chaque structure est indexée par la structure précédente.

L'intérêt du passage par les cartes de concepts au niveau de cette thèse est double :

Tout d'abord, les cartes de concepts constituent un moyen d'améliorer l'apprentissage actif (*meaningful*) constructiviste. Elles représentent la connaissance d'un domaine de manière graphique et schématique et sont plus faciles à retenir qu'un texte. De nombreux travaux ont prouvé l'intérêt de l'utilisation de cartes de concepts en formation

(Kumar, 2006) (Novak & Cañas, 2006). Toutefois, la majorité des travaux s'appuyaient sur des approches manuelles de construction (Kumar & Kahle, 2006). Les quelques travaux qui ont adopté la génération dynamique de cartes de concepts à partir de textes se sont limités à des patrons «sujet-verbe-objet » tels que (Clariana & Koul, 2004) (Valerio & Leake, 2006). Les solutions que nous avons proposées dans le chapitre 3 ont donc enrichi ces approches en permettant la prise en compte de patrons beaucoup plus nombreux plus à même de modéliser une plus grande partie du domaine. Enfin, en ce qui concerne le passage de cartes de concepts en ontologies du domaine, il est surprenant de constater que très peu de travaux ont établi un pont entre ces deux types de structures. A notre sens, ces approches figurent sur une même échelle sémantique mais à des degrés de formalismes très différents. Le travail de (Hayes, Eskridge, Saavedra, Reichherzer, Mehrotra, & Bobrovnikoff, 2005) (Eskridge, Hayes, Hoffman, & Warren, 2006) a toutefois avancé l'idée de patrons du Web sémantique permettant de faire le pont entre des structures sous forme de cartes de concepts et une ontologie du domaine.

Ensuite, contrairement à l'ontologie qui ne peut modéliser que des relations en forme de triplets, les cartes de concepts expriment des « ***chemins d'information*** » qui permettent de compléter, préciser et enrichir la relation initiale. Les cartes de concepts peuvent donc se lire comme un texte schématique.

L'ontologie du domaine peut ensuite être utilisée dans la formation par ordinateur mais également comme support à n'importe quel système à base de connaissances.

Un tel processus d'ingénierie doit toutefois se doter de mécanisme de contrôle de la qualité des ontologies produites et des ressources qui en résultent. L'évaluation d'ontologies, quelles soient automatiquement générées ou pas, est encore un sujet de recherche qui n'a pas atteint sa maturité. Certaines solutions parcellaires ont été trouvées, mais aucune approche intégrée satisfaisante n'a été proposée. Nous pensons qu'une combinaison de techniques devrait permettre des résultats plus probants mais qu'un expert humain doit normalement faire partie du processus afin de valider les décisions du système.

Dans la prochaine section, nous proposons une évaluation en trois niveaux de l'ontologie du domaine, qui, si elle n'atteint pas encore les critères d'une approche globale, combine plusieurs aspects d'évaluation.

4 Validation de l'ontologie du domaine produite par TEXCOMON

Ainsi que nous l'avons précédemment souligné, l'évaluation d'une ontologie est une étape critique d'autant plus lorsqu'on recourt à des systèmes de génération automatique ou semi-automatique du domaine. Chaque étape d'extraction doit pouvoir être évaluée (termes, concepts, taxonomie, relations non taxonomiques).

De manière générale, l'évaluation d'une ontologie doit permettre de répondre aux questions suivantes :

1. l'ontologie modélise-t-elle correctement le domaine ? quels sont les critères qui nous permettent de décider si une ontologie modélise correctement le domaine (c'est-à-dire quels critères permettent de confirmer ou d'infirmer le bien-fondé d'un concept ou d'une relation apparaissant dans l'ontologie) ?
2. l'ontologie modélise-t-elle entièrement le domaine (c'est-à-dire qu'elle comprend tous les concepts et relations nécessaires à la description du domaine) ? ou le modélise-t-elle de façon satisfaisante par rapport à l'application visée (c'est-à-dire que même si l'ontologie ne recouvre qu'une connaissance partielle du domaine, elle est suffisante compte tenu des objectifs de l'application qui l'exploite) ?
3. quels sont les critères qui nous permettent de comparer l'intérêt d'une ontologie par rapport à une autre ?

Pour l'évaluation des ontologies générées par TEXCOMON, nous proposons une approche d'évaluation basée sur trois dimensions : la dimension structurelle, la dimension sémantique et la dimension comparative (Zouaq, Nkambou, & Frasson, 2007c).

L'évaluation structurelle essaie de détecter les caractéristiques structurelles de l'ontologie du domaine. Basées sur un ensemble de métriques définies par (Alani &

Brewster, 2006), ces caractéristiques peuvent aider un concepteur d'ontologies à déterminer si l'ontologie correspond à ses besoins.

L'évaluation sémantique fait appel aux experts du domaine pour évaluer la qualité de l'ontologie générée et la plausibilité de ses concepts et relations.

L'évaluation comparative permet de comparer les résultats de différents outils de génération d'ontologies sur le même corpus de documents. Dans notre cas, nous avons choisi de comparer notre plateforme TEXCOMON avec TEXT-TO-ONTO (Maedche & Staab, 2000c), un des outils les plus cités dans le domaine de la génération d'ontologies à partir de textes. Bien que les algorithmes utilisés dans TEXCOMON diffèrent substantiellement de ceux de TEXT-TO-ONTO, les deux produisent des résultats que l'on peut comparer.

4.1 Description du corpus

Le corpus de documents est composé de manuels sur le standard SCORM (SCORM, 2007). Il contient 36 fichiers txt totalisant dans leur ensemble 30 000 mots. Ce corpus a été manuellement préparé en excluant les exemples de code et certaines constructions telles que «*Refer to Section* ». Nous avons également éliminé certaines structures telles que « terme : définition » en les remplaçant par une structure « terme + verbe au début de la définition » lorsque cela s'appliquait. Par exemple, la structure « *Scorm Runtime environment : defines....* » est convertie en « *Scorm Runtime environment defines...* ». D'autres transformations sur le corpus initial incluent d'entourer des chiffres tels que « 1.2 » de guillemets, de remplacer « e.g. » par « such as », ou de supprimer les tirets comme dans « *Run-Time* » (voir l'Annexe A concernant ce dernier point).

Afin de déterminer si les résultats se vérifiaient d'un corpus à l'autre, nous avons créé 7 mini-corpus à partir des 36 documents de départ, ces corpus étant imbriqués au fur et à mesure. Par exemple, le corpus 1 est constitué de 10 documents, auxquels on rajoute 4 documents dans le corpus 2 et ainsi de suite. Cela permet d'évaluer l'évolution de la valeur

des différentes métriques lorsque de nouveaux documents sont ajoutés au corpus précédent. La table suivante donne quelques statistiques sur ces corpus.

Corpus	*Nombre de fichiers*	*Nombre de paragraphes*	*Nombre de phrases*
Corpus 1	10	76	728
Corpus 2	14	85	781
Corpus 3	18	104	921
Corpus 4	22	121	1086
Corpus 5	26	144	1294
Corpus 6	30	169	1450
Corpus 7	36	188	1578

Tableau VIII. Quelques statistiques sur les différents corpus utilisés

Le tableau suivant montre également les tailles des différentes cartes de concepts créées à partir des corpus successifs. Cette taille se mesure en nombre de concepts et de relations.

Corpus	*Nombre de concepts*	*Nombre de relations*
Corpus 1	671	1052
Corpus 2	725	1149
Corpus 3	792	1270
Corpus 4	884	1445
Corpus 5	960	1622
Corpus 6	1076	1824
Corpus 7	1139	1973

Tableau IX. Statistiques sur la taille des cartes de concepts

Le tableau ci-dessous montre également les temps de l'analyse linguistique (temps de traitement) et les temps de sauvegarde de la structure des documents et des cartes de concepts. Ces tests ont été effectués sur un processeur *intel centrino* de 1.86 Ghz et une mémoire vive de 1 GB.

Corpus	*Temps de traitement (mn)*	*Temps de sauvegarde (structure des documents) (mn)*	*Temps de sauvegarde (Cartes de concepts) (mn)*	*Temps Total (mn)*
Corpus 1	15.78	35.32	72.83	124
Corpus 2	2.05	7.63	3.65	13.33
Corpus 3	2.2	14.45	6.92	23.57
Corpus 4	3.25	16.98	11.05	31.28
Corpus 5	4.57	32.87	17.27	54.71
Corpus 6	3.05	20.6	11.6	35.25
Corpus 7	3.17	14.18	6.85	24.2

Tableau X. Temps d'analyse et de sauvegarde pour les différents corpus

Nous pouvons remarquer que les temps de sauvegarde sont particulièrement longs. Cela est dû à l'utilisation d'un projet de type « base de données » dans Protégé. Les tests effectués avec un projet de type « file » étaient beaucoup plus rapides. Il semble donc que le temps total de l'algorithme pourrait être considérablement réduit avec une utilisation directe d'une base de données (sans passer par Protégé) ou par l'amélioration des performances de l'environnement Protégé en mode base de données.

La prochaine section décrit l'expérimentation effectuée sur les ontologies de TEXCOMON.

4.2 Description de l'expérimentation avec TEXCOMON

L'expérimentation consiste en la démarche suivante : l'objectif est de savoir si les ontologies générées représentent un domaine tel que décrit par des mots-clés préalablement choisis par des experts du domaine. Les critères choisis pour mesurer la représentativité de l'ontologie par rapport à ces mots-clés sont les suivants :

- les mots-clés existent comme classes dans l'ontologie ;
- ces classes sont structurellement proches les unes des autres ;

- ces classes sont richement décrites ;
- ces classes sont interconnectées via de multiples relations ;
- ces classes occupent une place centrale dans l'ontologie (l'ontologie est considérée comme un graphe).

Le tableau XI montre les mots-clés qui ont été choisis comme étant des concepts représentatifs dans le domaine du standard SCORM.

Mots-clés
Asset
SCO
SCORM Content Model
SCORM
LMS
Runtime Environment
Metadata
SCORM Content Packaging
Activity
Content Organization
API
PIF

Tableau XI. Les mots-clés représentatifs du domaine

Dans l'approche TEXCOMON, ainsi qu'expliqué précédemment, l'expert doit assigner une valeur à un paramètre I qui constitue le degré sortant d'un concept. Durant l'expérimentation, nous avons généré 4 ontologies du domaine à partir d'un même corpus. Ces ontologies correspondent à différentes valeurs pour le paramètre I en question. Autrement dit :

- KP-2 représente l'ontologie générée avec le paramètre I=2. Rappelons que ce paramètre indique le nombre de relations sortantes d'un terme du domaine. Ce nombre de relations permet d'indiquer que le terme en question est suffisamment important pour être considéré comme concept.

- KP-4 représente l'ontologie générée avec le paramètre I=4.
- KP-6 représente l'ontologie générée avec le paramètre I=6.
- KP-8 représente l'ontologie générée avec le paramètre I=8.

La première partie de l'expérimentation a consisté en l'analyse d'un ensemble de métriques et le calcul d'un score pour les ontologies générées. La seconde partie de l'expérimentation a permis de calculer les mêmes mesures sur les ontologies générées par TEXT-TO-ONTO et de comparer les résultats. Enfin, une analyse sémantique, effectuée par les experts, vient achever le cycle de validation de l'ontologie.

4.3 Analyse structurelle

L'évaluation structurelle se base sur un ensemble de métriques définies par (Alani & Brewster, 2006). A l'origine, ces métriques ont été développées pour évaluer les ontologies retournées par un moteur de recherche et leur assigner un score et un rang en fonction du besoin exprimé par la requête et de leur capacité à satisfaire ce besoin. En fonction d'un ensemble de mots-clés, l'objectif de (Alani & Brewster, 2006) est de retrouver la meilleure ontologie, c'est-à-dire celle qui soit la plus représentative des mots-clés recherchés.

Nous avons modifié quelque peu la vision initiale de l'approche. Dans notre cas, l'idée n'est pas de retrouver des ontologies en fonction de mots-clés mais de définir un ensemble de termes qui soient représentatifs du domaine et de vérifier si ces termes se retrouvent bien dans l'ontologie générée et si, en tant qu'objets ontologiques, ils disposent de certaines caractéristiques recherchées : interconnexion, richesse de description, etc.

Les différentes métriques impliquées dans l'analyse structurelle de l'ontologie sont : « *Class Match Measure (CMM)* », « *Density Measure (DEM)* », « *Betweeness Measure (BEM)* » et « *Semantic Similarity Measure (SSM)* ». Un score total est calculé à partir de la

valeur de ces métriques. Ce score peut ensuite être utilisé pour assigner un rang aux ontologies en fonction des mots-clés recherchés.

Nous avons implanté une librairie de fonctions (la librairie ONTO-EVALUATOR) qui encodent les formules des métriques ainsi que définies dans (Alani & Brewster, 2006). Nous avons également utilisé la librairie JUNG (Java Universal Network/Graph Framework, 2007) pour calculer certaines valeurs notamment la métrique *Betweeness*.

La section suivante aborde la première métrique utilisée dans l'analyse structurelle.

4.3.1 La métrique « *Class Match Measure* »

La métrique « *Class Match Measure* » ou CMM évalue le degré de recouvrement des mots-clés recherchés par l'ontologie. En fonction des mots-clés, ONTO-EVALUATOR parcourt les classes de l'ontologie pour voir si leurs labels correspondent totalement ou partiellement à ces mots-clés.

Soit C[o] un ensemble de classes dans l'ontologie o, et T l'ensemble des termes recherchés. Label(c) représente le label associé à une classe c. Par exemple, on cherche à savoir si le label « *asset* » se retrouve dans les labels des classes de l'ontologie. Une classe de label « *asset* » indique un recouvrement total du mot-clé recherché, une classe de label « *asset metadata* » indique un recouvrement partiel. Le fait de rechercher un recouvrement total ou partiel correspond au besoin de savoir si un des mots-clés recherchés figure en tant que classe de l'ontologie (ce qui est le but recherché ici) ou s'il figure au moins partiellement dans les labels des classes. Ce recouvrement partiel indique que les classes de l'ontologie réfèrent au moins partiellement au mot-clé recherché. Les formules suivantes permettent de calculer un total du nombre de mots-clés partiellement ou complètement retrouvés dans l'ontologie.

$$E(o,T) = \sum_{c \in C[o]} \sum_{t \in T} I(c,t) \qquad I(c,t) = \begin{cases} 1 : si\ label\ (c) = t \\ 0 : si\ label\ (c) \neq t \end{cases}$$

$$P(o,T) = \sum_{c \in C[o]} \sum_{t \in T} J(c,t) \qquad J(c,t) = \begin{cases} 1 : si\ label\ (c)\ contient\ t \\ 0 : si\ label\ (c)\ ne\ contient\ pas\ t \end{cases}$$

E(o,T) et ***P(o,T)*** sont le nombre de classes de l'ontologie o qui possèdent des labels qui correspondent respectivement exactement ou partiellement aux termes recherchés T.

La métrique CMM se calcule de la façon suivante :

$$CMM(o,T) = \alpha E(o,T) + \beta P(o,T)$$

Un poids (α et β) égal ou différent peut être assigné à ces recouvrements partiels et globaux. Ainsi, on peut décider que seules les classes dont le recouvrement est exact nous intéressent ou moduler la valeur du poids si les deux sont à prendre en compte. Par exemple, si le mot-clé recherché est « asset » mais que l'ontologie ne contient que la classe « asset metadata », alors un recouvrement exact donnera un CMM à 0 alors qu'un recouvrement partiel considérera qu'il y a un concept proche de « Asset » qui figure dans l'ontologie et donnera une meilleure valeur CMM.

La figure suivante montre l'évolution de la valeur de CMM dans les ontologies générées par TEXCOMON (KP-2, KP-4, KP-6 et KP-8) sur les différents corpus. Rappelons que KP-2, KP-4, KP-6 et KP-8 sont les ontologies générées par TEXCOMON en affectant une valeur différente au nombre de relations nécessaires I pour considérer qu'un terme du domaine est bien un concept du domaine. Le chiffre accolé à « KP » indique ce nombre.

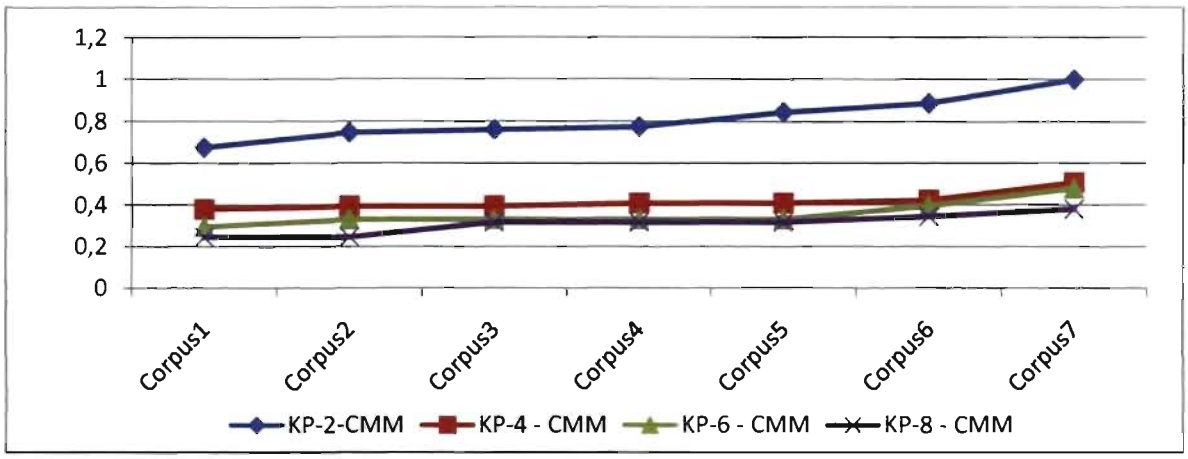


Figure 20. Évolution de la valeur de la métrique CMM sur les différents corpus

Sur un même corpus, on constate que le CMM tend à devenir meilleur lorsque la valeur du paramètre I (2, 4, 6,8) décroit dans un même corpus. On peut ainsi noter que KP-2 obtient une meilleure valeur CMM. Cela démontre que plusieurs concepts qui contiennent les mots-clés (partiellement ou totalement) sont supprimés lorsque le seuil augmente, ce qui peut causer des problèmes si ces concepts sont vraiment importants pour le domaine. Nous avons tenté une seconde expérimentation en ne tenant compte que des concepts dont les labels équivalaient totalement aux mots-clés (nous avons supprimé les recoupements partiels). Nous avons obtenu un graphe totalement différent (Figure 21).

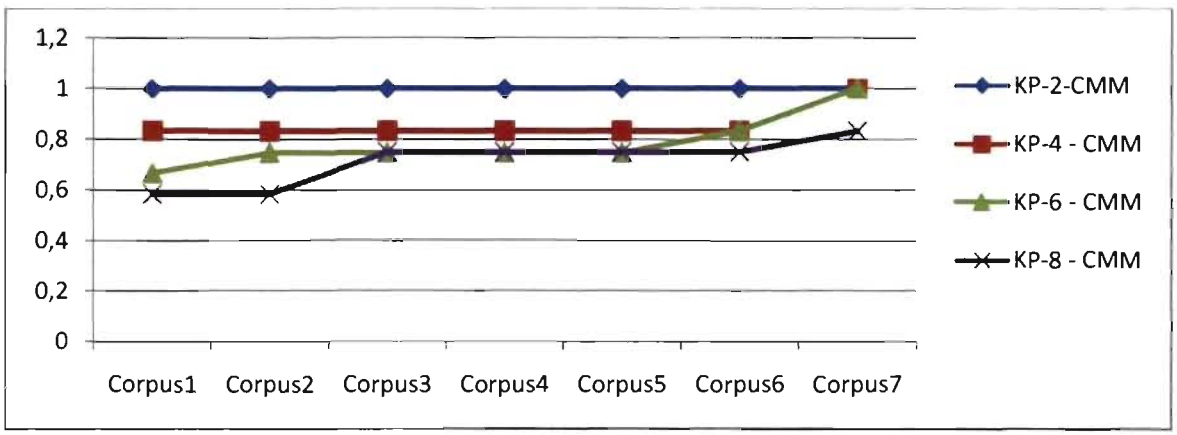


Figure 21. Évolution de la valeur de la métrique CMM avec des recoupements complets des labels

On peut ainsi noter que dans le dernier corpus (le plus riche), KP-2, KP-4 et KP-6 ont des résultats similaires. L'ontologie KP-8 est toutefois moins performante. Cela indique qu'en général, les concepts importants du domaine ont jusqu'à 7 relations sortantes qui les relient à d'autres concepts. Fixer la valeur à 8 est donc trop élevé étant donné le corpus en entrée. On peut donc en conclure que le fait de considérer les recoupements exacts et/ou partiels peut avoir un impact non négligeable, et que les autres métriques peuvent en être affectées : en effet, dans toutes les métriques suivantes, les résultats sont divisés par le nombre de classes retrouvées dans le CMM.

4.3.2 La métrique «*Density Measure*»

La métrique de densité (DEM) exprime le degré de détail associé à une classe ou la richesse de ses attributs. Cette mesure repose sur l'idée qu'une bonne représentation de concept doit être suffisamment détaillée. Par degré de détail, on entend le nombre de sous-classes, le nombre de superclasses, le nombre de voisins et le nombre de relations avec d'autres classes. La définition suivante est fournie par (Alani & Brewster, 2006) :

Etant donnée une classe c. Soit S= {S_1, S_2, S_3, S_4} = {relations[c], superclasses[c], sous-classes[c], voisins[c]}. Pour calculer la densité de la classe c, on compte le nombre de relations, de superclasses, de sous-classes et de voisins et on affecte un facteur de poids Wi (valeur par défaut à 1) à chacun de ces attributs. Cela est effectué par la formule suivante :

$$\boldsymbol{dem(c) = \sum_{i=1}^{4} Wi * cardinalité(Si)}$$

La densité d'une ontologie o correspond alors à la somme des densités des classes c divisée par $\boldsymbol{n=E(o,T)+P(o,T)}$, c'est-à-dire le nombre de classes appariées (avec les termes recherchés) dans l'ontologie $\boldsymbol{o}$ (cette valeur est calculée dans la première métrique CMM).

$$\boldsymbol{DEM(o) = 1/n \sum_{i=1}^{n} dem(c)}$$

La figure 22 indique l'évolution de la valeur du DEM des différentes ontologies sur les différents corpus.

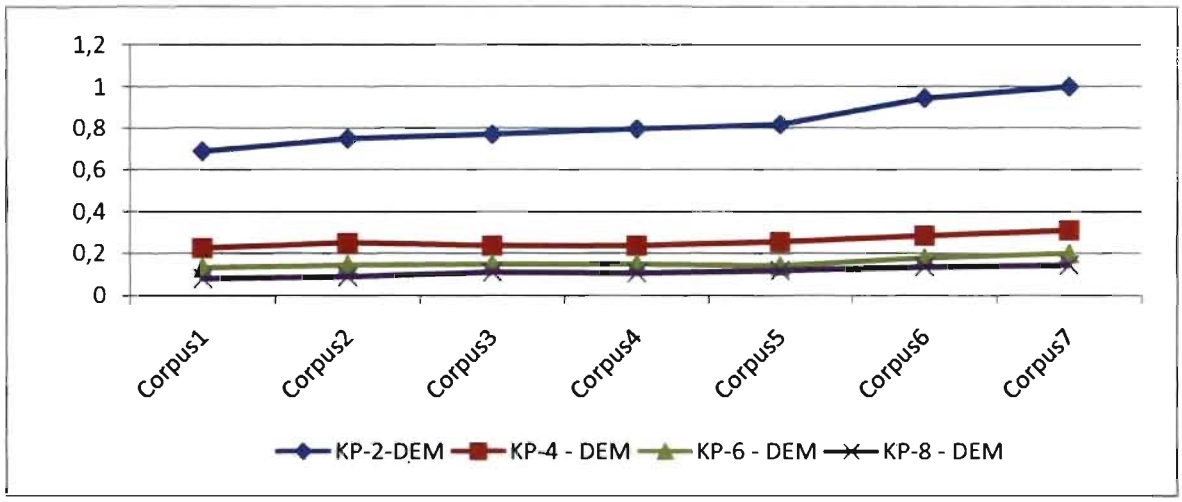


Figure 22. Évolution de la valeur de la métrique DEM sur les différents corpus

On constate que lorsque le nombre de concepts croit, le DEM tend à croitre également. Cette augmentation résulte des nouvelles informations apportées par les documents rajoutés au corpus précédent. Une telle métrique peut ainsi permettre de vérifier la contribution plus ou moins importante de documents à l'ontologie. Par exemple, les corpus 6 et 7 contiennent probablement de nouvelles relations, ce qui explique une augmentation du DEM entre le corpus 5 et le corpus 6 (spécialement pour KP-2).

4.3.3 La métrique « *Semantic Similarity Measure* »

La métrique « *Semantic Similarity Measure* » ou SSM calcule la proximité des classes qui correspondent aux mots-clés recherchés. Si les mots-clés donnés en entrée sont bien représentatifs du domaine, une ontologie bien constituée doit faire état de plusieurs relations entre les classes correspondant à ces mots-clés, qu'elles soient taxonomiques ou pas. Dans le cas inverse, cela peut indiquer un manque de cohésion dans la représentation de la connaissance du domaine. La définition suivante indique comment calculer la métrique (Alani & Brewster, 2006) :

Soient *Ci et Cj* $\in$*{classes[o]}* et *p(Ci* $\rightarrow$ *Cj)* est un chemin $p \in P$

P est l'ensemble des chemins entre les classes *Ci* et *Cj*.

$$ssm(Ci,Cj) = \begin{cases} 1/longueur(min\, p \in P\{p(Ci \rightarrow Cj)\}) & : Si\ i \neq j \\ 0 & : Si\ i = j \end{cases}$$

$$SSM(o) = 1/n \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} ssm(Ci, Cj)$$

Où *n* est le nombre de classes appariées.

La métrique SSM entre deux classes Ci et Cj se calcule en prenant en compte la longueur du plus court chemin entre Ci et Cj. Ci et Cj sont des classes de l'ontologie o.

Ainsi que l'on peut le remarquer dans la figure 23, il existe une corrélation entre le volume du corpus et la SSM. La SSM ne décroit jamais quelque soit le paramètre ou seuil retenu. En général, on peut également observer qu'un paramètre élevé (comme dans KP-8) résulte en une dégradation des performances de la SSM. Toutefois, lorsque le corpus est plus important que celui dont nous disposons, cela peut ne pas se vérifier.

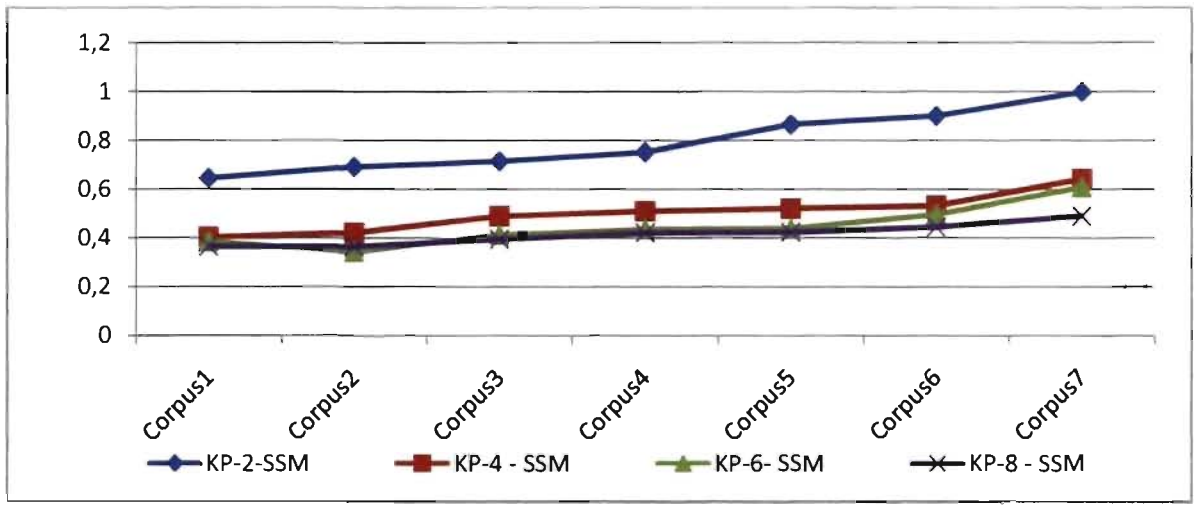


Figure 23. Évolution de la valeur de la métrique SSM sur les différents corpus

Cette métrique est fortement influencée par le fait de considérer les recoupements partiels (Figure 23) ou exacts (Figure 24). En cas de recoupements exacts seulement, des performances similaires sont obtenues pour les paramètres 2, 4 et 6, spécialement dans le corpus le plus riche (corpus 7) où ils obtiennent exactement les mêmes résultats. Cela n'est pas le cas si l'on considère également les recoupements partiels et cela peut se vérifier en comparant les deux figures (Figure 23 et Figure 24).

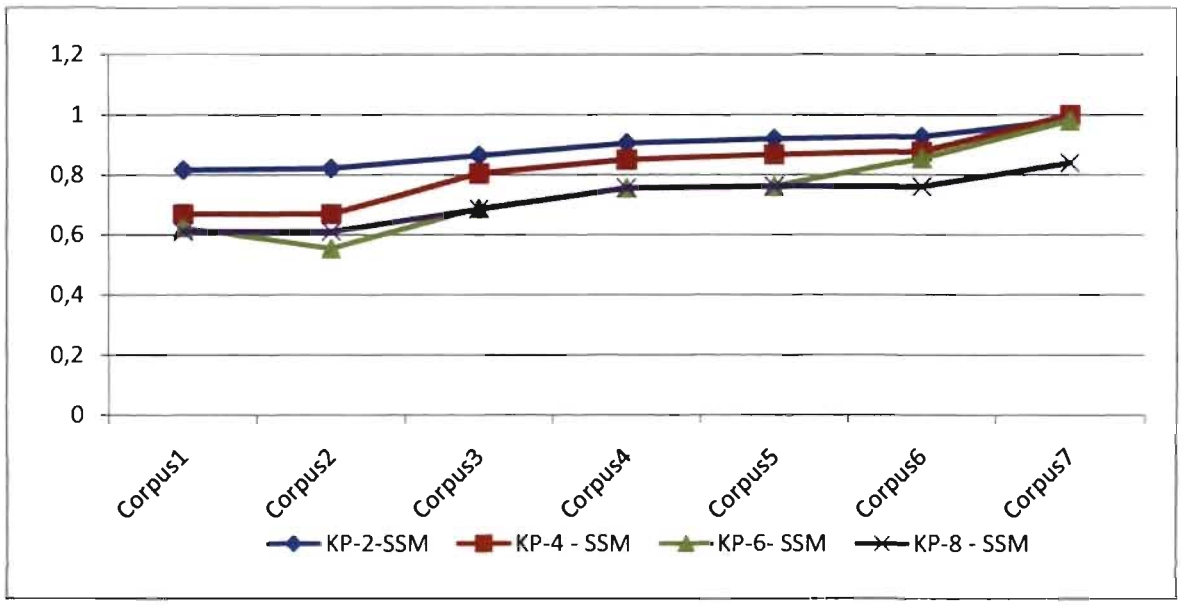


Figure 24. Évolution de la valeur de la métrique SSM sur les différents corpus avec recoupement exact des labels

4.3.4 La métrique « *Betweeness Measure* »

La dernière métrique « *Betweeness Measure* » (BEM) calcule la valeur de centralité « *betweeness* » de chaque mot-clé recherché dans les ontologies. Cette métrique décrit dans quelle mesure un concept se retrouve sur les chemins qui relient d'autres concepts (voir définition ci-dessous (Alani & Brewster, 2006)). L'idée derrière cette métrique est que la centralité d'une classe de l'ontologie par rapport aux autres concepts est importante. Une valeur élevée indique cette centralité.

Soient *ci* et *cj* ∈ {classes[o]}, *ci* et *cj* sont n'importe quelle paire de classes dans l'ontologie *o*, *C[o]* est l'ensemble des classes dans l'ontologie o et *bem(c)* est la mesure « *betweeness* » pour la classe *c*.

$$\boldsymbol{bem(c) = \sum_{ci \neq cj \neq c \in C[o]} \frac{\delta cicj(c)}{\delta cicj}}$$

$\delta cicj$ est le nombre de plus courts chemins entre *ci* et *cj* et $\delta cicj(c)$ est le nombre de plus courts chemins entre *ci* et *cj* qui passent par *c*. Donc bem(c) représente la proportion de plus courts chemins liant deux concepts *ci* et *cj* qui contiennent le concept *c*. Un concept c central dans l'ontologie a un bem(c) élevé. Enfin la valeur de centralité de l'ontologie o est calculée par la formule suivante :

$$\boldsymbol{BEM(o) = \frac{1}{n} \sum_{k=1}^{n} bem(C_k)}$$

Où *n* est le nombre de classes appariées dans l'ontologie *o* et *BEM(o)* la valeur « *betweeness* » moyenne de l'ontologie *o*.

La librairie ONTO-EVALUATOR utilise l'algorithme fourni par JUNG (Java Universal Network/Graph Framework, 2007) afin de calculer cette métrique. Cet algorithme calcule le nombre de plus courts chemins passant par chaque concept de l'ontologie. Une valeur plus élevée est assignée aux concepts qui se retrouvent sur plusieurs de ces plus courts chemins.

Nous avons constaté qu'un degré de connexion assez bas (I=2 ou I=4) doit être défini afin d'obtenir un BEM intéressant. La figure suivante suggère que les ontologies KP-2 et KP-4 donnent les meilleurs résultats concernant cette métrique.

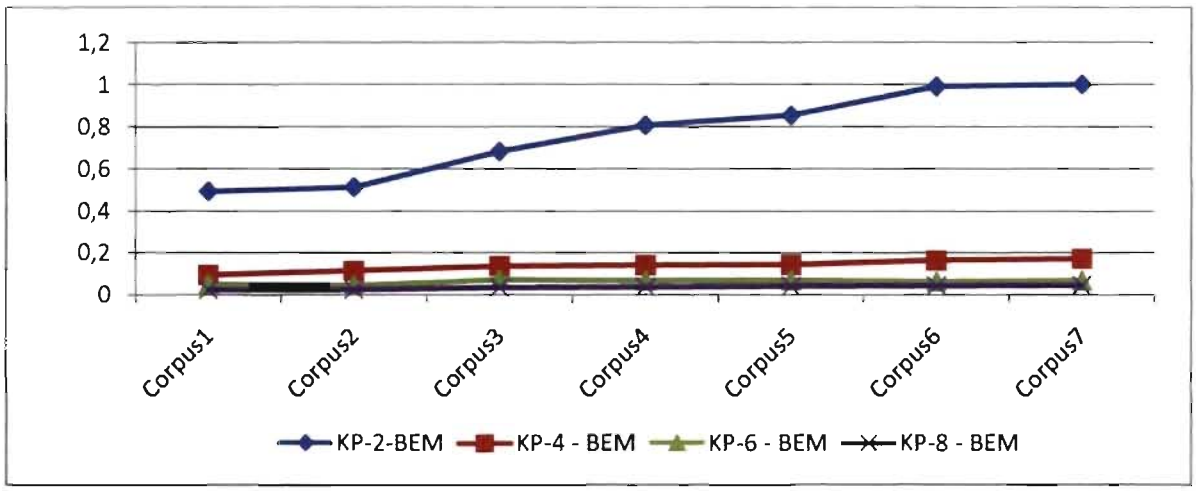


Figure 25. Évolution de la valeur de la métrique BEM sur les différents corpus

4.3.5 Calcul du score d'une ontologie

Une fois que toutes les valeurs des métriques sont obtenues, un score total peut être calculé en pondérant les métriques avec des poids similaires ou différents selon l'importance que revêt telle ou telle métrique pour l'expert (Alani & Brewster, 2006).

Soit $M= \{M[1], ..., M[i]\}=\{CMM, DEM, SSM, BEM\}$, Wi est le facteur de poids et O est l'ensemble des ontologies à classer, $|O|$ représente le nombre d'ontologies dans l'ensemble O.

$$Score(o \in O) = \sum_{i=1}^{4} Wi \left(\frac{M[i]}{max(M[j])} \right)_{1 \leq j \leq |O|}$$

Notons que la valeur de chaque métrique est normalisée de façon à être dans l'intervalle [o-1] en la divisant par la valeur maximale de cette métrique dans toutes les ontologies à comparer. Le premier rang est octroyé à l'ontologie ayant le plus grand score et ainsi de suite.

Le score total est important dans l'analyse comparative car il permet de comparer les ontologies que nous avons générées avec celles d'un autre outil, en l'occurrence TEXT-TO-ONTO (Text-To-Onto, 2007). C'est pourquoi nous indiquons les scores obtenus dans la prochaine section après avoir défini ce que nous entendons par «analyse comparative».

4.4 Analyse comparative

L'analyse comparative sert à comparer les résultats de 2 outils pour d'une part mesurer la différence de résultats et voir si l'outil développé a de meilleures performances et d'autre part afin de pallier l'absence d'une ontologie repère, le cas échéant.

L'analyse comparative nécessite de générer une ontologie avec TEXT-TO-ONTO en utilisant le même corpus documentaire. Contrairement à TEXCOMON, TEXT-TO-ONTO ne dispose pas de paramètres reliés au degré sortant d'un concept qui permettent d'obtenir plusieurs ontologies. Toutefois, d'autres paramètres peuvent être pris en compte notamment le **support** attribué aux règles d'association générées par l'algorithme.

Pour pouvoir expliciter un peu mieux cette notion de support, il importe de définir tout d'abord l'apprentissage par règles d'association. Ce type d'apprentissage permet de détecter les items (dans notre cas les termes) qui co-occurrent fréquemment et d'extraire des règles qui relient ces items. Le support d'une règle d'association est le pourcentage de groupes (en l'occurrence, de documents) qui contiennent tous les items de la règle.

Nous avons généré deux ontologies dans TEXT-TO-ONTO (TTO-1, TTO-2) à partir de chaque corpus (7 corpus au total). Deux **supports** ont été considérés : un support de 0 qui indique que n'importe quelle règle d'association générée est considérée comme valide (**TTO-1**) et un support de 0.1 (**TTO-2**). Autrement dit, chaque corpus nous a permis de générer, avec TEXT-TO-ONTO, deux ontologies différentes.

D'après nos expérimentations, nous avons constaté des disparités de résultats importantes entre TTO-1 et TTO-2 dans tous les corpus. En fait, TTO-1, qui correspond à un support de 0, contient beaucoup de propriétés qui n'ont aucun « sens » (leur degré de

support est très faible) mais qui contribuent à une meilleure valeur de certaines des métriques (notamment au niveau de la richesse de description des classes). Dans TTO-2, un support relativement bas comme 0.1 conduit à la disparition de toutes les règles d'association générées dans l'ontologie TTO-1, ce qui signifie que les règles extraites ont un support inférieur à 0.1. Pour cette raison, même si nous présentons les deux ontologies et leurs résultats, nous pensons que seule l'ontologie TTO-2 peut être effectivement comparée à celles générées par TEXCOMON.

4.4.1 Description de l'expérimentation avec TEXT-TO-ONTO

TEXT-TO-ONTO est exécuté sur chacun des corpus dans l'environnement *KAON Workbench* (KAON, 2007). Dans chaque corpus, les opérations suivantes sont effectuées :

- l'extraction des termes du domaine ;
- l'extraction d'instances ;
- l'extraction de règles d'association avec les deux supports évoqués (0 et 0.1). Les associations extraites sont ensuite considérées comme des relations de l'ontologie ;
- l'extraction de relations au moyen de l'utilisation de patrons linguistiques ;
- l'extraction de relations taxonomiques. Deux possibilités sont offertes à ce niveau par TEXT-TO-ONTO : l'utilisation d'une approche combinée basée sur des patrons et des heuristiques et une approche basée sur l'analyse formelle de concepts. Nous avons utilisé la première approche car nous n'avons pas du tout été convaincus par les résultats obtenus avec la seconde.

Les scores des ontologies générées par TEXT-TO-ONTO sont ensuite obtenus selon la même procédure que pour les ontologies générées par TEXCOMON et en utilisant les mêmes métriques et la formule du calcul du score d'une ontologie, évoquée en section 4.3.5. Rappelons que des poids peuvent être assignés aux différentes métriques calculées

(CMM, DEM, SSM, BEM) afin de pondérer la contribution d'une métrique particulière dans le score total d'une ontologie.

Dans ce qui suit, nous présentons un ensemble d'expérimentations où nous avons fait varier les poids des différentes métriques.

4.4.2 Variations des poids des métriques dans le score total

La première expérimentation a consisté en l'assignation d'un même poids (0.25) à toutes les métriques contribuant au calcul du score. D'après ses résultats, il est clair que les ontologies de TEXCOMON obtiennent de meilleurs scores sur tous les corpus. KP-8 est la seule ontologie qui obtienne des résultats moins bons que TTO-1 et TTO-2, ainsi que décrit par le tableau XII et la figure 26.

Ontologie	Score	Rang
KP2	1	1
KP4	0.409	2
KP6	0.339	4
KP8	0.265	6
TTO-1	0.385	3
TTO-2	0.290	5

Tableau XII. Scores et rangs dans l'expérimentation sur le plus grand corpus (corpus 7) assignant un même poids à toutes les métriques (0.25)

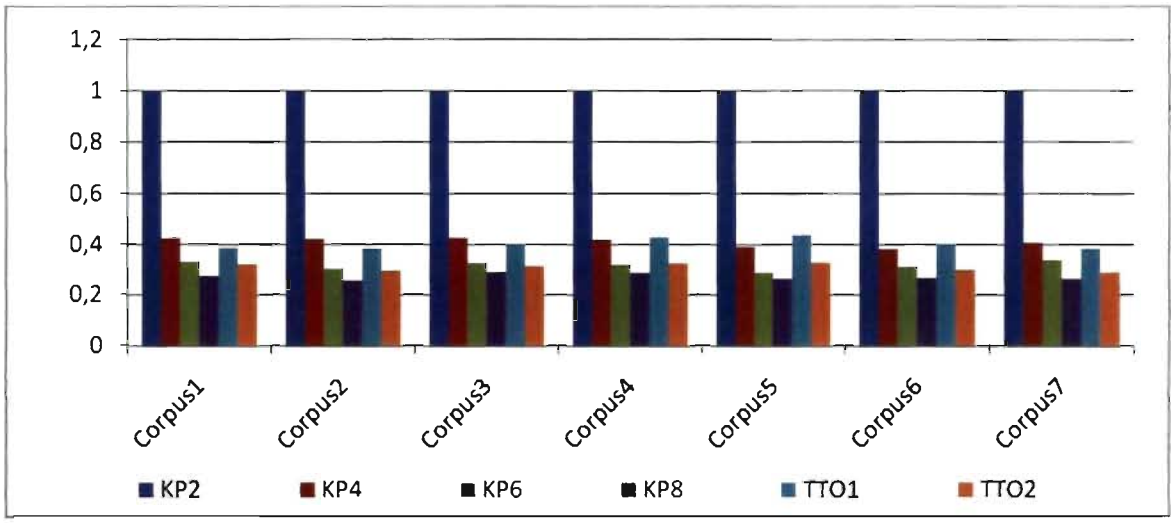


Figure 26. Distribution des scores sur tous les corpus- même poids pour toutes les métriques.

En faisant varier les poids et en assignant des valeurs de 0.2, 0.2, 0.2 et 0.4 respectivement pour les métriques CMM, DEM, BEM et SSM sur le plus grand corpus, on peut voir que toutes les ontologies de TEXCOMON obtiennent de meilleurs résultats que TTO-2 (ce qui est le plus probant) et que les résultats de TEXCOMON sont meilleurs que TTO-1 pour les ontologies KP-2, 4 et 6 (Tableau XIII et Figure 27).

Ontologie	Score	Rang
KP2	1	1
KP4	0.46	2
KP6	0.39	3
KP8	0.31	5
TTO-1	0.34	4
TTO-2	0.24	6

Tableau XIII. Scores et rangs sur le plus grand corpus (corpus 7) avec différents poids (0.2, 0.2, 0.2, 0.4) pour les métriques CMM, DEM, BEM et SSM respectivement

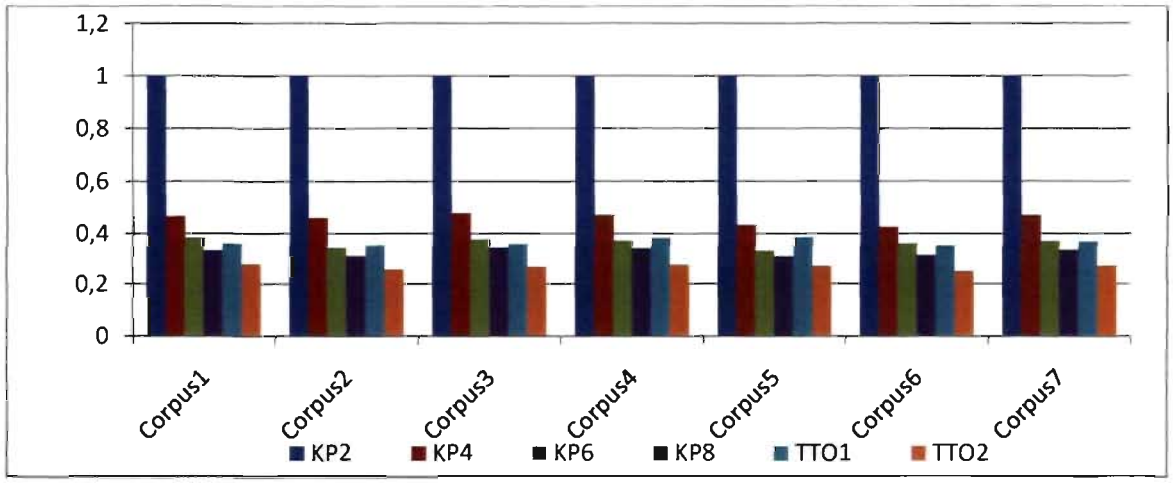


Figure 27. Score sur tous les corpus avec une distribution de poids 0.2, 0.2, 0.2, et 0.4 pour les métriques CMM, DEM, BEM et SSM respectivement

Dans une troisième expérimentation, nous avons ensuite considéré seulement les recoupements exacts dans la métrique CMM et nous avons assigné les poids 0.5, 0, 0, 0.5 aux métriques CMM, DEM, BEM et SSM respectivement sur le corpus 7 (Tableau XIV).

Ontologie	Score	Rang
KP2	0.99	2
KP4	1	1
KP6	0.99	3
KP8	0.83	4
TTO-1	0.61	5
TTO-2	0.46	6

Tableau XIV. Scores sur le plus grand corpus (corpus 7) avec les poids (0.5, 0, 0, 0.5) pour les métriques CMM, DEM, BEM et SSM respectivement

Pour la première fois, on constate que l'ontologie KP-4 a un meilleur score que KP-2. Cela indique que lorsque les seules métriques CMM et SSM sont considérées dans le

calcul du score, alors c'est l'ontologie KP-4 qui doit être considérée comme la meilleure ontologie (Tableau XIV et Figure 28).

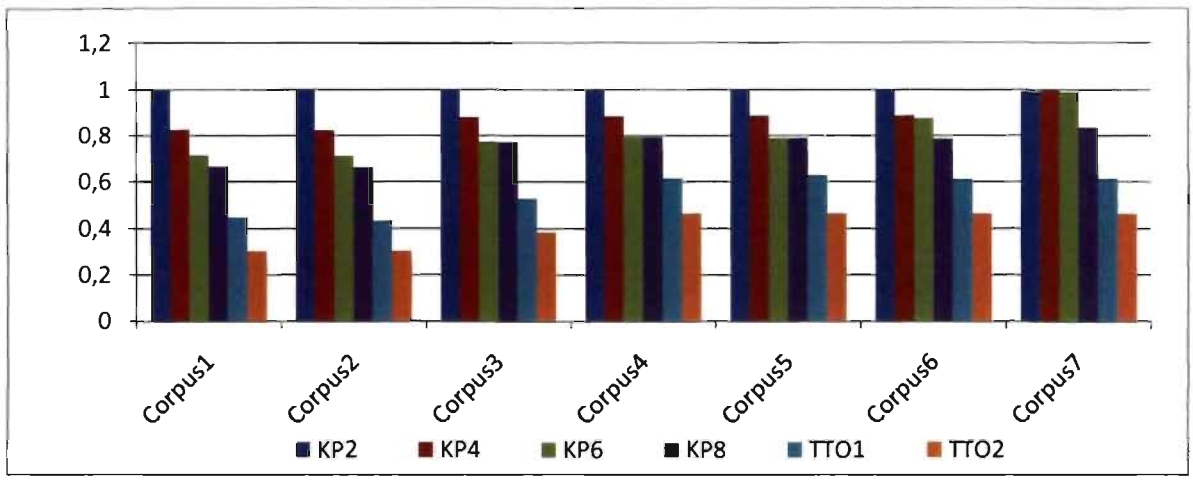


Figure 28. Score total avec distribution de poids de 0.5, 0, 0 et 0.5

Enfin, une dernière expérimentation considère le score comme dépendant uniquement de la métrique CMM. Dans ce cas, on constate que KP-2, 4 et 6 obtiennent le même score et le même rang, ce qui indique que les trois ontologies contiennent tous les mots-clés recherchés en tant que classes (Tableau XV).

Ontologie	Score	Rang
KP2	1	1
KP4	1	1
KP6	1	1
KP8	0.83	3
TTO-1	0.92	2
TTO-2	0.92	2

Tableau XV. Scores et rangs sur le plus grand corpus (corpus 7) avec différents poids (1, 0, 0, 0) pour les métriques CMM, DEM, BEM et SSM respectivement

Après ces macro-évaluations comparatives, nous nous sommes intéressés à quelques aspects plus précis dans les ontologies générées par TEXCOMON et TEXT-TO-ONTO.

4.4.3 Autres éléments de comparaison

Nous avons tout d'abord compilé quelques statistiques par rapport aux concepts, relations et propriétés des ontologies générées par TEXCOMON sur le corpus le plus riche (corpus 7). Ensuite, nous avons effectué la même opération pour les ontologies générées par TEXT-TO-ONTO et nous avons analysé les différences entre les ontologies générées (présence ou non de propriétés spécifiques, de concepts, de relations, plausibilité de ces éléments).

De manière générale, il ressort de cette comparaison que les ontologies générées par TEXCOMON sont plus intéressantes, notamment dans les liens non taxonomiques, et spécialement si on les compare avec l'ontologie TTO-2 (support=0.1). La table suivante (Tableau XVI) compare les résultats de TEXCOMON et TEXT-TO-ONTO en termes de nombre de concepts et de relations (taxonomiques et non taxonomiques).

	KP-2	***KP-4***	***KP-6***	***KP-8***	***TTO-1***	***TTO-2***
Nombre de classes primitives	413	139	82	57	336	336
Nombre de liens taxonomiques	372	125	84	66	223	223
Nombre de liens non taxonomiques	288	153	103	74	5683	33

Tableau XVI. Quelques statistiques sur les concepts et relations extraits

D'après ce tableau, on peut noter qu'une baisse du nombre de concepts et de relations dans TEXCOMON est consistante avec l'augmentation du paramètre ou seuil, ce qui est logique. Les résultats dans TEXCOMON peuvent être paramétrés, ce qui n'est pas le cas pour TEXT-TO-ONTO où le nombre de classes dans TTO-1 et TTO-2 reste stable. Les 33 relations qui figurent dans la colonne « Nombre de liens non taxonomiques » de TTO-2 sont les seules relations labélisées que TEXT-TO-ONTO a pu extraire du corpus. Ce nombre est très faible comparé aux relations conceptuelles labélisées extraites par TEXCOMON.

Un autre aspect intéressant peut être remarqué dans la différence entre le nombre de liens non taxonomiques dans TTO-2 (33) et TTO-1 (5683). Cette baisse spectaculaire est reliée au support de 0.1. Cela indique que TTO-1 possède des relations qui correspondent à des règles d'association avec un support inférieur à 0.1. Ces relations contribuent à la « bonne performance » de TTO-1 dans l'analyse structurelle, spécialement si l'on considère la mesure SSM. Toutefois, sémantiquement, ces relations n'ont aucun intérêt et ne devraient normalement pas se retrouver dans l'ontologie.

Une autre possibilité pour comparer les deux systèmes est de sélectionner un mot-clé et de l'analyser dans les ontologies issues de TEXCOMON et de TEXT-TO-ONTO. Là encore, on peut constater une différence significative de résultats entre TTO-1 et TTO-2. Notons encore que nous comparons les ontologies issues du corpus le plus riche (corpus 7).

Les tableaux XVII et XVIII illustrent ces comparaisons statistiques pour les mots-clés «*SCO*» et «*asset*». Notons que le nombre de superclasses ne comprend pas la classe générique « *OWL-Thing* ».

T=SCO	*KP-2*	*KP-4*	*KP-6*	*KP-8*	*TTO-1*	*TTO-2*
Super- classes	3	2	4	4	1	1
Sous-classes	2	2	2	2	2	2
Voisins	6	0	0	0	2	2
Propriétés	18	12	10	7	118	0

Tableau XVII. Statistiques sur le concept « *SCO* »

T=asset	*KP-2*	*KP-4*	*KP-6*	*KP-8*	*TTO-1*	*TTO-2*
Super- classes	3	2	2	2	1	1
Sous-classes	0	0	0	0	0	0
Voisins	7	1	0	0	4	4

Propriétés	11	8	7	4	172	0

Tableau XVIII. Statistiques sur le concept « *asset* »

Notons que TEXCOMON découvre plus de superclasses, de sous-classes et de voisins pour ces deux concepts. Notons également la disproportion entre le nombre de propriétés dans TTO-1 et TTO-2 et entre TTO-1, TTO2 et les ontologies de TEXCOMON. Là encore, les 172 relations proviennent des règles d'association de TTO-1 avec un support très faible, raison pour laquelle le nombre tombe à 0 dans TTO-2.

Cette comparaison statistique effectuée, nous avons voulu étudier de plus près un sous-ensemble des relations générées pour les mots-clés « asset » et « SCO » dans TEXCOMON (Tableau XIX).

asset - can_be_described_with - asset_metadata asset - are_electronic_representations_of - media asset - will_not_use - SCORM_API asset - to_provide - descriptive_information asset - will_not_be_included_within - package asset - is_basic_building_block_of - training_resources
SCO - is_tracked_by - LMS SCO - must_able_in - order SCO - can_be_copied_from - place SCO - to_report - progress SCO - has - ways SCO - initializes - communication SCO - must_establish - communication_session SCO - communicate_to - LMS SCO - finds - API_Instance SCO - to_find_out - descriptive_information SCO - to_provide - descriptive_information SCO - is_collection_of - is_collection_of_Assets SCO - are_content_objects_in - SCORM SCO - terminates - communication SCO - invokes - functionality SCO - can_be_launched_in - Web_browsers SCO -can_be_described_with - metadata

SCO - is_required_to - issue

Tableau XIX. Un extrait des relations générées pour les termes « *asset* » et « *SCO* » dans TEXCOMON

Nous avons effectué la même opération pour TEXT-TO-ONTO (Tableau XX).

asset - defaultProperty3,206 - file
asset - defaultProperty3,698 - perform
asset - defaultProperty1,521 - inform
asset - defaultProperty656 – package
asset - defaultProperty3,923 - component
asset - defaultProperty4,693 - creat
asset - defaultProperty1,167 – collect
asset - defaultProperty5,430 - e
asset - defaultProperty1,292 – commun
asset - defaultProperty2,461 - context
asset - defaultProperty3,118 - compon
asset - defaultProperty713 - resource

Tableau XX. Un extrait des relations générées pour le concept « *Asset* » dans TTO- 1

Aucune relation labélisée n'a pu être extraite par TEXT-TO-ONTO pour les concepts « asset » et « SCO » (raison pour laquelle nous ne présentons pas d'exemple avec le concept « SCO »). De manière générale, les problèmes constatés avec TEXT-TO-ONTO sont les suivants :

- Etant donné que la majorité des relations proviennent de la détection de règles d'association, TEXT-TO-ONTO n'extrait souvent pas de label pour les relations entre concepts contrairement à TEXCOMON. Ce manque de label se traduit par un libellé comme « *defaultProperty* » et il est ensuite assez difficile d'affecter un sens à ce type de relations ;
- TEXT-TO-ONTO ne conserve pas le label complet d'un concept mais seulement sa racine (stem) au niveau de l'ontologie générée, contrairement à TEXCOMON qui

conserve les deux. Cela rend très difficile la compréhension de l'ontologie dans un éditeur tel que Protégé.

Il convient toutefois de noter que l'exécution de TEXT-TO-ONTO prend beaucoup moins de temps que pour TEXCOMON en raison, notamment, de l'absence d'analyse linguistique profonde.

4.5 Analyse sémantique

En plus de l'analyse comparative et structurelle, il est important d'effectuer une évaluation sémantique auprès des experts du domaine. Cette évaluation s'appuie sur les experts pour détecter dans quelle mesure l'ontologie du domaine générée reflète la connaissance du domaine.

Pour effectuer l'analyse sémantique des ontologies de TEXCOMON, les experts ont eu le choix entre visualiser les ontologies en utilisant l'environnement Protégé (Protégé, 2007) et ses plugins graphiques et visualiser les concepts et les relations sous forme de triplets dans un fichier txt généré par TEXCOMON. Les experts pouvaient également visualiser les cartes de concepts via *un outil de recherche de concepts* disponible dans TEXCOMON.

En ce qui concerne TEXT-TO-ONTO, les experts pouvaient visualiser l'ontologie dans l'environnement Protégé et également directement dans l'environnement de TEXT-TO-ONTO (KAON, 2007). Cette dernière option était plus simple en raison de problèmes lors de l'exportation de l'ontologie en OWL et notamment le manque de labels explicites pour les classes.

L'évaluation sémantique de la qualité de l'ontologie repose sur l'examen d'un ensemble de facettes :

- Une facette « syntaxique » pour estimer si les concepts sont corrects en termes de label et si les relations sont correctes en termes de labels et d'arguments ;

- Une facette sémantique pour estimer la pertinence des concepts et des relations par rapport au domaine.

Par exemple, dans la phrase : « *Assets can be described with Asset Metadata* », et en supposant que *Asset* et *Asset Metadata* aient été retenus comme concepts, la relation "*can be described with*" est une propriété OWL (*OWL Object Property*) sémantiquement pertinente dont le domaine est la classe *Asset* et dont la portée (range) est la classe *Asset Metadata*. On peut également remarquer que chaque label est syntaxiquement correct, soit : *asset, asset metadata, can be described with.*

Nous commencerons par décrire l'analyse sémantique des ontologies générées par TEXCOMON avant d'aborder celles de TEXT-TO-ONTO. Notons que nous nous concentrons, au niveau de cette analyse, sur les ontologies issues du corpus le plus riche.

4.5.1 Expérimentation avec TEXCOMON

A partir du corpus 7, TEXCOMON a extrait des cartes de concepts composées de 1139 termes du domaine et de 1973 relations entre les termes. Comme nous l'avons expliqué, seulement un sous-ensemble de ces termes et relations a été retenu dans l'ontologie. Le nombre de concepts et relations qui se sont finalement retrouvés dans les ontologies du domaine est indiqué dans le tableau suivant (Tableau XXI).

Par classes primitives, nous entendons des concepts correspondant à un terme du domaine précis tandis que les classes définies réfèrent à la définition de certains liens d'équivalence entre classes, notamment via les abréviations et acronymes.

Ontologie	Nombre de classes primitives	Nombre de classes définies	Nombre de relations hiérarchiques	Nombre de relations conceptuelles
KP2	472	9	430	359
KP4	142	10	145	174
KP6	90	10	107	113

KP8	62	10	87	80

Tableau XXI. Nombre de concepts et relations dans les ontologies du domaine générées par TEXCOMON

Nous avons demandé à deux experts d'évaluer les ontologies générées par TEXCOMON en tenant compte des critères d'évaluation cités ci-dessus : les deux experts avaient pour consigne d'éliminer les concepts et relations trop vagues tels que « section », « example », etc. et de considérer seulement les concepts et relations pertinents au domaine.

Les deux tableaux suivants montrent le taux de pertinence des concepts et relations selon les deux experts du domaine.

Ontologie	**Classes primitives pertinentes (%)**	**Classes définies pertinentes (%)**	**Relations hiérarchiques pertinentes (%)**	**Relations conceptuelles pertinentes (%)**
KP2	87.5	55.55	86.51	80.22
KP4	90.84	100	86.21	87.93
KP6	91.11	100	78.5	91.15
KP8	91.93	100	75.86	92.5

Tableau XXII. Évaluation des ontologies générées par TEXCOMON (Expert 1)

Ontologie	**Classes primitives pertinentes (%)**	**Classes définies pertinentes (%)**	**Relations hiérarchiques pertinentes (%)**	**Relations conceptuelles pertinentes (%)**
KP2	85.8	55.55	82.09	79.94
KP4	90.84	100	83.45	91.38
KP6	88.89	100	75.70	91.15

KP8	88.71	100	74.71	93.75

Tableau XXIII. Évaluation des ontologies générées par TEXCOMON (Expert 2)

Le tableau suivant retrace les taux de pertinence moyens dérivant de la double évaluation citée ci-dessus.

Ontologie	**Classes primitives pertinentes (%)**	**Classes définies pertinentes (%)**	**Relations hiérarchiques pertinentes (%)**	**Relations conceptuelles pertinentes (%)**
KP2	86.65	55.55	84.3	80.08
KP4	90.84	100	84.83	89.65
KP6	90	100	77.1	91.15
KP8	90.32	100	75.28	93.12

Tableau XXIV. Évaluation humaine moyenne de la pertinence des concepts et relations générés par TEXCOMON

En moyenne, dans le pire des cas, les experts ont jugé que 86.65% des classes primitives et 80.08% des relations conceptuelles étaient correctes et plausibles. Cela veut dire que les classes et relations effectivement extraites avaient souvent un label correct et un sens dans le domaine. Toutefois, cela ne veut pas dire que toutes les classes et relations nécessaires au domaine ont été extraites (lorsqu'elles se trouvaient exprimées dans le corpus). En effet, si la connaissance se trouve exprimée dans le corpus selon des patrons lexico-syntaxiques non reconnus par TEXCOMON, alors cette connaissance ne peut donc pas être extraite par le système.

Des outils sont mis à la disposition de l'ingénieur ontologique pour évaluer et améliorer l'ontologie du domaine suite aux résultats des expérimentations, soit en intervenant directement sur l'ontologie, soit en intervenant sur les cartes de concepts.

4.5.2 Expérimentation avec TEXT-TO-ONTO

Nous avons effectué le même type d'évaluation sémantique sur les deux ontologies générées par TEXT-TO-ONTO (Tableaux XXV, XXVI et XXVII). Notons que cela a été un exercice assez difficile en raison de certains bogues dans TEXT-TO-ONTO et du fait d'un mécanisme d'exportation en *owl* déficient. Par exemple, dans TEXT-TO-ONTO, l'option « *Relation Learning* » du menu permet de découvrir quelques relations conceptuelles labélisées, mais lorsque l'ontologie est générée en format *owl*, ces relations ne se retrouvent pas dans l'ontologie finale.

Un autre inconvénient de TEXT-TO-ONTO résulte du fait que les classes sont libellées avec un identificateur généré et qu'il est ensuite assez difficile de repérer une classe ayant un label précis comme « SCO » dans l'ontologie OWL (les racines des labels sont stockées dans la propriété d'annotation « *rdfs : label* »). Les experts ont donc étudié les résultats de TEXT-TO-ONTO dans Protégé mais également dans l'environnement KAON où les classes disposent de leurs libellés et où toutes les relations sont conservées. La figure 29 montre d'ailleurs une partie des 33 relations labellisées générées par TEXT-TO-ONTO.

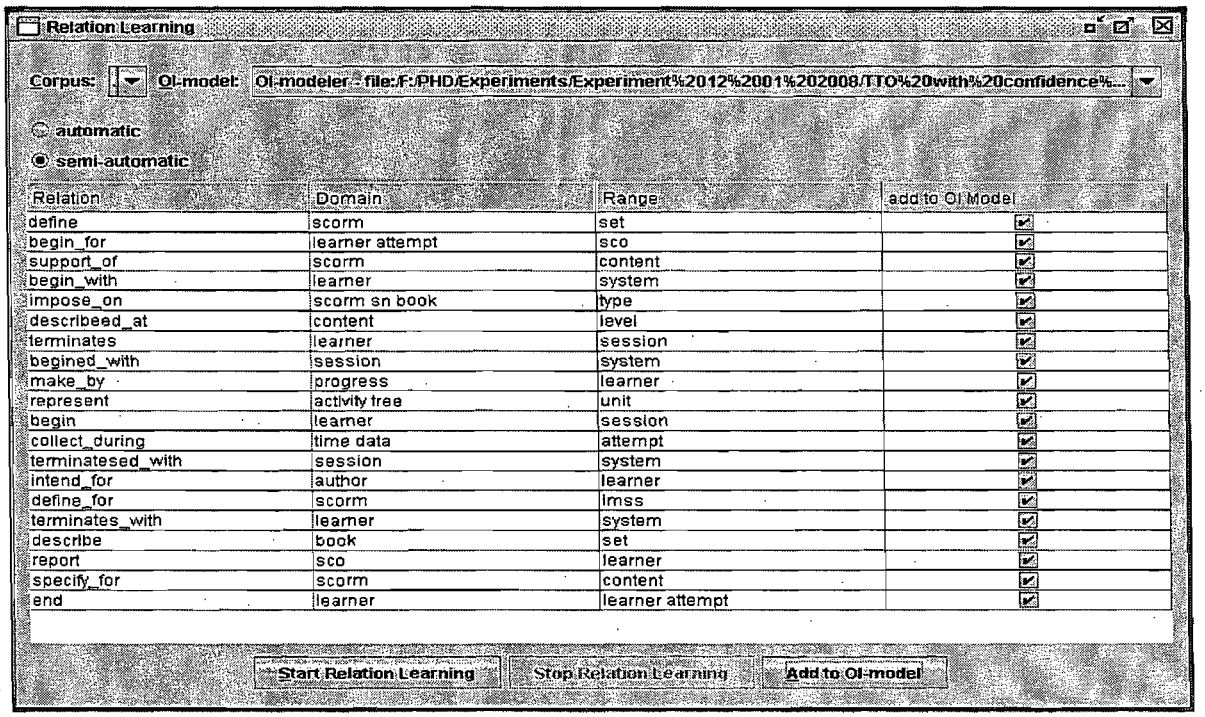


Figure 29. Une partie des 33 relations labellisées générées par TEXT-TO-ONTO

Le tableau XXVI retrace le nombre de classes et de relations générées par TEXT-TO-ONTO. Seul le nombre de relations conceptuelles diffère entre TTO1 et TTO2.

Ontologie	**Nombre de classes primitives**	**Nombre de classes définies**	**Nombre de relations hiérarchiques**	**Nombre de relations conceptuelles**
TTO1	336	0	223	5683
TTO2	336	0	223	33

Tableau XXV. Nombre de concepts et relations dans les ontologies du domaine générées par TEXT-TO-ONTO

Là encore, nous avons demandé aux deux experts d'évaluer la pertinence des classes et relations. Les tableaux XXVII et XXVIII synthétisent ces évaluations.

Ontologie	Classes primitives pertinentes (%)	Relations hiérarchiques pertinentes (%)	Relations conceptuelles pertinentes (%)
TTO1	72.02	58.74	0.3
TTO2	72.02	58.74	51.51%

Tableau XXVI. Évaluation des ontologies générées par TEXT-TO-ONTO (Expert 1)

Ontologie	Classes primitives pertinentes (%)	Relations hiérarchiques pertinentes (%)	Relations conceptuelles pertinentes (%)
TTO1	74.10	36.32	0.32
TTO2	74.10	36.32	54.55

Tableau XXVII. Évaluation des ontologies générées par TEXT-TO-ONTO (Expert 2)

Le tableau suivant retrace les taux de pertinence moyens dérivant de la double évaluation citée ci-dessus.

Ontologie	Classes primitives pertinentes (%)	Relations hiérarchiques pertinentes (%)	Relations conceptuelles pertinentes (%)
TTO1	73.06	47.53	0.31
TTO2	73.06	47.53	53.03

Tableau XXVIII. Évaluation humaine moyenne de la pertinence des concepts et relations générés par TEXT-TO-ONTO

En moyenne, on peut constater que les ontologies générées par TEXT-TO-ONTO ont un taux de pertinence moyen de 73.06% pour les classes primitives et de 47.53% pour les relations hiérarchiques. On peut également constater qu'aucune classe définie n'est

générée. Au niveau des concepts, TEXT-TO-ONTO génère deux fois le même concept, une fois sous sa forme racine (stem) et l'autre sous sa forme complète. Par exemple, on peut retrouver « *aggreg* » et « aggregation ». Il en résulte une redondance de concepts et de leur hiérarchie. Dans l'ontologie finale présentée aux experts, nous avons donc choisi de conserver uniquement les concepts sous leur forme complète.

Un point important à souligner est la faiblesse de TEXT-TO-ONTO concernant l'extraction de relations conceptuelles labélisées (en utilisant des patrons linguistiques). TEXT-TO-ONTO n'en génère que 33 en tout alors que TEXCOMON en extrait 80 dans son ontologie la plus compacte (KP8) et 359 dans KP2. Parmi ces 33 relations, seules 17 (expert 1) ou 18 (expert 2) relations conceptuelles ont été considérées comme valides et pertinentes, aboutissant un taux moyen de 53.03%. La deuxième remarque a trait, à l'inverse, au nombre impressionnant de relations résultant de l'apprentissage par règles d'associations. Malheureusement, toutes ces règles ont un support inférieur à 0.1, et il n'est pas possible de les parcourir manuellement une par une en raison de leur nombre très important. TTO1 a obtenu un taux moyen de 0.31% (18 relations pertinentes parmi les 5683). Nous avons considéré que des règles d'association avec un support inférieur à 0.1 n'étaient pas satisfaisantes et avons donc considéré les 5650 associations générées (sans label) comme non pertinentes. Pour appuyer cette démarche, nous avons extrait au hasard un échantillon de relations qui se sont effectivement avérées vides de sens (Il a été très compliqué d'en trouver une qui reflète une relation « ontologique »).

4.6 Analyse des résultats

Quels types de conclusions peuvent être tirés de ces expérimentations ?

Tout d'abord, TEXCOMON permet de calibrer l'ontologie générée et sa taille dans l'analyse structurelle. Un concepteur d'ontologie peut être intéressé par une ontologie plus ou moins compacte (c'est-à-dire contenant plus ou moins de concepts et relations). D'ailleurs, nous avons remarqué que plus le paramètre est élevé dans les ontologies de

TEXCOMON (2, 4, 6 et 8), moins le bruit généré l'est, particulièrement dans les relations conceptuelles. Toutefois, si le seuil est trop élevé, certaines connaissances pertinentes peuvent être occultées. Il faut donc arriver à un calibrage du paramètre qui tienne compte du nombre de phrases à traiter (taille du corpus) et du nombre de patrons extraits. Par ailleurs, nous avons remarqué que de nombreuses relations hiérarchiques pertinentes étaient perdues lorsque le seuil était augmenté. Il est possible que ces relations hiérarchiques n'aient pas à être influencées par le degré de connexion d'un concept et puissent demeurer dans l'ontologie quelque soit ce degré.

Une caractéristique intéressante à retenir également dans le cas de l'analyse structurelle est la possibilité de faire varier le poids des différentes métriques et de faire varier le recoupement (total ou partiel) pour la métrique CMM. Ces variations ont des impacts importants sur le score des ontologies et il y a d'importantes questions qui doivent être soulevées pour appliquer ces variations :

- tout d'abord, quel est l'objectif du concepteur ontologique ?
- deuxièmement, quelles sont les métriques les plus importantes compte tenu du domaine, de l'objectif et des besoins ontologiques ?
- troisièmement, étant donné une ontologie à connexité faible comme KP-2 mais ayant un bon score, est-il possible d'obtenir une ontologie plus compacte qui préserve le même score que KP-2 ?

Si la réponse à la dernière question est affirmative, alors une ontologie plus compacte doit être préférée à une moins compacte, car elle comprend des concepts plus richement interconnectés et conserve les mots-clés recherchés en tant que classes. Par exemple, dans le tableau XIV, KP-4 doit être choisie comme étant la meilleure ontologie alors que dans le tableau XV, c'est KP-6 qui est la meilleure ontologie. Elle conserve le même score que KP-2 et KP-4 mais elle est bien plus compacte.

Enfin, étant donné un ensemble de mots-clés considérés comme des concepts importants du domaine, on peut formuler les remarques suivantes :

- la calibration des paramètres peut être effectuée par rapport à la métrique CMM si la caractéristique la plus importante est le recoupement total ou partiel des mots-clés recherchés en tant que concepts ontologiques ;
- si la caractéristique la plus importante est que les concepts aient un nombre important d'attributs et de relations, alors c'est la mesure de densité (DEM) qui doit avoir un poids supérieur dans le calcul du score total ;
- si la caractéristique la plus importante est non seulement la richesse de connexion des concepts mais également leur centralité, alors ce sont les mesures BEM et SSM qui doivent primer.

Notre opinion est que toutes ces mesures sont importantes. En général, dans ces expérimentations, on peut conclure que les ontologies KP-2, KP-4 et KP-6 sont satisfaisantes. Si on se fie aux résultats de l'analyse sémantique, KP-4 semble effectivement l'ontologie la plus intéressante. Enfin, il est important de noter qu'il n'existe pas (ou pas encore) de science exacte pour évaluer une ontologie. Certaines leçons peuvent toutefois être retenues de ce qui précède :

- lorsqu'il n'existe pas d'ontologie repère « *gold standard* » pour un domaine particulier, il n'est pas toujours possible d'en créer une. Une autre façon d'évaluer les ontologies doit donc être adoptée ;
- la comparaison de l'ontologie obtenue avec d'autres ontologies générées par d'autres outils sur un même corpus peut être intéressante et peut souligner les forces et faiblesses de votre outil. C'est ce que tend à prouver l'analyse comparative ;
- enfin, l'évaluation d'une ontologie d'un point de vue structurel, c'est-à-dire en considérant les propriétés de l'ontologie en tant que graphe, peut être importante.

Comparer ces résultats avec l'analyse structurelle d'autres ontologies peut aussi s'avérer un bon indicateur.

De manière générale, une analyse sémantique s'avère nécessaire pour valider l'ontologie. Nous avons pu constater que TEXCOMON produit des ontologies plus satisfaisantes que TEXT-TO-ONTO et que les évaluations des experts sont globalement bonnes. Notons toutefois que TEXCOMON prend beaucoup plus de temps que TEXT-TO-ONTO pour traiter le corpus.

4.7 En résumé

La première partie de cette thèse a présenté un état de l'art sur l'acquisition d'ontologies à partir de textes et a expliqué notre approche, concrétisée par l'outil TEXCOMON. Nous avons également effectué une évaluation des ontologies générées et avons obtenu des résultats satisfaisants. Comme nous l'avons souligné dans le début de cette thèse, l'acquisition des connaissances du domaine s'insère dans une problématique plus large d'apprentissage par ordinateur. La génération semi-automatique d'une ontologie du domaine vise à répondre à un besoin : la difficulté de la construction d'un tel modèle pour les systèmes tutoriels intelligents et l'absence d'un modèle du domaine dans le cadre du e-Learning. En sus de ce modèle du domaine, d'autres types de connaissances sont nécessaires pour un EIAH. Dans ce cadre, le chapitre suivant présente le paysage des EIAH actuels et effectue une critique de l'état de l'art. Il aborde les différentes problématiques auxquelles doivent faire face les communautés e-Learning et STI et conclut à la nécessité d'une proposition d'architecture commune pour les EIAH.

5 Paysage des EIAH : Où en sommes-nous ?

Le paysage des EIAH semble revêtir une forme différente selon que l'on regarde ce qui est fait dans la pratique et ce qui est accompli dans les travaux de recherche. Au niveau de la pratique, on peut dire que ce paysage est actuellement dominé par la formation en ligne (e-Learning), basée (c'est souvent le cas) ou non sur des normes et standards. Les recherches universitaires continuent à explorer des techniques plus sophistiquées que le e-Learning standard, notamment via des recherches sur les systèmes tutoriels intelligents, les systèmes hypermédias adaptatifs et des recherches sur les manières d'améliorer la formation en ligne dite traditionnelle.

Etant donné que cette thèse a pour but de proposer une base de connaissances qui soit utilisable aussi bien par des STI que par des environnements de formation en ligne, cette section présente brièvement les spécificités de chaque système de formation : les systèmes tutoriels intelligents (STI), le e-Learning et enfin les standards qui ont réussi à s'imposer dans la formation en ligne.

5.1 Les systèmes tutoriels intelligents

De manière générale, les STI sont des systèmes à base de connaissances dédiés à l'enseignement et à la formation et qui intègrent des techniques d'intelligence artificielle. Ils se composent de quatre modules principaux (Burns & Capps, 1988) (Siemer & Angelides, 1998) :

- *le modèle du domaine* fournit la connaissance du domaine ainsi que l'expertise dans la résolution de problèmes ;
- *le modèle de l'apprenant* retrace la connaissance effective de l'apprenant par rapport à la connaissance du domaine. Il peut également représenter les conceptions erronées de l'apprenant et permet d'effectuer des opérations de diagnostic ;

- *le modèle du tuteur* contient les stratégies pédagogiques d'enseignement et les compétences à acquérir. Ces stratégies doivent être adaptées au modèle de l'apprenant ;
- *l'interface* dans laquelle l'apprenant effectue son apprentissage. C'est la partie visible du STI, au travers de laquelle l'apprenant « dialogue » avec ce dernier.

La figure 30 illustre cette architecture.

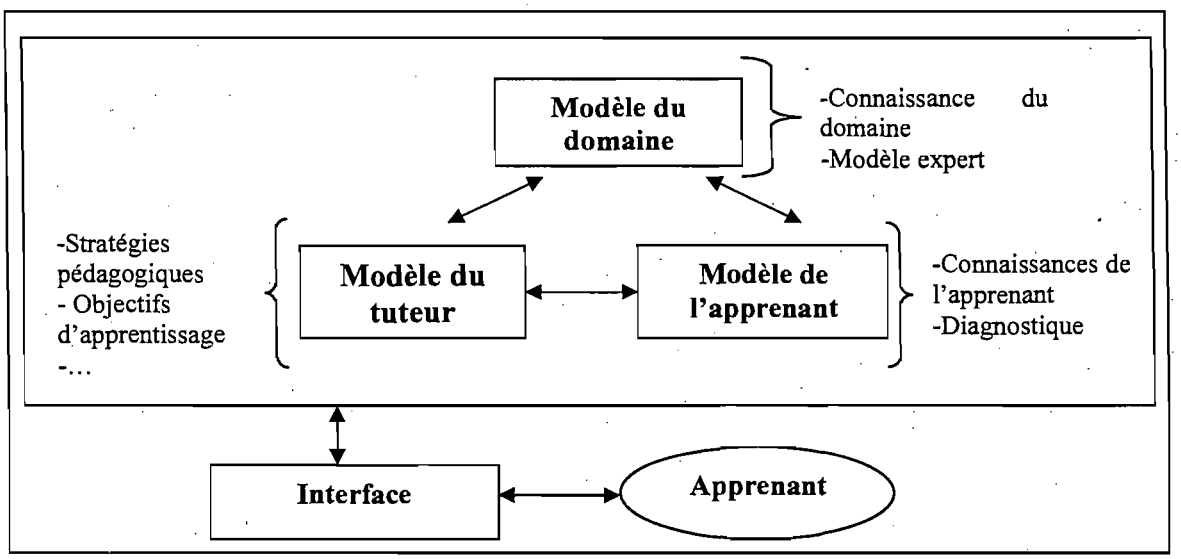


Figure 30. Architecture traditionnelle d'un STI

Traditionnellement, depuis leur création, les STI ont été déployés en utilisant des ressources statiques ou ont modélisé leurs connaissances selon des méthodes et langages propriétaires. Cet état de fait a largement contribué à leur absence dans la pratique et a réduit considérablement leur portée. Cela est d'autant plus vrai dans le cadre de l'acquisition d'un modèle du domaine, entièrement pris en charge (lors de sa création et de sa mise à jour) par un expert humain. L'avènement du Web sémantique et de langages standards permet maintenant d'entrevoir la possibilité de contourner cette limite (technologies propriétaires non réutilisables) et d'adapter les STI à des problèmes à plus grande échelle. Nous reviendrons sur ce point un peu plus tard.

Par ailleurs, l'essor du Web et de son utilisation ont favorisé l'apparition d'une alternative aux STI (ou du moins à ce qui a été considéré comme tel à l'époque): le e-Learning tel qu'on le connaît actuellement était né.

5.2 La formation en ligne (*e-Learning*)

Le e-Learning est la mise en œuvre d'un enseignement en se basant, d'une part, sur l'accès à des ressources et à des services sur le Web, et d'autre part, sur les échanges et la collaboration à distance.

Le e-Learning a donné lieu à une explosion de ressources d'apprentissage (plutôt que de services) sous forme de sites Web ou de pages HTML. On a ainsi utilisé le navigateur Web comme interface pour le déploiement de ces ressources, nommées objets d'apprentissage. Ces derniers sont sensés contenir l'ensemble de l'expertise nécessaire en terme de connaissance et de méthode pédagogique. Ce paradigme a beaucoup séduit, notamment les entreprises, pour sa rapidité (relative) de mise en œuvre et de déploiement. Mais une question demeure : qu'est-ce qu'un objet d'apprentissage ?

Wiley (Wiley, 2000) indique qu'un objet d'apprentissage consiste en n'importe quelle ressource digitale qui peut être réutilisée dans un processus d'apprentissage. Koper (Koper, 2003) insiste également sur les propriétés de réutilisation et d'autonomie des objets d'apprentissage. De telles définitions peuvent englober de multiples objets tels qu'un livre entier ou une simple image. Englober une telle diversité de ressources peut constituer une force mais également une faiblesse : comment mettre au point un modèle qui permette de décrire aussi bien une image qu'un livre ? Et comment faire en sorte que ce livre soit réutilisable ? La réponse à de telles questions ne fait pas encore l'unanimité dans la communauté de recherche.

Statique au départ, le e-Learning s'est ensuite tourné vers des systèmes d'apprentissage hypermédia adaptatifs. Les systèmes hypermédia conventionnels ont d'abord pris la forme de pages Web reliées par des hyperliens que l'apprenant pouvait

parcourir à sa guise. Cette approche a toutefois causé des problèmes de compréhension et d'orientation de l'apprenant. Différentes techniques ont ensuite été utilisées pour pallier ces problèmes et se sont appliquées à produire des systèmes d'apprentissage adaptatifs. Ces techniques se sont inspirées des STI (la planification du curriculum, l'analyse des solutions de l'apprenant, l'aide à la résolution de problèmes de manière interactive ou basée sur des exemples, l'aide à la collaboration) et ont également proposé une adaptation de la navigation et de la présentation des ressources en fonction d'un modèle de l'apprenant. Là encore, comme pour les STI, les systèmes adaptatifs tels que ELM-ART (Brusilovsky, Schwarz, & Weber, 1996) ont aussi fait appel à une modélisation du modèle du domaine et se sont retrouvés face aux mêmes problèmes : coût, complexité, et difficulté de mise à jour.

La multitude des formats de ressources et la nécessité de standardiser les échanges et les contenus pour une plus vaste exploitation et interopérabilité ont mené à la création de standards e-Learning tels que SCORM (SCORM, 2007), les standards IMS (IMS, 1997), LOM (LOM, 2002), etc.

Deux de ces standards occupent une place prépondérante actuellement : le standard SCORM (SCORM, 2007) et les standards IMS notamment IMS-LD (IMS-LD, 2007). Ces standards permettent non seulement d'annoter sémantiquement les objets d'apprentissage via des métadonnées, mais offrent en plus un environnement d'exécution de ces objets d'apprentissage et un modèle de données. Les sections suivantes présentent ces deux standards.

5.2.1 SCORM (Sharable Content Object Reference Model)

Le département de la défense américain a mis en place en 1997 un projet appelé ADL (*Advanced Distributed Learning*) (ADL, 2007) visant à l'établissement de standards pour la formation en ligne. L'objectif de cette initiative était de rendre les contenus d'apprentissage réutilisables, accessibles, durables et interopérables. Pour ce faire, SCORM (SCORM, 2007) a intégré diverses démarches de standardisation présentes sur le marché et

développées par des groupes comme *IMS Global Learning Consortium Inc., AICC, ARIADNE* et *IEEE LTSC.*

Brièvement, SCORM est un modèle de référence qui se scinde en un modèle d'agrégation de contenu et un environnement d'exécution. Il définit un ensemble de normes (techniques, conceptuelles et méthodologiques) pour la production du contenu mais aussi pour l'environnement d'enseignement de ce contenu. A terme, cette initiative vise à la création de librairies digitales composées d'objets d'apprentissage (*Learning objects*). L'objectif final serait de pouvoir générer des cours dynamiquement à partir de ces objets de connaissances selon les besoins de l'apprenant, quelque soit l'endroit où ces objets sont physiquement entreposés.

Le modèle d'agrégation du contenu SCORM a pour but de définir des ressources d'apprentissage réutilisables et partageables, ainsi que de les agréger en un contenu structuré. Plusieurs types de ressources ont été définis dans ce modèle : les actifs, les objets de contenu partageables, les activités, les organisations de contenu et les agrégations de contenu.

Les actifs (*assets*) : il s'agit de la forme la plus élémentaire de contenu d'apprentissage comme des textes, des images, des sons, des pages Web, des fonctions JavaScript, etc.

Les objets de contenu partageables (*SCOs* ou *Sharable Content Objects*) : un SCO est composé d'un ensemble d'actifs, ainsi que d'un dispositif de communication normalisé avec l'environnement d'apprentissage. Les SCOs sont des unités qui doivent rester relativement petites, mais il n'existe pas de norme pour définir la taille d'un SCO, et c'est au concepteur de la ressource d'apprentissage d'évaluer cette taille. Par ailleurs, il doit veiller à ce que ce SCO soit indépendant d'un contexte d'apprentissage afin d'être réutilisable, ou à définir cette réutilisation selon les contextes.

Le problème avec les actifs et les SCOs est la difficulté de les différencier sémantiquement à cause de leur définition plutôt vague. Considérons par exemple des actifs

représentés par plusieurs pages Web et un SCO sous forme de plusieurs pages Web. Quelle différence alors entre les deux ? (Zouaq, Nkambou, & Frasson, 2007b).

Les activités peuvent être vues comme des unités d'apprentissage intégrées. Une activité peut soit être composée de sous-activités ou prodiguer une ressource (SCO ou actif) à l'apprenant.

Les organisations de contenu : une organisation de contenu (*Content Organization*) est une représentation qui définit l'utilisation du contenu d'apprentissage sous forme d'unités d'apprentissage structurées représentées par les activités.

Les agrégations de contenu : elles constituent les structures qui permettent de rassembler les ressources d'apprentissage en un contenu d'enseignement sous forme de modules, cours etc. Elles peuvent contenir une ou plusieurs organisations de contenu.

Toutes les ressources du modèle de contenu SCORM peuvent être associées à des métadonnées qui décrivent leur contenu et facilitent leur recherche. Les métadonnées établissent une correspondance entre les éléments LOM (LOM, 2002) (*Learning Objects Metadata*) et chacun des composants du modèle de contenu SCORM.

SCORM définit également un environnement d'exécution (*SCORM Runtime Environment*) qui doit permettre d'assurer l'interopérabilité entre les contenus d'apprentissage SCO (*Sharable Content Object*) et les systèmes de gestion de l'apprentissage (*LMS*). Le sigle LMS (*Learning Management System*) désigne l'ensemble des fonctionnalités de présentation, de suivi, de production de rapports et de gestion d'un contenu d'apprentissage, des progrès des élèves et de leurs interactions. Selon SCORM, les contenus d'apprentissage doivent être interopérables avec des LMS multiples, quels que soient les outils utilisés pour créer le contenu. Afin d'assurer cette interopérabilité, il doit exister un moyen commun :

- permettant de lancer un contenu d'apprentissage ;
- permettant d'établir une communication entre ce contenu et un LMS ;

- permettant l'échange d'éléments d'information prédéfinis entre le LMS et le contenu lors de son exécution.

A cet effet, SCORM définit un ensemble de normes de communication qui se présentent sous forme d'une API (*Application Programming Interface*) et d'éléments de données standards qui permettent aux ressources d'apprentissage de communiquer avec l'environnement d'exécution durant la session d'apprentissage.

Il est à noter qu'ADL fournit un ensemble de ressources aux développeurs et aux concepteurs de cours afin de leur permettre de créer des objets d'apprentissage et des environnements conformes au standard. Plus particulièrement, ADL met à la disposition des usagers un environnement d'exécution exemple et des outils de tests de la conformité des ressources avec le standard.

5.2.2 IMS-LD

Parallèlement à SCORM, un autre standard, IMS-LD, s'impose de plus en plus dans le domaine des normes e-Learning. Plutôt que sur le contenu d'apprentissage, point central dans SCORM, IMS-LD se focalise plutôt sur la manière d'enseigner et de transmettre ce contenu. IMS-LD est une spécification de l'organisme IMS, qui a d'ailleurs mis au point divers standards et spécifications utilisés, entre autres, par SCORM et qui visent différents aspects de la formation en ligne : la modélisation de l'apprenant via la spécification LIP (*Learner Information Package*), les métadonnées (*Learning Resource Metadata Specification*), le packaging des ressources (*IMS Content Packaging*) et leur séquence (*IMS Sequencing*), etc.

De plus en plus répandu, IMS-LD a comme spécificité de présenter une sorte de méta-modèle pédagogique. Dérivant du travail de Koper (Koper, 2003), IMS-LD a intégré divers modèles pédagogiques et œuvre dans le même esprit que SCORM, à savoir, permettre la réutilisation et l'interopérabilité des ressources d'apprentissage tout en fournissant un cadre conceptuel d'ingénierie pédagogique.

Dans IMS-LD, le modèle de contenu contient les éléments suivants : l'unité d'apprentissage, les méthodes, les pièces et les actes.

L'unité d'apprentissage est un peu le pendant du « *content package* » dans SCORM. Elle rassemble l'ensemble des ressources et services d'apprentissage dans un dossier ZIP unique. L'unité d'apprentissage est constituée de **méthodes** qui, selon des conditions et des objectifs d'apprentissage donnés, sont déclenchées. Ces méthodes sont divisées en **pièces** et **actes** qui permettent de dérouler une ou plusieurs activités d'apprentissage. IMS-LD s'appuie sur la notion d'activité plutôt que sur des ressources.

IMS-LD permet également de modéliser la formation aussi bien à un niveau individuel qu'au niveau d'un groupe, ce qui n'est pas fait dans SCORM. IMS-LD propose des choix de modélisation par niveau : le niveau A définit des scénarios prescriptifs, le niveau B permet de personnaliser l'apprentissage et le niveau C propose des scénarios dynamiques (Burgos, Arnaud, Neuhauser, & Koper, 2005).

Différents environnements permettent l'édition et l'exécution d'un scénario IMS-LD, notamment Reload (Reload Project, 2004) ou CopperCore (CopperCore project website, 2007).

5.3 Critique et état des lieux

La formation en ligne s'est donc imposée dans le paysage de la formation par ordinateur. On peut lier cet essor au succès de la technique même : peu coûteuse par comparaison aux STI, ne nécessitant généralement qu'un navigateur Web pour le déploiement des cours, génératrice de revenus à grande échelle, la formation en ligne a accompagné l'essor du Web. On peut également lier cet essor à l'échec de techniques plus sophistiquées comme celles employées par les systèmes tutoriels intelligents qui ont souffert quant-à elles, du coût prohibitif de la création et de la maintenance de tels systèmes.

Toutefois, après quelques années d'utilisation, où la formation s'est focalisée sur les objets d'apprentissage et les métadonnées, en clair sur le contenu de formation, les acteurs du domaine (Ullrich, 2005) (Koohang & Harman, 2006) (Brooks & McCalla, 2006) (Fournier-Viger, Najjar, Mayers, & Nkambou, 2006) (Zouaq, Nkambou, & Frasson, 2007a) ont relevé plusieurs problèmes dans la philosophie même du e-Learning.

Tout d'abord, un objet d'apprentissage est une **boîte noire**. Il se présente comme un contenu intégré (*content package*) contenant toutes les ressources nécessaires à l'apprentissage. Des métadonnées permettent la description et la recherche des objets d'apprentissage. La nécessité d'avoir un langage commun de description a donné lieu à des standards, notamment IEEE LOM (*Learning Object Metadata*) (LOM, 2002) et Dublin-Core (Dublin Core Metadata Initiative, 1995). Ces métadonnées, bien que nécessaires, ne sont pas suffisantes. En effet, elles décrivent le monde autour de l'objet d'apprentissage (le langage utilisé, le contexte d'utilisation, l'auteur, etc.) et restent très vagues sur son contenu. S'il existe la possibilité de faire référence à une taxonomie du domaine dans LOM par exemple, une telle référence reste marginale dans les usages et ne permet que d'indiquer certains concepts et sûrement pas l'ensemble du contenu de l'objet. Par ailleurs, lorsque cela est fait, l'ontologie utilisée est construite manuellement par des experts, séparément du contenu des objets d'apprentissage, et on tente d'apparier le contenu de l'objet d'apprentissage avec l'ontologie créée.

Avec les seules métadonnées actuelles, il apparait clairement que le contenu d'un objet d'apprentissage est inaccessible à un agent logiciel. Concrètement, l'objet d'apprentissage ne dispose d'aucune connaissance du domaine et d'aucune expertise pédagogique. En effet, si les objets d'apprentissage mettent bien en œuvre une méthode pédagogique fournie par le concepteur de l'objet, celle-ci demeure implicite. Les objets d'apprentissage restent donc tributaires de cette méthode et ne peuvent la modifier si des difficultés d'apprentissage surgissent.

Enfin, les objets d'apprentissage souffrent de leur aspect statique et du manque ou du peu d'adaptation de leur contenu à un modèle de l'apprenant. C'est pourquoi diverses recherches se sont attachées à cette question et le e-Learning a donné lieu, de nos jours, à diverses normes pour la modélisation de l'apprenant telles que IMS LIP (*Learner Information Package*) (IMS Learner Information Package Specification, 2001) ou IMS ePortfolio (IMS ePortfolio Specification, 2007). Des normes ont également émergé pour la modélisation de scénarios pédagogiques notamment IMS-LD (*IMS Learning Design*) (IMS-LD, 2007), etc. Toutefois, ces normes restent insuffisantes pour répondre aux besoins de la formation par ordinateur actuelle : mettre au point des techniques qui s'appuient sur la modélisation du domaine d'apprentissage, sur la modélisation des théories d'apprentissage et sur la modélisation de l'apprenant. On peut ainsi voir, à travers la description du standard SCORM, que les mêmes critiques auparavant indiquées pour les systèmes e-Learning s'applique aux environnements SCORM : absence de la notion de connaissance du domaine, d'expertise pédagogique et de modélisation réelle de l'apprenant. Par ailleurs, bien qu'IMS-LD s'intéresse à l'ingénierie pédagogique, il se limite à un modèle (défini par le standard) et ne prétend pas intégrer des théories pédagogiques provenant de l'éducation. Les autres critiques relatives au domaine et à la modélisation de l'apprenant restent également valables pour ce standard.

A l'inverse du e-Learning, les STI se sont dès le départ intéressés à ces problématiques. De manière générale, les STI se distinguent par un modèle de connaissances très riche qui leur permet d'adapter leur enseignement aux besoins précis de l'apprenant. Ils sont dotés de stratégies pédagogiques pour rendre cet enseignement le plus efficace possible. Toutefois, ils souffrent souvent d'une architecture, de langages et de composants propriétaires ce qui limite drastiquement leur réutilisation et augmente leur coût de production. L'avènement du Web sémantique devrait toutefois aider à remédier à cet état de fait. A l'inverse, les systèmes d'e-Learning ont été conçus pour faciliter la réutilisation de leurs ressources. Dotés d'un haut degré de granularité, les objets d'apprentissage se présentent sous forme de boites noires où les stratégies pédagogiques

sont implicitement encodées par le concepteur de l'objet. Il n'y a pas de séparation entre les connaissances du domaine et les connaissances pédagogiques. Le e-Learning s'appuie sur des métadonnées pour annoter les objets d'apprentissage et les réutiliser, une activité très peu pratiquée par les STI. Le Tableau XXIX synthétise ces différences.

STI	E-Learning
◆ Fin degré de granularité des connaissances, riche représentation ◆ Adaptation de l'enseignement selon un modèle de l'apprenant ◆ Stratégies pédagogiques ◆ Composants propriétaires	◆ Haut degré de granularité ◆ Peu d'adaptation aux besoins de l'apprenant ◆ Peu ou pas de stratégies pédagogiques explicites ◆ Réutilisation des objets d'apprentissage

Tableau XXIX. Comparaison entre les STI et les systèmes de e-Learning

Le but de cette thèse est donc bien de revenir aux objectifs des STI en offrant des outils et une méthodologie permettant de répondre à ces besoins tout en évitant les inconvénients des STI, à savoir : un langage de représentation des connaissances propriétaire, et une nécessité de produire toute la connaissance de manière manuelle. Cela doit permettre également d'utiliser les mêmes sortes d'outils et de langages par les deux types de systèmes et s'appuie sur l'adoption des techniques et idées issues du Web sémantique (Berners-Lee, Hendler, & Lassila, 2001). Dans les sections suivantes, nous brossons un tableau de l'état de l'art des techniques du Web sémantique appliquées à la formation.

5.4 Apport du Web sémantique au domaine de la formation

Les limites du e-Learning et l'avènement du Web sémantique ont conduit la communauté de recherche à recentrer ses travaux sur la nécessité de créer des représentations plus complexes que de simples agrégations de contenu, sur la nécessité d'adapter les formations à des apprenants particuliers ou encore sur la nécessité de doter les

systèmes de représentations de connaissances et de mécanismes de raisonnement à même d'exploiter les connaissances. De plus en plus de travaux reliés au Web sémantique ont été appliqués au domaine de la formation, donnant naissance à la notion de Web sémantique éducationnel (*Educational Semantic Web*) (Aroyo & Dicheva, 2004). De manière générale, ces recherches se sont attaquées à divers aspects de la formation :

1. La représentation des connaissances dans les EIAH ;
2. La personnalisation de l'apprentissage et la modélisation du modèle de l'apprenant ;
3. L'annotation et la recherche de contenus d'apprentissage ;
4. La création de ressources et de services éducationnels dynamiques ;

Ces divers aspects tentent de répondre, en partie, à l'ensemble des problématiques évoquées jusqu'à maintenant. Les sections suivantes introduisent les différents travaux utilisant le Web sémantique pour améliorer la représentation, l'indexation et la composition de ressources d'apprentissage.

5.4.1 La représentation des connaissances par des ontologies

5.4.1.1 La représentation des connaissances du domaine

L'ingénierie des connaissances est une discipline qui a toujours tenu une place importante dans les EIAH dits « intelligents ». Avec la maturation des technologies et des langages disponibles, les ontologies se sont imposées en tant que représentations de connaissances à part entière (Devedžić, 2004). Étant donné la place centrale du contenu et de la connaissance dans la formation, les ontologies permettent de formaliser et d'expliciter ce contenu d'apprentissage de manière à ce qu'il soit transmissible à un apprenant.

(Krdzavac, Gasevic, & Devedzic, 2004) soulignent d'ailleurs l'intérêt des logiques de description pour les systèmes de formation dits intelligents, notamment pour les capacités d'inférences qu'elles offrent par rapport à d'autres formalismes de représentation du raisonnement. Ils montrent notamment comment un modèle en logique de description

peut servir de support à un système d'explication. (Suraweera, Mitrovic, & Martin, 2004) décrivent également une méthodologie pour l'utilisation d'ontologies du domaine dans les STI basés sur les contraintes (*Constraint-based modeling*). Si l'intérêt des ontologies du domaine pour formaliser la connaissance a été relevé par plusieurs chercheurs, peu de travaux se sont toutefois attaqués, dans cette communauté, à des méthodologies d'acquisition semi-automatiques d'une ontologie du domaine. C'est à ce niveau que se situe tout l'intérêt de l'outil TEXCOMON précédemment présenté.

5.4.1.2 La représentation du modèle de l'apprenant

Toutefois, les ontologies ne se limitent pas à la modélisation de la connaissance du domaine. Elles représentent également le modèle de l'apprenant (Lougheed, Bogyo, Brokenshire, & Kumar, 2005) et comprennent alors des informations sur le profil de l'apprenant, son style d'apprentissage, ses préférences, etc. Par exemple, (Brooks, McCalla, & Winter, 2005) proposent d'annoter un objet d'apprentissage avec les instances du modèle de chaque apprenant ayant eu une interaction avec cet objet d'apprentissage. (Dolog & Nejdl, 2003), quant-à eux, proposent une modélisation du modèle de l'apprenant basé sur RDF (RDF/XML Syntax Specification, 2004) et un langage de requête sur ce modèle. Pareillement, (Dolog & Schaefer, 2005) proposent une ontologie du modèle de l'apprenant basée sur différents standards (PAPI, IMS RDCEO, IMS LIP, IMS QTI) et implantent une API (sous forme Java et services Web) pour interroger le modèle. Ce modèle et cette API peuvent être utilisés par des systèmes différents pour interroger et mettre à jour un modèle de l'apprenant de manière collaborative. Les auteurs décrivent aussi un mécanisme d'importation du modèle de l'apprenant d'un système à l'autre mais ne traitent pas vraiment des conflits qui peuvent résulter d'un tel mécanisme.

(Sosnovsky, Dolog, Henze, Brusilovsky, & Nejdl, 2007) exposent une méthode originale d'initialisation du modèle de l'apprenant basée sur l'appariement des ontologies du domaine : on utilise le modèle existant de l'apprenant sur un domaine donné pour initialiser son modèle dans un autre domaine relié mais différent (même domaine mais

ontologies différentes, ou domaine différent ayant certaines similarités de contenu). En fait, on essaie d'inférer les connaissances déjà connues au moyen de recoupements entre les deux domaines considérés. Cette approche est un premier pas vers des approches automatiques pour l'appariement de modèles de l'apprenant différents.

5.4.1.3 La représentation du modèle pédagogique

Le modèle pédagogique est un autre aspect abordé par l'ingénierie ontologique notamment en offrant une **représentation des rôles pédagogiques** comme l'ontologie des rôles pédagogiques présentée dans (Ullrich, 2004) (Zouaq, Nkambou, & Frasson, 2007b), et en offrant une **modélisation des théories et modèles pédagogiques** (Zouaq, Nkambou, & Frasson, 2007a) (Bourdeau, Mizoguchi, Psyché, & Nkambou, 2004) (Psyché, Bourdeau, Nkambou, & Mizoguchi, 2005). Concernant ce dernier point, les travaux de R. Mizoguchi et de ses collègues s'attachent à modéliser les théories pédagogiques et à les intégrer dans des systèmes dits conscients de leurs théories (*theory-aware*). D'autres initiatives tentent de représenter la modélisation pédagogique (*learning design*) sous forme ontologique tels que les travaux de (Knight, Gasevic, & Richards, 2006) ou (Paquette, 2004).

Notons que ces trois représentations (domaine, apprenant et pédagogique) permettent de personnaliser l'apprentissage en tenant compte non seulement de l'historique de l'apprenant, mais également des ressources et stratégies pédagogiques disponibles. Bien souvent, les travaux de l'état de l'art lient l'aspect de la personnalisation de l'apprentissage au modèle de l'apprenant seul, mais nous pensons que ces trois dimensions sont essentielles et nous proposons, dans le chapitre suivant, une démarche intégrant toutes ces dimensions.

5.4.2 L'annotation sémantique des ressources d'apprentissage

De manière générale, le Web sémantique a permis l'émergence des ontologies comme ressources fondamentales pour l'indexation des objets d'apprentissage. Par ailleurs, l'augmentation continue du nombre de ressources d'apprentissage nécessite de mettre au point des mécanismes pour les indexer et les rechercher, non seulement par des humains,

mais également par des agents logiciels. Dans ce cadre, de plus en plus de travaux se sont penchés sur l'annotation sémantique des ressources d'apprentissage en utilisant des ontologies. L'annotation consiste à, d'une part, construire une ontologie et d'autre part, à décrire ces ressources au moyen de cette ontologie.

Divers angles peuvent être abordés pour annoter les ressources. (Stojanovic, Staab, & Studer, 2001) recensent 3 dimensions essentielles pour l'annotation des ressources : des métadonnées pour décrire le contenu des objets d'apprentissage, des métadonnées sur leur contexte (en termes de rôles pédagogiques) et enfin des métadonnées sur leur structure. En ce qui nous concerne, nous avons, dans cette thèse, dénombré 4 angles (la dimension compétences venant s'ajouter aux dimensions précitées) dans la littérature :

- **un angle compétences** : les compétences servent à définir un objectif d'apprentissage. Les ressources annotées peuvent permettre une indexation en fonction d'objectifs d'apprentissage (Zouaq, Nkambou, & Frasson, 2007g) (Paquette, 2007).
- **un angle domaine** : ainsi que précédemment décrit, une annotation domaine permet d'expliciter le contenu des ressources d'apprentissage (Gasevic, Jovanovic, & Devedzic, 2004) (Zouaq, Nkambou, & Frasson, 2007d).
- **un angle structure** : cela permet de déterminer les portions de documents et donc permet une indexation plus fine (Bergsträßer, Rensing, Zimmermann, & Steinmetz, 2007) (Zouaq, Nkambou, & Frasson, 2006a).
- **un angle pédagogique** : une ontologie de rôles pédagogiques permet de distinguer ces rôles dans les contenus d'apprentissage. Cela permet de rechercher des ressources de type pédagogique précis (Verbert, Jovanovic, Duval, & Gaševic, 2006).

De nombreux travaux ont utilisé des ontologies pour indexer les ressources d'apprentissage sous l'un ou plusieurs de ces différents angles (mais pas forcément dans leur ensemble). (Gasevic, Jovanovic, & Devedzic, 2004) ont souligné l'importance de l'annotation de contenus d'apprentissage en utilisant des ontologies du domaine. (Dehors,

Faron-Zucker, Giboin, & Stromboni, 2005) ont, de leur côté, présenté une approche d'annotation semi-automatique qui se base sur les caractéristiques structurelles d'un objet d'apprentissage, en se fondant sur l'idée qu'une décomposition structurelle permet de découvrir la stratégie pédagogique du concepteur de l'objet. Des annotations finales validées par un expert humain sont exprimées en termes de rôles pédagogiques. Les travaux de (Verbert, Jovanovic, Duval, & Gaševic, 2006) ont permis de définir une ontologie (ALOCOM) pour représenter la structure de documents en fonction d'un modèle dérivant de l'architecture DITA (Priestley, 2001).

Par ailleurs, d'autres approches ont tenté de générer des métadonnées selon des standards. Ainsi, par exemple, (Cardinaels, Meire, & Duval, 2005) présentent un outil à base de services Web pour générer des métadonnées LOM de manière automatique. A cet effet, les auteurs utilisent des techniques d'analyse de contenu et de contexte et s'appuient également sur des techniques de traitement du langage naturel. Des techniques plus sophistiquées sont utilisées dans la version subséquente du logiciel (Meire, Ochoa, & Duval, 2007). Enfin l'outil TANGRAM (Jovanovic, Gasevic, & Devedzic, 2006b) (Jovanovic, Gasevic, & Devedzic, 2006a) est un environnement d'apprentissage intégré qui utilise une architecture à base d'ontologies pour modéliser le domaine, la structure de document, le modèle de l'apprenant, les rôles pédagogiques et qui annote les ressources avec des métadonnées en fonction de ces ontologies.

5.4.3 L'agrégation automatique de ressources d'apprentissage

Parallèlement à l'annotation des objets d'apprentissage, d'autres travaux ont exploité l'indexation fournie non seulement pour retrouver les objets d'apprentissage dans leur entièreté dans les entrepôts, mais également pour agréger automatiquement de nouvelles ressources, plus à même de répondre à des besoins précis de l'apprenant.

La composition de ressources d'apprentissage pour un besoin précis et à un moment précis (*Just in time, Just enough learning*) est un des défis récurrents de différentes communautés de recherche du e-Learning (Zouaq, Nkambou, & Frasson, 2006b). Ainsi par

exemple, le projet SeLeNe (Keenoy, et al., 2004) décompose des documents DocBook en fragments. Ces fragments servent à composer des objets d'apprentissage sous forme de séquences RDF en exploitant les liens sémantiques inférés à partir du processus de décomposition des documents et des métadonnées déjà présentes dans le document. (Buffa, Dehors, Faron-Zucker, & Sander, 2005) quant-à eux décomposent également des livres électroniques LATEX ou WORD en unités élémentaires et les annotent de manière manuelle en fonction de rôles pédagogiques et de mots-clés. Là encore, ce sont les métadonnées qui servent ensuite à constituer un scénario d'apprentissage personnalisé. Pareillement, le projet IMAT (Desmoulins & Grandbastien, 2002) s'attaque à la réutilisation de manuels de formation techniques. L'indexation et la décomposition suit la table des matières des manuels et correspond aux structures logiques du document (sections, tables, images). Par la suite, le document est annoté en fonction du domaine et de la pédagogie.

Au Canada, on peut citer le réseau de recherche LORNET qui s'est intéressé à cette problématique et a développé des outils et méthodologies d'ingénierie pédagogique et d'agrégation d'objets d'apprentissage (Ingénierie pédagogique et agrégation des objets d'apprentissage, 2007).

D'autres projets, notamment (Verbert, Jovanovic, Duval, & Gaševic, 2006), ont proposé un « nouveau » modèle de contenu basé sur une ontologie (ALOCOM) représentant la structure des objets d'apprentissage (*Content Fragments, Content Objects, Learning objects*) (Verbert, Klerkx, Meire, Najjar, & Duval, 2004). Le but de cette ontologie est de faciliter la réutilisation d'objets d'apprentissage et de portions de ces objets pour créer de nouvelles ressources. L'idée est implantée avec des documents Powerpoint où chaque diapositive est décomposée en segments. Ces segments peuvent être utilisés par un concepteur de cours pour composer de nouvelles ressources. L'idée de représenter des rôles pédagogiques servant à la création de nouvelles ressources est certes très intéressante (Ullrich, 2004) (Ullrich, 2005), toutefois, on voit mal, dans le cadre de l'ontologie ALOCOM, la valeur ajoutée du modèle de contenu proposé, si on le compare par exemple

avec le modèle de contenu SCORM. En fait, les auteurs disent présenter un modèle de contenu abstrait qui puisse en englober plusieurs (*SCORM* (SCORM, 2007), *Learnativity* (Wagner, 2002), *CISCO* (Barritt, Lewis, & Wieseler, 1999) et *Netg* (L'allier, 1997)). Notre critique porte sur le fait que cette comparaison pourrait très bien être faite au profit du modèle de contenu SCORM où chaque élément peut également représenter les éléments des autres modèles (voir Tableau XXX).

ALOCOM	***SCORM***
Fragments de contenu (Content Fragments)	Actifs (assets)
Objets de contenu (Content Objects)	Objets de contenu partageables (Shareable Content Objects)
Objets d'apprentissage (Learning objects)	Agrégations de contenu (Content Agregation)

Tableau XXX. Correspondance entre les éléments d'ALOCOM et du modèle de contenu SCORM

Enfin, d'autres recherches ont proposé des approches moins directives pour la composition de ressources. TANGRAM (Jovanovic, Gasevic, & Devedzic, 2006b), par exemple, permet à un apprenant de sélectionner une partie de l'ontologie du domaine qui l'intéresse. Le système recherche les objets d'apprentissage reliés à cette ontologie en se basant essentiellement sur des métadonnées concernant le sujet de l'objet, les relations hiérarchiques et les relations d'ordre ainsi que sur le modèle de l'apprenant (préférences, style d'apprentissage, historique d'apprentissage).

En résumé, on peut dire que les ontologies tiennent une place grandissante dans les recherches récentes. Néanmoins, la majorité des approches optent pour des ontologies produites manuellement. Par ailleurs, bien que l'idée de décomposition et de réutilisation de ressources existantes (manuels, documents divers) semble s'être imposée, elle s'appuie essentiellement sur la structure première de la ressource (la table des matières dans IMAT par exemple) ou sur des métadonnées, là encore souvent définies manuellement. Quand la

réutilisation des ressources est partielle, comme lorsque des rôles pédagogiques sont identifiés dans les documents, ces derniers sont assemblés manuellement par un concepteur humain, qui a à sa disposition des outils de recherche de ressources.

L'intérêt de telles approches est grand, par rapport à l'existant, mais il est amoindri par le manque de mécanismes automatiques d'agrégation ou de composition de ressources d'apprentissage. Nous pensons qu'une telle automatisation doit non seulement s'appuyer sur des ressources indexées et annotées par des ontologies, mais qu'elle doit nécessiter moins d'interventions manuelles. C'est la raison pour laquelle nous nous intéressons aux techniques du traitement de la langue naturelle. Nous avons déjà montré comment ces techniques sont exploitées pour la génération (semi) automatique d'une ontologie du domaine. Un bref aperçu de l'utilisation de telles techniques dans les EIAH peut nous mettre au fait des besoins encore insatisfaits du domaine.

5.5 Les techniques de traitement de la langue naturelle dans les EIAH

Nous avons déjà vu que le traitement du langage naturel sert généralement, dans la communauté e-Learning, à la génération de métadonnées pour les objets d'apprentissage (Meire, Ochoa, & Duval, 2007) ou à la détection de rôles pédagogiques (Jovanovic, Gasevic, & Devedzic, 2006b).

Nous nous sommes également intéressés à l'emploi des techniques du langage naturel dans les systèmes tutoriels intelligents. Généralement, ces techniques ont servi à diversifier les modes de communication avec l'apprenant, à les rendre plus naturels et surtout à permettre la compréhension des réponses de l'apprenant formulée en langue naturelle. Beaucoup de travaux se sont penchés sur la génération de dialogues dans les STI (Di Eugenio, Fossati, Yu, Haller, & Glass, 2005) (Graesser, VanLehn, Rosé, Jordan, & Harter, 2001) et surtout sur la compréhension de dialogues formulés en langage naturel. C'est le cas de (Jordan, Makatchev, & VanLehn, 2004) qui ont utilisé plusieurs techniques

du traitement de la langue naturelle (symbolique, statistique, hybride) pour comprendre des explications fournies par l'apprenant en langue naturelle. Les techniques du traitement du langage naturel servent donc plutôt au niveau du modèle de l'apprenant. A notre connaissance, peu ou pas de travaux ont abordé ces techniques dans le but d'acquérir un modèle du domaine, ainsi que nous l'avons présenté dans le chapitre 3. La même critique peut être adressée au forage de données éducationnelles (*Educational Data Mining*) où l'apprenant et son modèle reste l'élément central à étudier (Educational Data Mining, 2007).

5.6 En résumé

Ce chapitre a été consacré à l'étude de l'état de l'art dans le domaine des EIAH et a focalisé sur l'apport des ontologies dans les différents modèles (apprenant, domaine, pédagogique) ainsi que sur les travaux visant à réutiliser, exploiter, indexer et composer des ressources d'apprentissage pour répondre à un besoin précis. Nous avons formulé quelques critiques sur les différents travaux et avons souligné certaines spécificités de cette thèse. Notons également que la majorité des approches se sont davantage préoccupées des systèmes e-Learning que des STI.

Nous abordons, dans le chapitre suivant, l'architecture conceptuelle de notre projet « *The Knowledge Puzzle* » qui englobe non seulement des outils d'acquisition des connaissances incluant l'outil TEXCOMON, mais offre également des outils d'exploitation des connaissances dans un contexte de formation. Cette architecture s'appuie sur les ontologies pour former sa base de connaissances et tente de répondre aux questions suivantes : Quels peuvent être les impacts des techniques d'intelligence artificielle utilisées par les STI sur le domaine de la formation en ligne ? Comment faire en sorte de combiner les avantages du e-Learning et des STI ?

6 Une vue d'ensemble du projet «The Knowledge Puzzle»

Le projet « *The Knowledge Puzzle* » a vu le jour pour tenter de répondre aux questions présentées dans la fin du chapitre précédent et surtout pour former un pont entre les communautés e-Learning et STI de manière à ce que les deux puissent bénéficier de leurs avantages respectifs et envisager des développements communs futurs (Zouaq, Nkambou, & Frasson, 2007f). L'objectif est de présenter un modèle qui fasse la synergie des deux systèmes en permettant aux STI de réutiliser des objets d'apprentissage et en produisant des objets d'apprentissage plus « intelligents » utilisables par des systèmes de e-Learning.

Nous croyons que cela peut se réaliser au moyen d'une base de connaissances commune. Cette base de connaissances doit permettre de créer des objets d'apprentissage dotés des caractéristiques suivantes :

- ils doivent disposer d'un modèle du domaine qu'ils sont sensés transmettre ;
- ils doivent séparer la connaissance du domaine de la connaissance pédagogique ;
- ils doivent pouvoir s'adapter aux besoins de l'apprenant.

« *The Knowledge Puzzle* » est une architecture intégrée qui s'appuie sur diverses ontologies pour créer et annoter des ressources d'apprentissage selon divers aspects, et qui permet, et c'est le but principal, de composer automatiquement de nouvelles ressources d'apprentissage. Ces ressources doivent pouvoir pallier les limites des objets d'apprentissage existants à savoir l'aspect de boite noire tant au niveau domaine que pédagogique et l'aspect statique. Ces objets de connaissances et d'apprentissage doivent être également exploitables par des STI, qui peuvent bénéficier de notre modèle de deux manières : soit en exploitant directement les objets d'apprentissage, soit en exploitant les différentes ontologies proposées comme base de connaissances à part entière. En effet,

alors que les systèmes de e-Learning disposent d'entrepôts d'objets d'apprentissage, les STI, ainsi que nous l'avons précédemment expliqué, ne favorisent pas la réutilisation de leurs connaissances de par leur structure propriétaire. Or vu le nombre croissant d'entrepôts d'objets d'apprentissage, ces ressources ne peuvent continuer à être ignorées par les systèmes tutoriels intelligents (Zouaq, Nkambou, & Frasson, 2007e).

Par ailleurs, étant donné que notre base de connaissances commune doit également servir aux systèmes d'e-Learning, il importe de se préoccuper des standards de la formation en ligne et de comprendre comment ces standards pourraient intégrer une telle base de connaissances.

6.1 Proposition d'une architecture commune aux EIAH

Dans les chapitres précédents, nous avons pu constater que généralement, les EIAH (Environnements Informatiques pour l'Apprentissage Humain) dits «intelligents» suivent dans leur ensemble l'architecture traditionnelle des systèmes tutoriels intelligents. Autrement dit, ils possèdent un modèle du domaine, un modèle de l'apprenant et mettent en œuvre des stratégies pédagogiques.

Nous proposons une vision « Web sémantique » de cette architecture, qui consiste à représenter les différents modèles figurant dans l'architecture d'un STI traditionnel par des ontologies (Figure 31).

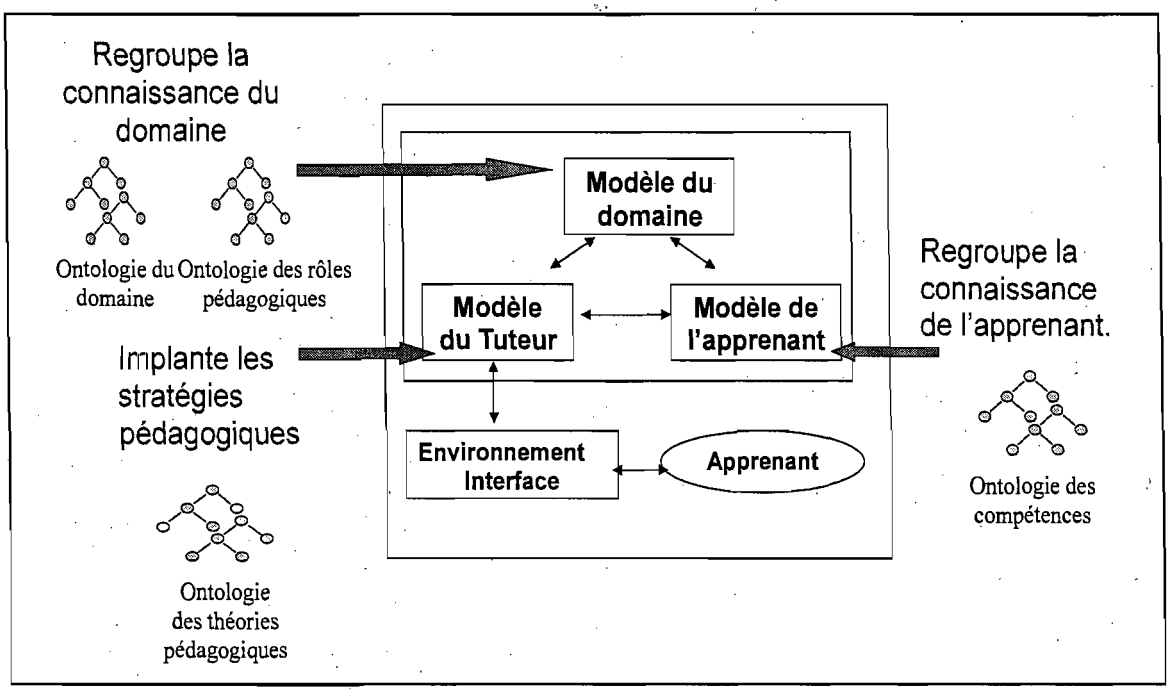


Figure 31. Une vue ontologique de l'architecture d'un STI

Dans cette architecture, nous proposons que le modèle du domaine soit représenté par une ontologie du domaine et une ontologie des rôles pédagogiques pour modéliser le contenu sur lequel porte la formation. Le modèle du tuteur est doté d'une ontologie des théories pédagogiques qui lui permet de présenter le même contenu selon différentes stratégies pédagogiques et enfin le modèle de l'apprenant peut être exprimé sous forme de compétences représentant les objectifs de formation. Nous reviendrons sur chacune de ces ontologies en détail un peu plus loin. On pourrait noter à ce niveau l'absence du modèle expert dans l'architecture du STI. Ce modèle expert, chargé de la connaissance procédurale et de la résolution de problèmes est en effet le seul modèle que nous n'avons pas traité au niveau de notre architecture (nous avons indiqué préalablement que cette thèse se préoccupe essentiellement de la connaissance déclarative).

Par ailleurs, notre approche vise à proposer une alternative aux entrepôts d'objets d'apprentissage. Le nombre croissant de ressources a souligné les faiblesses du stockage de contenus d'apprentissage classiques et de leur annotation par métadonnées (Brooks & McCalla, 2006) :

Premièrement, les métadonnées sont conçues avec l'idée qu'elles sont destinées à des humains et produites par des humains. Le problème avec cette vision est que tout d'abord elle est coûteuse en temps et qu'ensuite, les experts se contentent bien souvent de remplir quelques champs de métadonnées mais en négligent une bonne partie. Par ailleurs, étant donné la subjectivité inhérente aux humains, les métadonnées souffrent d'un manque de cohérence et d'un manque de validité sémantique. Une automatisation au moins en partie de ce processus est donc nécessaire.

Deuxièmement, les formats de métadonnées existants limitent les possibilités d'exprimer certaines interactions entre les EIAH et les objets d'apprentissage. Cela est d'ailleurs un des inconvénients reliés aux standards de façon générale.

Dans notre thèse, les entrepôts d'objets d'apprentissage ne sont plus destinés à être exploités par les systèmes de formation comme fournisseurs d'objets d'apprentissage : les entrepôts sont exploités par des services tutoriels comme fournissant la matière première de l'enseignement. Ces services sont ensuite utilisés pour créer des ontologies du domaine qui indexent automatiquement les contenus, ainsi que pour annoter les objets selon différents contextes (rôles pédagogiques, compétences, etc.). Ces contextes sont ensuite réutilisés pour créer des ressources d'apprentissage dynamiquement.

La Figure 32 présente l'architecture ontologique d'un système tutoriel dont les ressources proviennent d'entrepôts d'objets d'apprentissage.

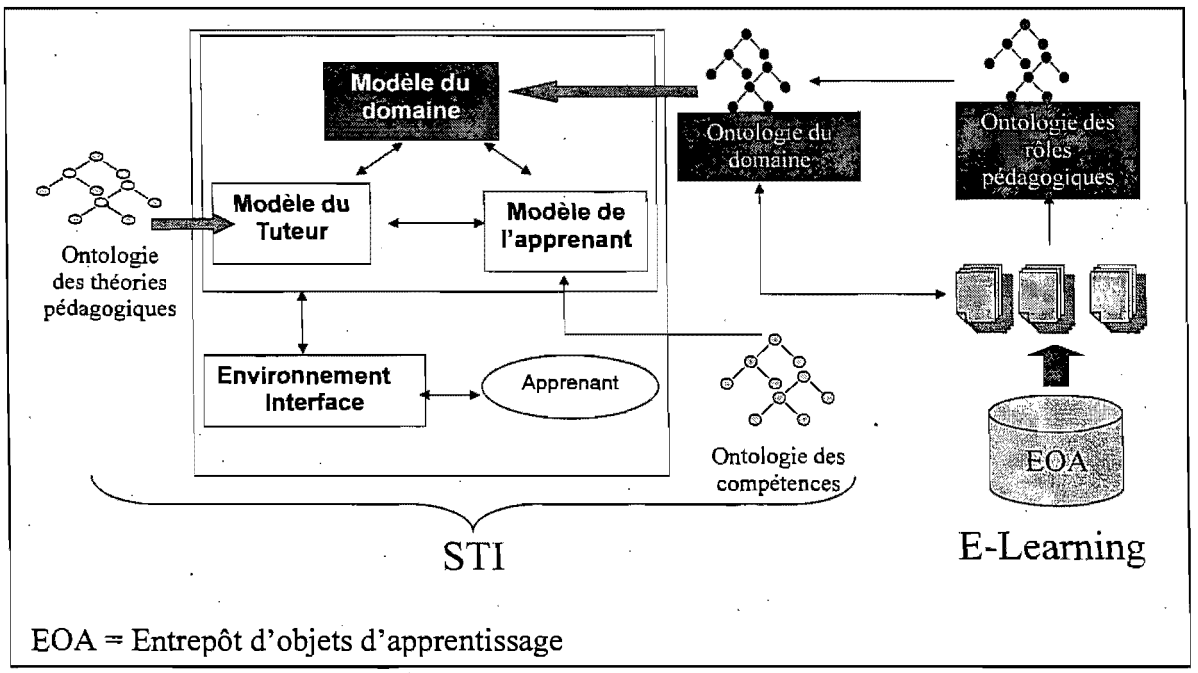


Figure 32. Un pont entre les STIs et les systèmes d'e-Learning

Dans ce qui suit, nous présentons les différents composants de l'architecture et surtout les méthodes employées pour la concrétiser. L'architecture se scinde en deux composantes principales : un module d'acquisition des connaissances et un module d'exploitation des connaissances. Nous nous focalisons dans la prochaine section sur l'acquisition des connaissances nécessaires pour la mise en place du système proposé.

6.2 Architecture du module d'acquisition des connaissances

Dans le système «*The Knowledge Puzzle*», l'architecture du module d'acquisition des connaissances vise à implanter des mécanismes d'extraction de connaissances (outils auteurs) à partir d'objets d'apprentissage (textuels) provenant d'entrepôts. Les outils auteurs permettent d'extraire des connaissances dans les documents en entrée et de les stocker dans une **mémoire organisationnelle** (MO) (plutôt que dans un entrepôt d'objets d'apprentissage).

6.2.1 Une mémoire organisationnelle à base d'ontologies

Pourquoi parler de mémoire plutôt que simplement de base de connaissances ou de base de données ? En fait, le concept de mémoire organisationnelle s'est imposé dans le monde des entreprises et désigne non seulement les connaissances mais les processus menant à l'intégration de ces connaissances de manière à former un tout cohérent.

C'est une vision similaire qui prévaut dans les sciences cognitives et les neurosciences en ce qui a trait au processus d'apprentissage. La mémoire humaine est composée d'un ensemble de connaissances inter-reliées. Le souci de créer une mémoire organisationnelle en remplacement des entrepôts d'objets d'apprentissage répond à ce besoin de stockage de fragments de connaissances dynamiques et dynamiquement reliés. Le souci de structurer cette mémoire sous forme ontologique répond également au souci de capacités d'inférence et donc de connaissances constamment mises à jour, réassemblées, et revérifiées en contexte. Par ailleurs, les ontologies offrent de par leur structure une terminologie commune nécessaire, ce qui a déjà été développé dans les chapitres précédents.

Quelques travaux dans le domaine de l'éducation tels que (Abel, Benayache, Lenne, Moulin, Barry, & Chaput, 2004) (Benayache, 2005) ont parlé de mémoire organisationnelle pour l'apprentissage (*Learning Organizational Memory*) orientée vers le stockage de contenu pédagogique indexé par des notions à acquérir et des liens entre ces notions, mais la majorité des travaux sur les mémoires organisationnelles sont reliés à la gestion de connaissances en entreprise (Gandon, 2002) (Van Elst & Abecker, 2002).

Dans notre thèse, la mémoire organisationnelle est composée de ressources, d'ontologies et de règles. Les ressources ne sont pas des agrégations de contenus comme dans un entrepôt d'objets d'apprentissage, mais des composants granulaires (rôles pédagogiques, portions de documents) nécessaires à l'agrégation automatique de ressources d'apprentissage.

Les ontologies de la mémoire organisationnelle se décomposent en axiomes définissant les classes et relations de l'ontologie (*T-BOX*) et en instances ou ressources (*A-BOX*). Elles permettent de soutenir le processus de composition automatique de ressources d'apprentissage plus intéressantes que des objets d'apprentissage dit traditionnels. Elles englobent une ontologie des compétences (CMP-ONTO), une ontologie de la structure des documents (DOC-ONTO), une ontologie du domaine (DOM-ONTO), une ontologie des rôles pédagogiques (IRO-ONTO) et enfin une ontologie des théories pédagogiques (ILT-ONTO). Toutes ces ontologies sont introduites et expliquées dans les sections suivantes. Enfin les règles de la mémoire organisationnelle servent à relier des compétences et des théories d'apprentissage à des ressources d'apprentissage pertinentes

Des outils d'interrogation et de mise à jour de la mémoire organisationnelle sont mis à la disposition de l'usager. Ce dernier peut parcourir les différentes classes de la mémoire et mettre à jour leurs instances au travers d'un navigateur d'ontologies (Figure 33).

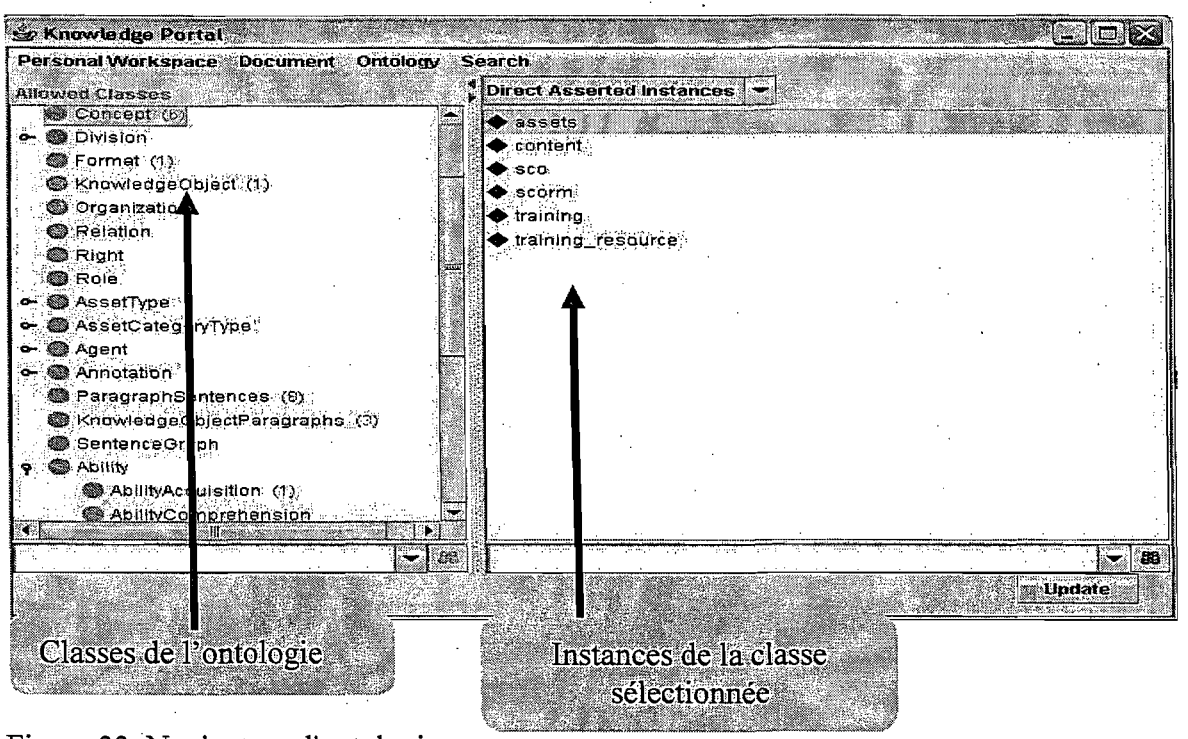


Figure 33. Navigateur d'ontologie

La figure suivante (Figure 34) schématise l'architecture générale du module d'acquisition des connaissances. Ce module est composé de la mémoire organisationnelle, et de deux suites logicielles : ONTO-AUTHOR et ONTO-ENGINE. Ces deux suites permettent de créer le contenu de la mémoire organisationnelle en exploitant le corpus documentaire en entrée.

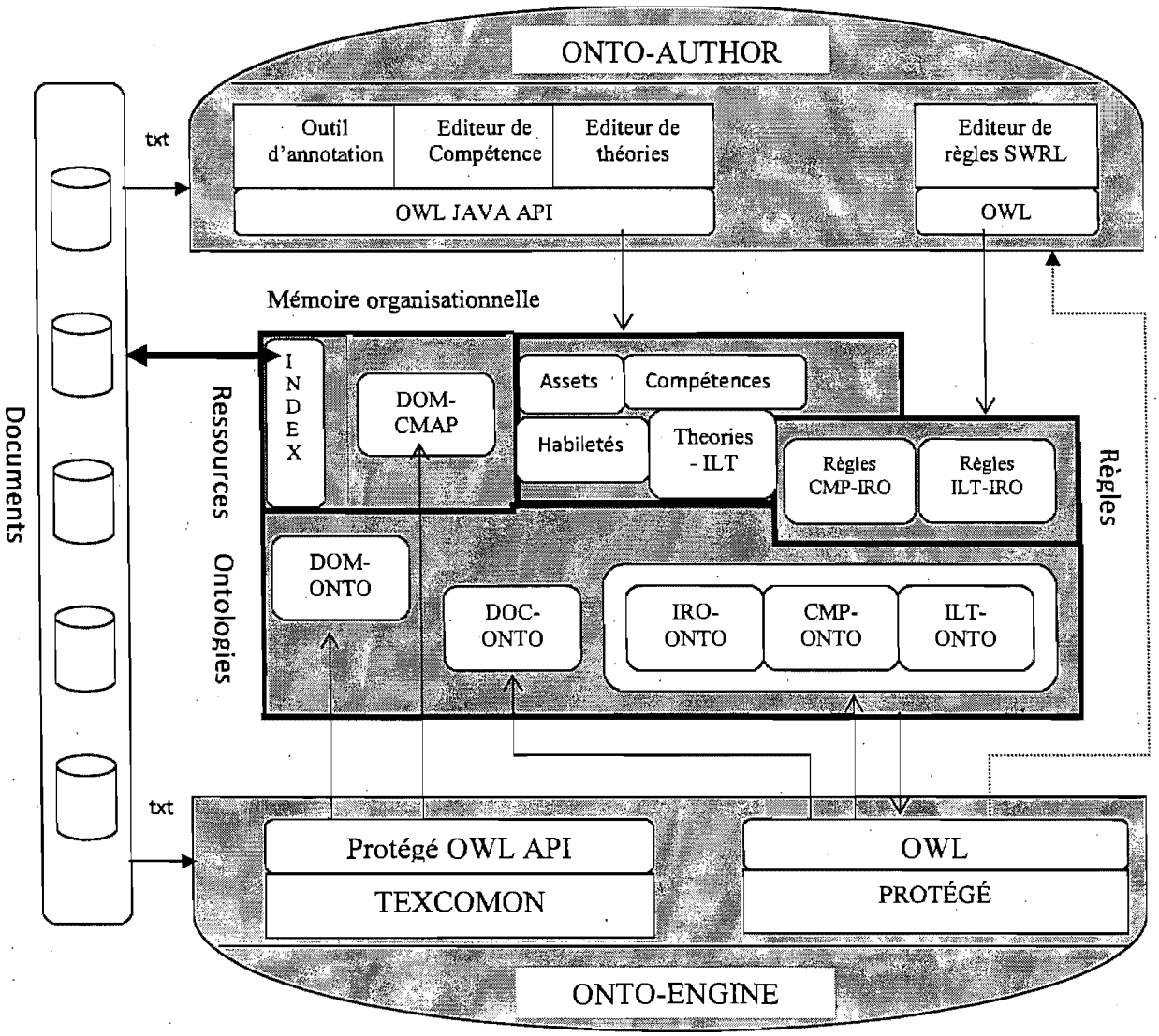


Figure 34. Architecture du module d'acquisition des connaissances

La légende suivante permet d'expliquer un peu mieux la figure 34.

- **TEXCOMON** : L'outil TEXCOMON permet l'acquisition automatique de cartes de concepts et d'une ontologie du domaine.
- **DOM-ONTO** : représente l'ontologie du domaine
- **DOM-CMAP** : représente les cartes de concepts
- **DOC-ONTO** : représente l'ontologie de structure de documents. Elle stocke les paragraphes, phrases et autres représentations structurelles.
- **IRO-ONTO** : représente l'ontologie des rôles pédagogiques
- **ILT-ONTO** : représente l'ontologie des théories pédagogiques
- **CMP-ONTO** : représente l'ontologie des compétences
- **Règles CMP-IRO** : représentent les règles qui permettent de relier une compétence donnée à certains rôles pédagogiques.
- **Règles ILT-IRO** : représentent les règles qui permettent de lier une étape dans une théorie pédagogique à un rôle pédagogique. Par exemple, l'étape « Attirer l'attention de l'apprenant » peut être liée à la présentation d'un rôle pédagogique « introduction ».
- **Habiletés** : habiletés qui constituent les compétences à atteindre.

La mémoire organisationnelle occupe le centre de l'architecture. Elle est composée des différentes ontologies ainsi que de ressources d'apprentissage granulaires créées par les deux suites logicielles : ONTO-ENGINE et ONTO-AUTHOR.

La suite ONTO-ENGINE met en œuvre le processus d'extraction automatique de l'ontologie du domaine expliqué dans la partie acquisition des connaissances de la thèse. Elle est composée de l'outil TEXCOMON, détaillé en chapitre 3, ainsi que de l'environnement d'édition d'ontologies Protégé (Protégé, 2007), qui sert à mettre à jour et à accéder aux ontologies créées (notamment via l'API Protégé OWL API).

La suite ONTO-AUTHOR sert à annoter les objets d'apprentissage et de manière générale à éditer le contenu de la mémoire organisationnelle. ONTO-AUTHOR contient également des outils pour éditer les théories pédagogiques et les compétences et pour annoter les rôles pédagogiques.

Les communications entre les différentes suites logicielles et la mémoire organisationnelle sont effectuées au moyen de l'interface Protégé OWL API (Protégé OWL API, 2007) (voir l'annexe B pour plus de détails).

L'ensemble de ces outils d'édition et d'annotation est destiné à être utilisé par un concepteur de cours humain. Ce concepteur de cours, en tant qu'expert du domaine et en raison de la simplicité de l'outil TEXCOMON, peut également manipuler l'acquisition automatique d'une ontologie du domaine à partir de textes.

Les sections suivantes détaillent chacune des ontologies de la mémoire organisationnelle ainsi que les outils permettant de créer des instances dans ces ontologies.

6.2.2 L'ontologie du domaine

L'ontologie du domaine représente la pierre angulaire de notre structure ontologique. Une grande partie des ontologies définies par la suite réfèrent à cette ontologie. L'ontologie du domaine sert à indexer les ressources d'apprentissage selon le domaine de connaissance. Contrairement à la majorité des approches qui utilisent des ontologies du domaine existantes ou qui les construisent manuellement et tentent de faire coïncider les concepts de l'ontologie avec les objets d'apprentissage, notre approche, présentée en détail dans le chapitre 3, est radicalement différente : l'ontologie du domaine émerge « automatiquement » des objets d'apprentissage. Cela permet de conserver une certaine cohérence sémantique avec le contenu et évite la difficulté de l'appariement de deux modèles et visions différents. Cela ne répond pas toutefois à tous les problèmes inhérents à l'ingénierie ontologique. Par exemple, si deux synonymes sont utilisés pour désigner le même concept, le module TEXCOMON n'est pas à même, dans sa version courante, de le détecter. Cela nécessite une intervention humaine, de même que pour les connaissances provenant du sens commun ou des idiomes, etc.

Les ontologies présentées par la suite font partie intégrante du processus de composition automatique de nouvelles ressources d'apprentissage. Ce processus nécessite

un déclencheur pour la composition de ressources. Ce déclencheur est représenté par un besoin en compétence.

6.2.3 L'ontologie des compétences

De plus en plus de projets en éducation se basent sur une approche guidée par les compétences (Paquette, 2007) (Ng, Hatala, & Gasevic, 2006) (Voorhees, 2001) (Tuso & Longmire, 2000), et cela est spécialement requis et désirable dans les formations en entreprise. Sans en faire une condition sine qua none, nous pensons qu'une approche d'apprentissage basée sur une définition des compétences permet d'assembler des objets d'apprentissage de manière plus ciblée. Dans notre cas, une compétence s'applique à un ensemble d'habiletés définies sur les concepts du domaine, et représente un objectif d'apprentissage. Nous utilisons la taxonomie de Bloom (Bloom, 1956) pour exprimer les habiletés (Figure 35). Cette taxonomie a prouvé son utilité dans le domaine de l'apprentissage et de l'éducation (Nkambou, Frasson, & Gauthier, 2003).

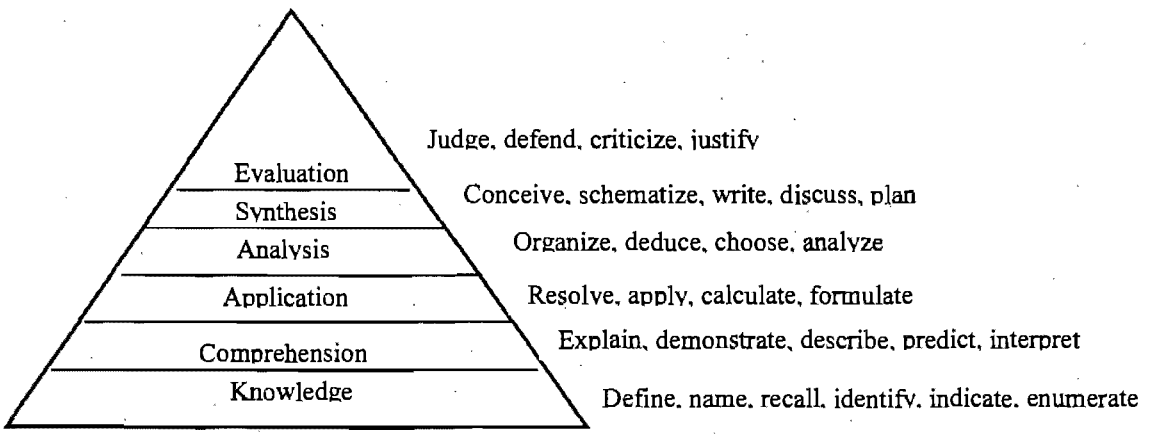


Figure 35. Taxonomie de BLOOM

La taxonomie de Bloom a été choisie car elle est suffisamment générique pour s'appliquer à des milieux académiques et professionnels. Une autre taxonomie pourrait toutefois être utilisée sans altérer le modèle proposé. En effet, la seule condition est que la taxonomie utilisée soit indépendante de la connaissance du domaine. Cela permet une plus

grande facilité de mise à jour, une plus grande flexibilité et un passage plus facile d'un domaine à l'autre. Par ailleurs, plus elle utilise de niveaux, plus il est possible de spécifier les besoins en apprentissage de manière granulaire.

Les compétences sont créées à l'aide d'un éditeur de compétences que nous avons développé et sont stockées dans l'ontologie des compétences. Les habiletés visées par la compétence sont reliées à l'ontologie du domaine générée lors de la phase d'acquisition des connaissances (Chapitre 3).

Un exemple de compétence touchant au premier niveau de la taxonomie de Bloom serait : ***définir*** le concept « *asset* » où ***définir*** est une habileté et ***asset*** est un concept du domaine.

Une approche basée sur les compétences nécessite un remodelage des ressources d'apprentissage afin que les portions de ces ressources les plus à mêmes de répondre au besoin de formation soient exploitées. Par exemple, pour définir le concept « *asset* », comme dans l'exemple précédent, il est nécessaire de disposer d'une ressource de type ***Définition*** concernant le concept en question, ou d'isoler une portion de texte relative à ce concept. Il est également nécessaire de disposer du concept « *asset* » lui-même en tant que concept du domaine. Ceci est effectué par la modélisation de la structure et de la sémantique des documents et par leur indexation domaine et selon des rôles pédagogiques.

6.2.4 L'ontologie de structure

L'ontologie de structure sert à indexer les composants structurels des documents sources tels que les paragraphes, les phrases, les tables, les figures, etc. Ces composants sont en partie détectés automatiquement via des annotateurs (pour les paragraphes et les phrases) basés sur l'architecture UIMA (UIMA Java Framework, 2007) (cf. section 3.3). Les autres composants doivent être annotés de manière manuelle dans la version actuelle du logiciel. Les instances de cette ontologie sont donc des portions plus ou moins granulaires de documents. L'outil utilisé pour l'extraction automatique de la structure et des

connaissances du document est montré dans la figure 36. Cet outil a déjà été présenté dans le chapitre 3. Pour rappel, et ainsi qu'indiqué dans la figure, le document se trouve au centre et la structure extraite (sur la gauche) organise le document en paragraphes et en phrases. L'outil d'extraction de structure et de connaissances fait partie intégrante de TEXCOMON. Il implante toutes les fonctionnalités détaillées dans le chapitre 3, à savoir la détection de mots-clés et de phrases-clés, le lancement de l'analyse grammaticale via l'analyseur de Stanford, et enfin la transformation des cartes de concepts grammaticales en représentations sémantiques.

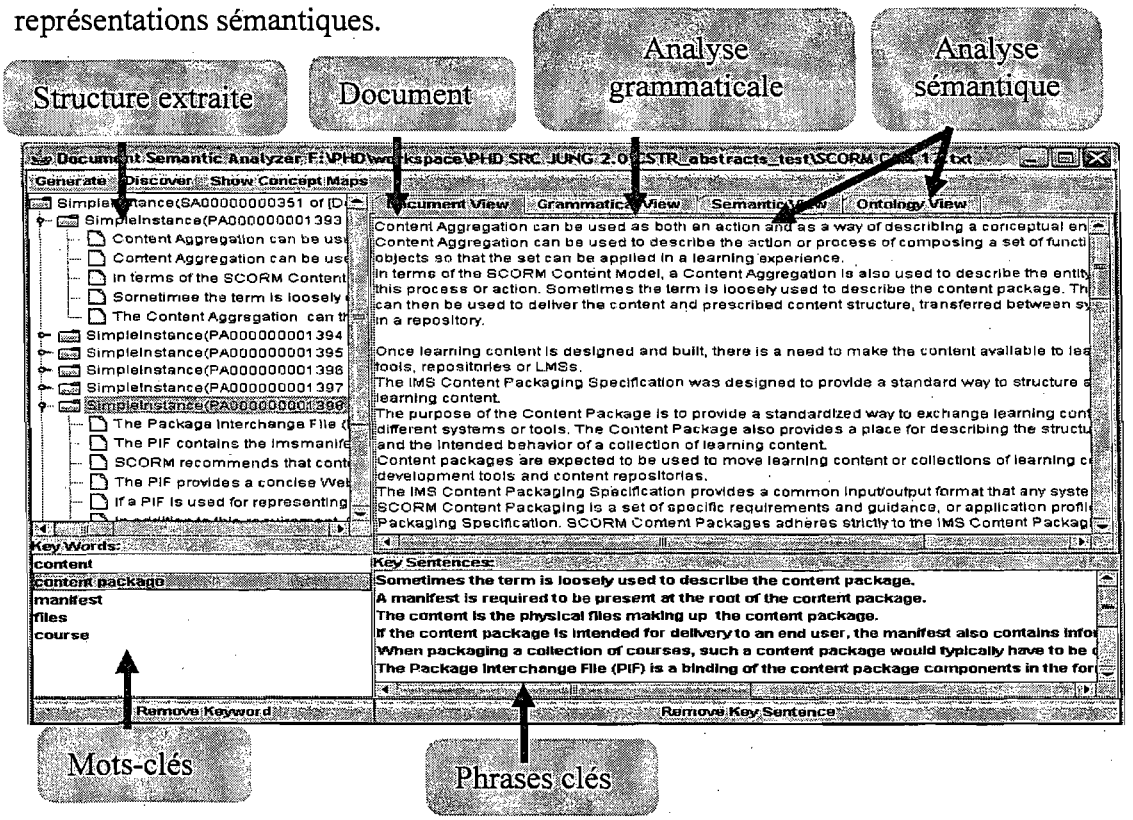


Figure 36. Outil d'extraction de structure et de connaissances

Les phrases et les paragraphes stockés dans l'ontologie de structure ne disposent pas, à ce stade, d'une étiquette permettant de connaître leur rôle éventuel dans un processus de composition automatique. C'est à ce niveau qu'intervient l'ontologie des rôles pédagogiques.

6.2.5 L'ontologie des rôles pédagogiques

Nous définissons un rôle pédagogique comme une fonction pédagogique remplie par une ressource d'apprentissage ou d'une portion de cette ressource (ressource textuelle). L'ontologie des rôles pédagogiques permet de définir un ensemble de rôles pédagogiques que l'on retrouve souvent dans les objets d'apprentissage comme les définitions, les explications, les exemples, les introductions, les descriptions, les conclusions, etc. Ces fonctions, par ailleurs largement utilisées en éducation, sont utiles pour les concepteurs de cours mais aussi pour les apprenants. Elles permettent des recherches plus ciblées et favorisent donc une plus grande réutilisation (Verbert, Jovanovic, Duval, & Gaševic, 2006). Elles évitent le syndrome du copier-coller, qui conduit à la duplication de la même connaissance (Verbert & Duval, 2004) et donc à des difficultés de mise à jour. En effet, plutôt que de coller la même définition du concept « *asset* », par exemple, dans plusieurs objets d'apprentissage, il suffit de référer à l'objet définition (créé manuellement au moyen d'une édition ou d'un mécanisme d'annotation) dans chacun des objets d'apprentissage qui l'utilisent. Les concepteurs de cours bénéficient de cette ontologie pour construire des cours plus rapidement et plus facilement. Par ailleurs, dans le cadre d'un mécanisme de composition basé sur les compétences, cette approche permet une indexation plus fine des ressources d'apprentissage et donc un moyen plus flexible pour combler les écarts de compétences d'un apprenant.

Cette ontologie est donc cruciale dans un processus d'agrégation (manuel ou automatique) de ressources d'apprentissage à même de combler un besoin précis (via des services Web, des applications *stand-alone*). Avec l'augmentation de l'utilisation des services Web dans le cadre du Web sémantique, une telle ontologie peut également jouer un rôle dans la composition adéquate de services par exemple (Ullrich, 2005).

L'annotation des rôles pédagogiques se fait dans l'outil « *Knowledge Annotator* » (Figure 37) en sélectionnant la portion de texte correspondant au rôle pédagogique désiré (ici, une définition) et en la glissant sur le rôle en question dans l'onglet « *Instructional*

View ». Ensuite, ce rôle pédagogique doit être rattaché à un concept du domaine en utilisant la fenêtre des propriétés en bas du texte. La portion de texte sélectionnée peut correspondre à un paragraphe ou une phrase préalablement stockés dans l'ontologie de structure, mais elle peut également être composée d'un ensemble de phrases, d'un sous-ensemble de paragraphe, ou chevaucher plusieurs paragraphes.

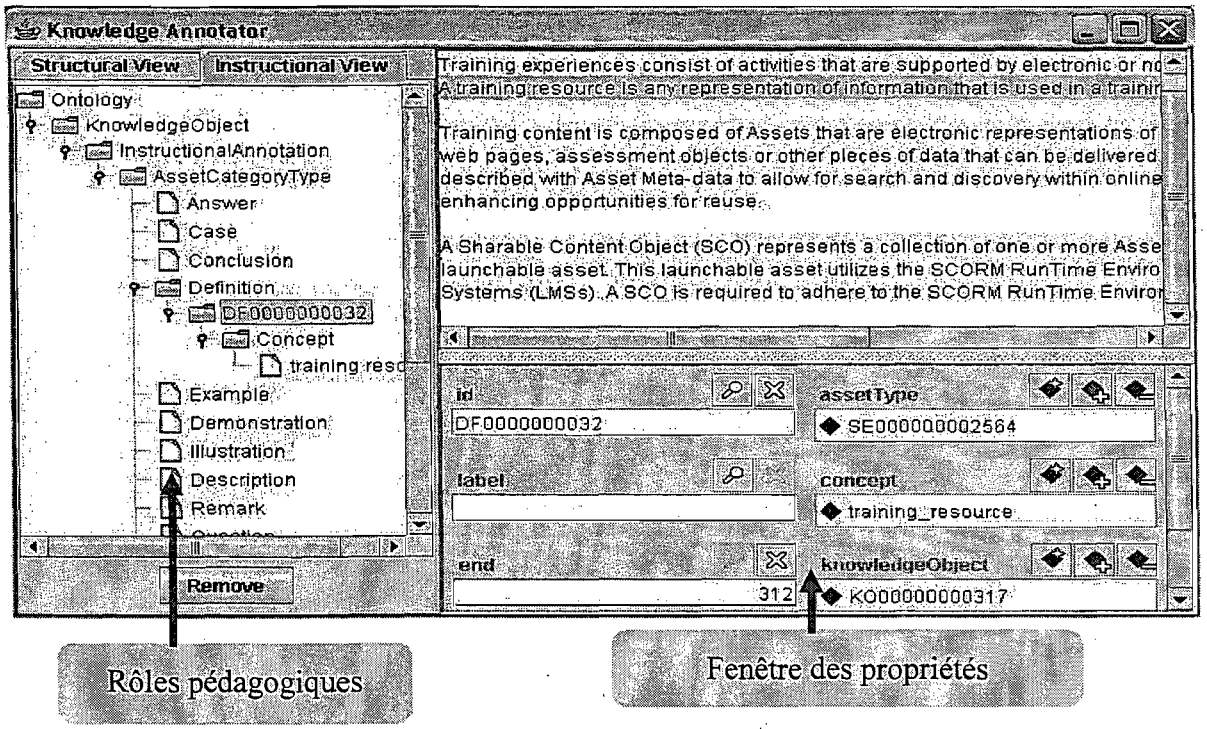


Figure 37. L'outil d'annotation pédagogique

A terme, notre objectif est d'automatiser l'annotation des rôles pédagogiques. En effet, les mêmes critiques se rapportant à la difficulté de mise à jour et à la lourdeur d'un processus manuel s'appliquent à cette annotation.

Les portions de texte sont ensuite stockées en tant qu'instances de rôles pédagogiques. Il est alors possible de rechercher un rôle pédagogique concernant un concept du domaine donné à l'aide de l'interface présentée ci-dessous. Le système retourne alors le rôle surligné dans son texte original (Figure 38).

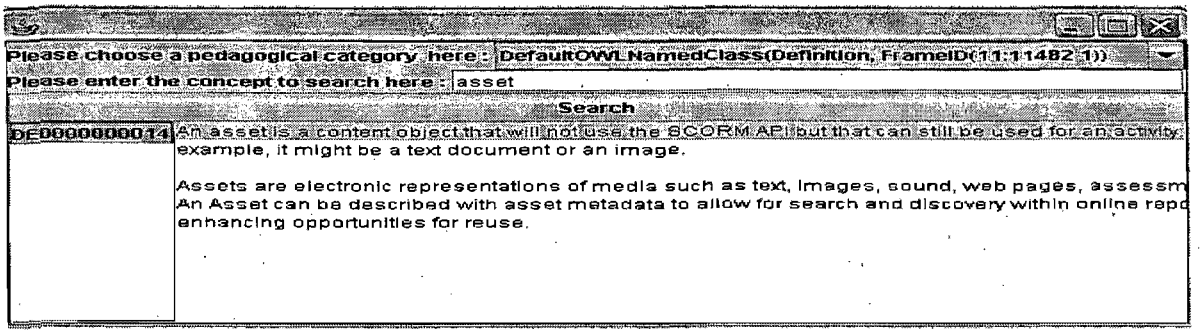


Figure 38. Interface de recherche de rôles pédagogiques

A ce stade, nous disposons d'un ensemble de composants plus ou moins granulaires devant servir dans la composition de nouvelles ressources. Nous ne disposons toutefois pas de la logique d'assemblage, celle qui va contenir la pédagogie du déroulement de l'apprentissage. L'ontologie des théories pédagogiques permet de formaliser cette logique.

6.2.6 L'ontologie des théories pédagogiques

L'un des problèmes des objets d'apprentissage classiques est qu'ils ne disposent pas d'une théorie pédagogique explicite qui guide leur conception. Cette théorie existe bien sûr, mais le concepteur de l'objet d'apprentissage l'indique seulement implicitement par les choix pédagogiques qu'il effectue (Ullrich, 2004). Or cela peut constituer un obstacle à des programmes automatiques pour la recherche d'objets d'apprentissage pertinents, pour l'agrégation automatique d'objets d'apprentissage ou encore pour l'explicitation du contenu d'un objet d'apprentissage.

Afin de pallier ces inconvénients, la génération de ressources d'apprentissage doit être guidée par des théories d'apprentissage. L'article de (Bourdeau, Mizoguchi, Psyché, & Nkambou, 2004) évoque d'ailleurs la nécessité d'incorporer des structures conceptuelles communes pour modéliser les théories d'apprentissage. L'article indique également que ces structures doivent être encodées de manière déclarative pour doter les systèmes de formation d'une expertise pédagogique (*theory-awareness)*.

Dans notre thèse, les outils d'exploitation de l'architecture servent essentiellement à générer des ressources d'apprentissage selon un scénario pédagogique tiré des théories de l'éducation. Ces théories permettent de garantir une bonne approche pédagogique pour la mise en œuvre d'un plan de cours. L'ontologie des théories pédagogiques considère une théorie comme un ensemble d'étapes pédagogiques. Chacune de ces étapes est reliée à un ensemble de règles utilisant le formalisme SWRL (*Semantic Web Rule Language*). Ces règles représentent la partie déclarative de la théorie. Elles permettent d'incorporer les différents composants d'une ontologie (les classes, les instances, et les propriétés) dans les prémisses ou les corps des règles.

Pour déterminer le contenu d'une formation, il est nécessaire de rechercher les habiletés à maitriser. Chaque type d'habileté *(Knowledge, comprehension, ...)* renvoie à des rôles pédagogiques différents permettant de la maitriser. Afin de relier une certaine étape pédagogique à des ressources appropriées, nos règles sont couplées aux rôles pédagogiques (nous les appelons les règles de Bloom). Par exemple, la règle suivante permet d'indiquer ce qui est fait pour définir (selon l'acception de Bloom) un concept :

AbilityAcquisition(define) -> query : select(Definition)

Où « *define* » est une instance du niveau « *Knowledge* » de Bloom. Cette règle spécifie qu'il faut retrouver un rôle pédagogique de type « *Definition* » pour acquérir cette habileté.

Les règles peuvent également faire appel à des méthodes prédéfinies. En effet, parfois, une étape spécifiée par une théorie ne consiste pas seulement à retrouver une connaissance déclarative comme un rôle pédagogique, mais peut nécessiter des opérations sur des données. Par exemple, pour l'étape «*Fournir un résultat à l'apprenant* », il est nécessaire de calculer d'abord le score de ce dernier. C'est pourquoi nous avons mis en place dans l'ontologie quelques méthodes génériques qui représentent des sortes **d'actions primitives**. Ces méthodes comprennent :

- une méthode pour rechercher les pré-requis d'un concept,

- une méthode pour calculer le score de l'apprenant dans un exercice,
- une méthode pour générer ou retrouver des exercices,
- une méthode pour retrouver les objectifs d'apprentissage,
- et enfin, une méthode pour déterminer le contenu de la formation proprement dite : cette méthode fait appel, en fonction de l'habileté à acquérir, à certaines des règles de Bloom.

De manière générale, une étape de la théorie déclenche soit une recherche de rôle pédagogique, soit une action primitive qui est exécutée par l'environnement de formation. Les règles SWRL sont créées dans un éditeur SWRL fourni comme plugin dans l'éditeur d'ontologie Protégé (Protégé, 2007) (Figure 39).

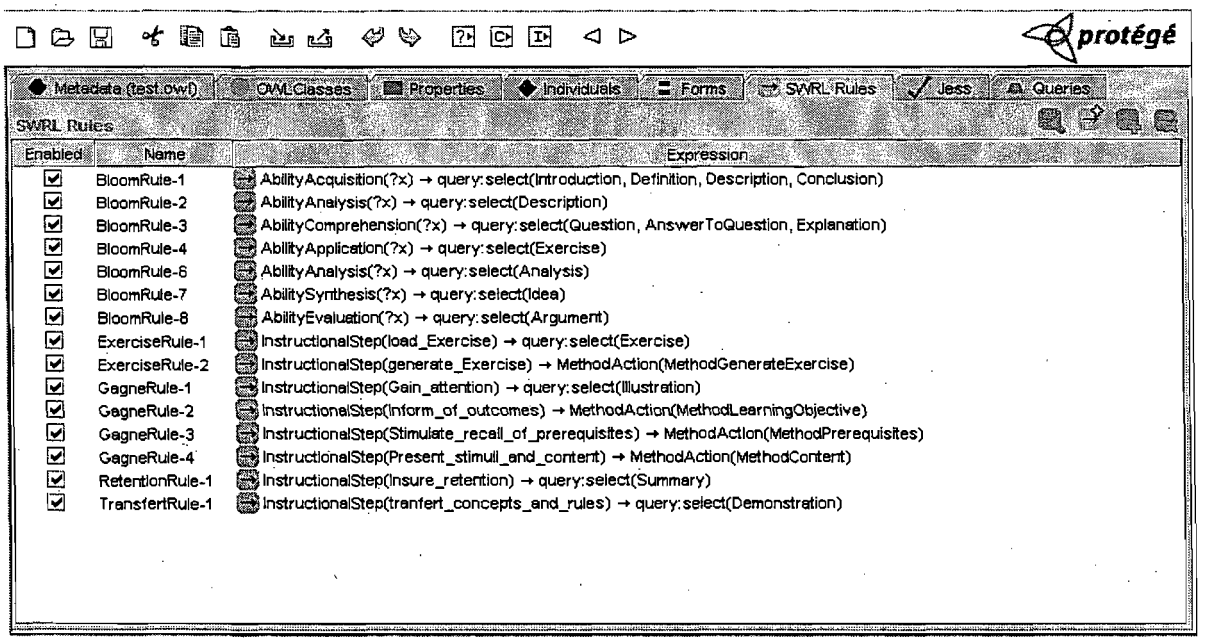


Figure 39. Les règles de la théorie pédagogique issue de Gagné dans un éditeur SWRL

Pour effectuer une preuve de concepts, nous avons choisi de modéliser la théorie des conditions d'apprentissage de Gagné (Gagné, Briggs, & Wagner, 1992) dans notre ontologie des théories (Figure 39). Gagné indique que des événements externes (il en a

dénombré 9) peuvent influencer le processus d'apprentissage. Parmi ces événements, on peut citer par exemple :

- **attirer l'attention de l'apprenant** : cet événement consiste à attirer et à conserver l'attention de l'apprenant. Diverses stratégies sont proposées comme de présenter un message précis, une question, une image ou une vidéo sur le concept à maitriser. Dans ce cadre, une règle possible est :

  InstructionalStep(Gain_attention) -> query : select(Illustration). Cette règle spécifie que pour attirer l'attention de l'apprenant, on peut lui présenter un rôle pédagogique de type Illustration.

- **stimuler le rappel des connaissances pré-requises** : le rappel de pré-requis permet à l'apprenant d'établir des liens entre des notions déjà connues et les nouvelles. Ce rappel s'appuie sur l'ontologie du domaine et les cartes de concepts pour connaître le contexte du concept à enseigner (les relations et concepts voisins constituant sa carte conceptuelle). Il s'appuie également sur la définition de la compétence à acquérir qui spécifie toutes les habiletés à connaître pour pouvoir entamer l'apprentissage du concept courant.

- **permettre la rétention, la généralisation des concepts et le transfert des connaissances** : cela permet de vérifier que la notion est vraiment bien comprise en la transférant à des situations nouvelles et en récapitulant les connaissances acquises. Par exemple, il est utile, à cette étape, de fournir des cartes de concepts, un résumé, etc. Un exemple de règle SWRL adéquate serait :

  InstructionalStep(Insure_retention) -> query : select(Summary).

Évidemment, la théorie de Gagné n'est qu'un exemple d'application et il serait intéressant d'ajouter d'autres théories à l'ontologie et de les associer à d'autres règles.

Les différentes règles sont exécutées dans un moteur de règles compatible avec l'interface « *SWRL Rule Engine Bridge* ». Dans notre cas, cela a été fait avec Jess

(Friedman-Hill, 2003). Cette interface permet à un modèle OWL couplé à des règles SWRL de communiquer avec un moteur de règles. Elle permet à ce dernier d'effectuer les inférences et le traitement requis par les règles. Dans notre cas, ainsi que nous l'avons dit, l'exécution des règles indique soit le type de ressources pédagogiques que le système doit aller chercher dans la mémoire organisationnelle, soit les méthodes à exécuter à une étape donnée de la théorie pédagogique.

La spécification des théories de manière indépendante du contenu de formation permet une plus grande flexibilité. Par ailleurs, notre approche est suffisamment ouverte pour permettre d'intégrer de nouvelles théories ou d'utiliser une autre ontologie des théories pédagogiques. Il « suffit » alors d'adapter les règles à cette nouvelle ontologie. Notons en effet que certains travaux, notamment le projet OMNIBUS (Mizoguchi, Hayashi, & Bourdeau, 2007) ont produit une ontologie des théories pédagogiques beaucoup plus riche qu'il serait intéressant d'intégrer à notre travail.

6.3 Architecture du module d'exploitation des connaissances

L'exploitation des connaissances dans notre projet permet d'utiliser le contenu de la mémoire organisationnelle pour assurer des services tutoriels. La mémoire n'est plus là pour stocker des objets d'apprentissage complets, mais plutôt pour organiser ces objets en fragments réutilisables. Cette mémoire est notamment utilisée pour la composition automatique de ressources d'apprentissage que nous appelons *objets de connaissances et d'apprentissage* (*Learning Knowledge Objects ou LKO*).

La figure 40 illustre l'architecture du module d'exploitation des connaissances, composé essentiellement de trois services : un service de composition, un service de standardisation et un service de déploiement. Ces services puisent leurs ressources dans la mémoire organisationnelle.

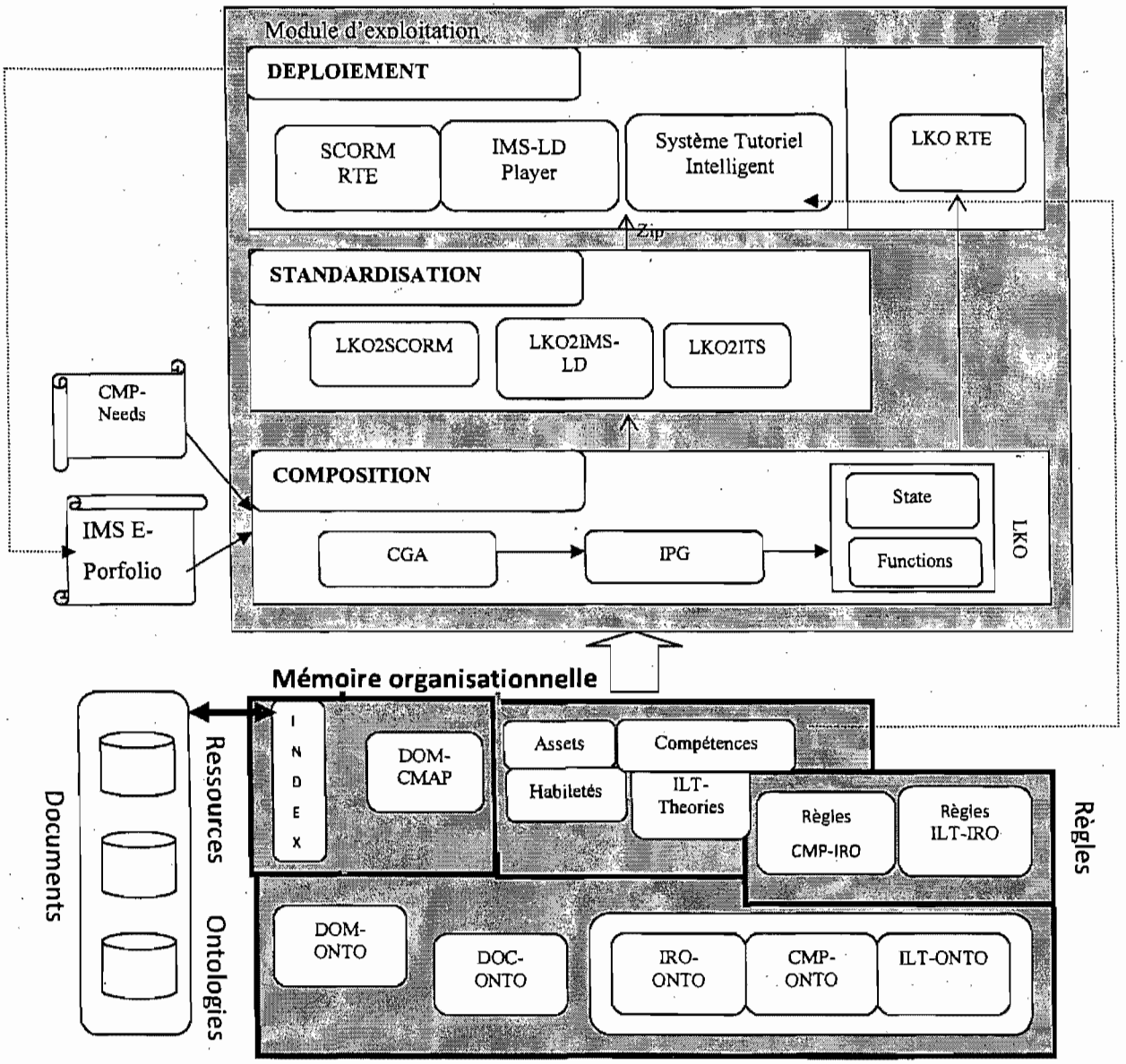


Figure 40. Architecture du module d'exploitation des connaissances.

Les sections suivantes expliquent plus en détail les objets de connaissances et d'apprentissage ainsi que les services du module d'exploitation des connaissances.

6.3.1 Les Objets de connaissance et d'apprentissage (LKO)

Ainsi que nous l'avons expliqué précédemment, la vision actuelle du e-Learning repose entièrement sur la notion d'objet d'apprentissage. Cette vision est la plus proche de la formation dite traditionnelle, basée sur des documentations textuelles. Toutefois, nous avons déjà évoqué pourquoi cette vision ne nous semblait pas convenir aux besoins actuels du e-Learning : stockage d'objets complets, nécessité de métadonnées bien souvent incomplètes, description du contexte de l'objet d'apprentissage (langue, auteur, etc.) plutôt que son contenu même, etc.

Le e-Learning s'est fait le chantre de la réutilisation des objets d'apprentissage mais dans la réalité, cette vision tarde à se concrétiser et aucun standard n'est réellement parvenu à mettre en place un modèle encourageant réellement cette réutilisation. Cet état de fait est dû, selon nous, à la définition même des objets d'apprentissage considérés comme un tout opaque accessible au monde extérieur uniquement via des métadonnées (également externes). L'avènement du Web sémantique a toutefois favorisé des approches permettant la réutilisation d'objets d'apprentissage en totalité ou en partie. De fait, beaucoup d'efforts ont été entrepris pour réutiliser les objets d'apprentissage (Li & Huang, 2006) (Verbert, Jovanovic, Duval, & Gašević, 2006) (Jovanovic, Gasevic, & Devedzic, 2006b). Ces efforts portent généralement un nom : ils désignent l'opération d'agrégation, de composition, de génération et de réutilisation des objets d'apprentissage. D'autres problèmes émergent de ce désir de réutilisation, auxquelles nous avons déjà tenté d'apporter des éléments de réponse :

- Comment cette réutilisation peut-elle être implantée ? Est-il intéressant de réutiliser un objet d'apprentissage en totalité ?
- Existe-t-il des portions qui puissent effectivement servir de briques de connaissances à assembler par des agents humains ou logiciels ?

- Comment guider cette agrégation ? Par des théories pédagogiques ? Comment ces théories peuvent-elle être implantées dans un objet d'apprentissage tout en laissant son contenu indépendant de la théorie ?
- Les ressources agrégées sont-elles ensuite exploitables par des environnements standards tels que SCORM et IMS-LD ?

La création d'objets de connaissances et d'apprentissage (*Learning Knowledge Objects* ou LKO) vise à obtenir des ressources qui répondent de façon satisfaisante aux critères invoqués ci-dessus et qui permettent d'éviter les inconvénients des objets d'apprentissage actuels. Diverses propriétés caractérisent les LKOs :

- Les LKOs doivent être *dynamiques*. Ils ne doivent pas exister en tant qu'entités statiques stockées dans des entrepôts. Les LKOs doivent être générés pour répondre à un besoin précis à un moment précis.
- Les LKOs doivent *posséder un modèle du domaine* qu'ils sont sensés transmettre. Ils doivent pouvoir ainsi offrir non seulement un apprentissage didactique mais également un apprentissage constructiviste. Par apprentissage didactique, nous entendons un apprentissage guidé par un plan de formation. Par apprentissage constructiviste, nous entendons un apprentissage permettant à l'apprenant d'explorer certaines notions au travers de cartes conceptuelles.
- Les LKOs doivent connaître *les théories pédagogiques* qui ont permis de les générer (*theory-awareness*).
- Les LKOs doivent être *actifs*. Ils ne doivent pas être pensés comme des structures de données statiques mais comme des objets dotés de services tutoriels.
- Enfin, les LKOs doivent pouvoir *s'adapter à un apprenant* particulier.

Ces différentes caractéristiques vont être respectées dans le processus de génération automatique des LKOs.

6.3.2 Génération automatique d'objets de connaissances et d'apprentissage (LKOs)

Trois services essentiels composent le mécanisme de génération des LKOs : le service de composition, le service de déploiement et le service de standardisation.

6.3.2.1 Le service de composition

Le service de composition est en charge de l'opération d'agrégation des LKOs en fonction d'un besoin spécifique en compétences (*CMP-Needs*) et selon un modèle d'apprenant spécifique (sous forme d'un IMS ePortfolio (IMS ePortfolio Specification, 2007)). La spécification IMS ePortfolio permet de définir et de communiquer, sous forme de portfolio numérique, les différentes données d'un individu relatives à son parcours éducatif et professionnel.

Les objectifs d'apprentissage ou compétences sont exprimés dans un fichier OWL qui décrit les habiletés à acquérir et les concepts du domaine concernés. Le modèle de l'apprenant est exprimé en tant qu'IMS ePortfolio et converti en OWL afin de pouvoir être utilisé par le service de composition. Ce modèle sert à stocker les compétences acquises ainsi que les caractéristiques de l'apprenant.

Une fois que les objectifs d'apprentissage sont spécifiés, la définition de la compétence est comparée avec le modèle de l'apprenant en utilisant un *analyseur d'écart de compétences*. Pour chaque habileté de la compétence, le modèle de l'apprenant est parcouru pour savoir si cette habileté est déjà maitrisée ou pas, et si des pré-requis sont nécessaires. Dans l'affirmative, ces derniers sont rajoutés à la définition de la compétence et permettent de produire une compétence ajustée aux besoins de l'apprenant.

Cette compétence ajustée sert de point de départ à un *générateur de plan d'apprentissage (IPG)* qui est en charge de composer le LKO en fonction d'une théorie pédagogique. L'IPG (Figure 41) exploite l'ontologie des théories pédagogiques pour retrouver les événements d'instruction qui vont guider la composition et aboutir à un plan

de formation. Cela permet d'incorporer les théories de manière flexible et indépendante dans les LKOs. Cela dote également le LKO d'une structure pédagogique explicite qui peut être comprise par des concepteurs humains et des agents logiciels. Chaque habileté de la compétence visée est transformée en activité. Chaque activité est composée de l'ensemble des étapes pédagogiques nécessaires à sa réalisation. Ces étapes proviennent d'une théorie pédagogique choisie au moment de l'agrégation. Par exemple, la figure 41 montre la génération d'un LKO en fonction de la théorie de Gagné.

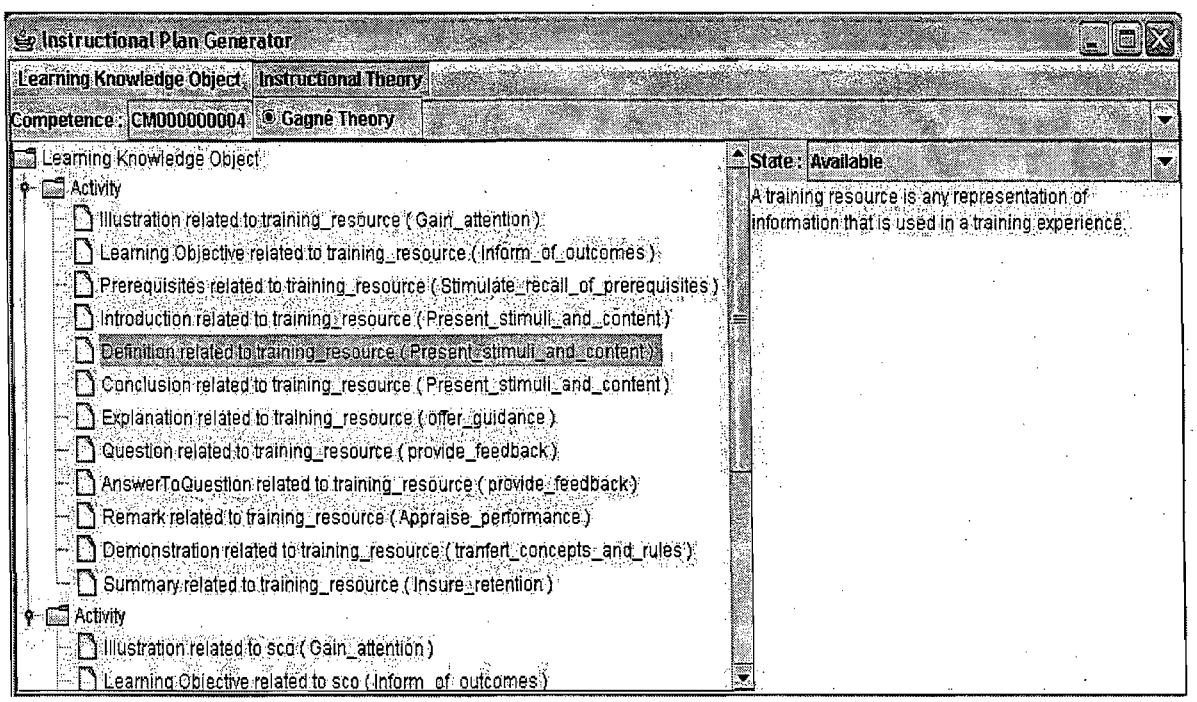


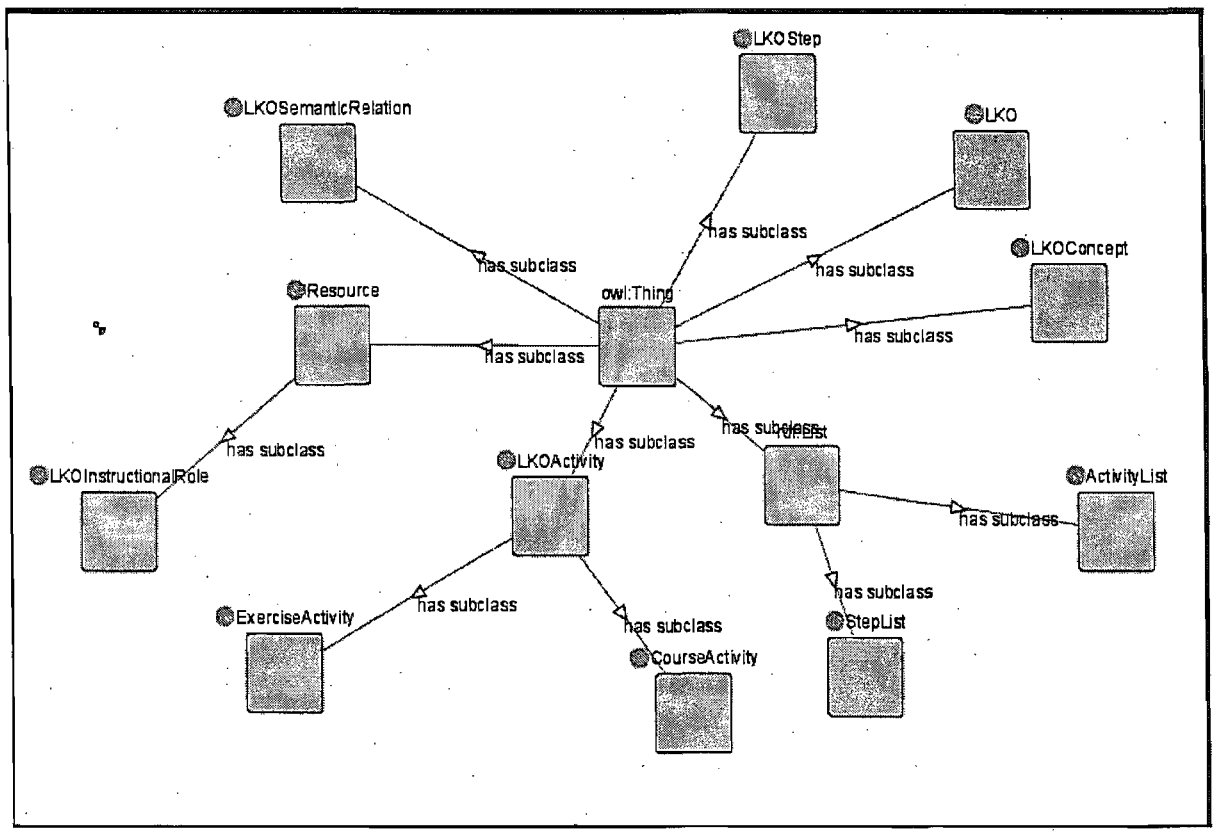
Figure 41. Le générateur de plans d'apprentissage

L'exécution de l'IPG produit un objet de connaissance et d'apprentissage (LKO). Ce dernier peut être vu comme un objet indépendant composé d'un état de données (*state*) et d'un ensemble de fonctions (*functions*) pour manipuler cet état.

Les données d'un LKO sont exprimées comme instances d'une ontologie OWL qui regroupe les différentes ressources nécessaires à l'exécution du LKO. Ces ressources

comprennent la compétence à maitriser, les habiletés, les concepts concernés, l'ontologie du domaine autour de ces concepts, les cartes de concepts et le modèle de l'apprenant.

La figure suivante (figure 42) illustre l'ontologie d'un LKO. La première image montre la hiérarchie des classes de l'ontologie alors que la seconde montre les relations conceptuelles entre ces classes.



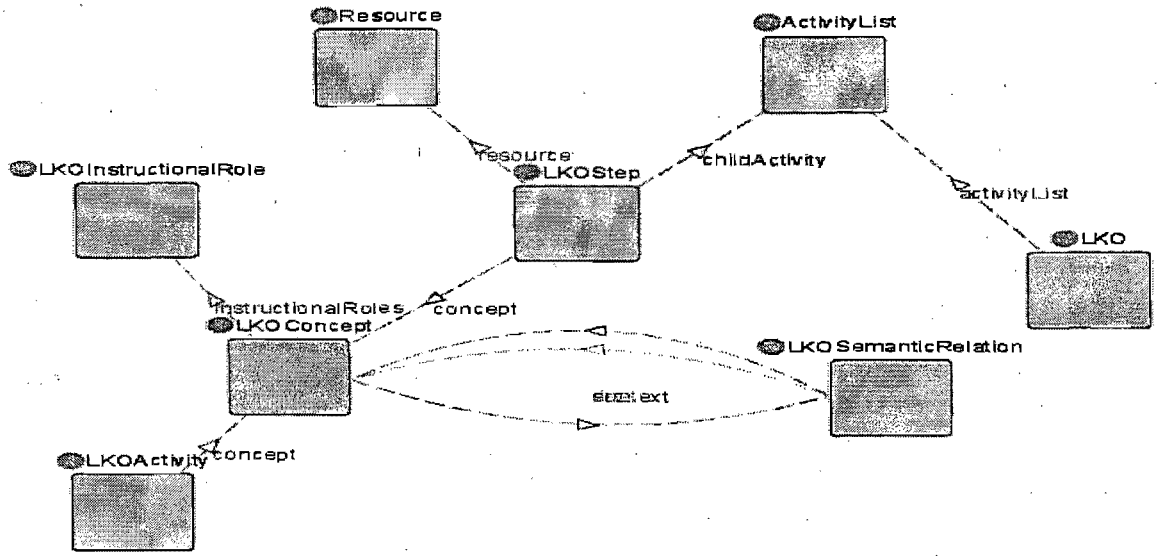


Figure 42. L'ontologie d'un LKO (données)

Les fonctions du LKO sont regroupées en une interface standard qui permet à un LKO d'agir comme un mini système tutoriel ou comme un environnement de formation auto-suffisant ainsi que nous le verrons dans l'étape de déploiement.

6.3.2.2 Le service de déploiement

Les LKOs doivent pouvoir être exécutés dans n'importe quel environnement de formation. Le projet « The Knowledge Puzzle » fournit un environnement d'exécution (*LKO Runtime Environment (LKO-RTE)*) aux usagers qui n'ont pas accès à d'autres environnements et qui ne souhaitent pas particulièrement utiliser des environnements e-Learning standards. Le LKO-RTE permet d'exécuter un LKO comme une ressource indépendante (*stand-alone*). L'interface du LKO-RTE permet à l'usager d'accéder à l'ensemble des fonctions implantées dans le LKO qui supportent les services tutoriels. Cette interface est composée des fonctions suivantes :

- contrôle du scénario pour guider la progression de l'apprenant à travers le contenu ;
- évaluation des actions et exercices effectués par l'apprenant ;

- mise à jour du portfolio de l'apprenant ;
- exploration de l'ontologie du domaine et des cartes de concepts ;
- génération d'exercices à partir de l'ontologie du domaine ;
- explication des différents concepts par leur contexte ;
- génération à la demande de LKOs reliés au contexte du concept ;

Le contrôle de scénarios est mis en œuvre en suivant les étapes de la théorie pédagogique utilisée pour générer le LKO. La figure 43 montre le déploiement d'un LKO dans l'environnement d'exécution LKO-RTE. Cet environnement est composé d'une vue sur le plan et contenu (*Course View*), une vue sur la carte de concepts associée (*Concept Map View*) et une vue sur l'exploration des cartes de concepts (*Concept Map Exploration*).

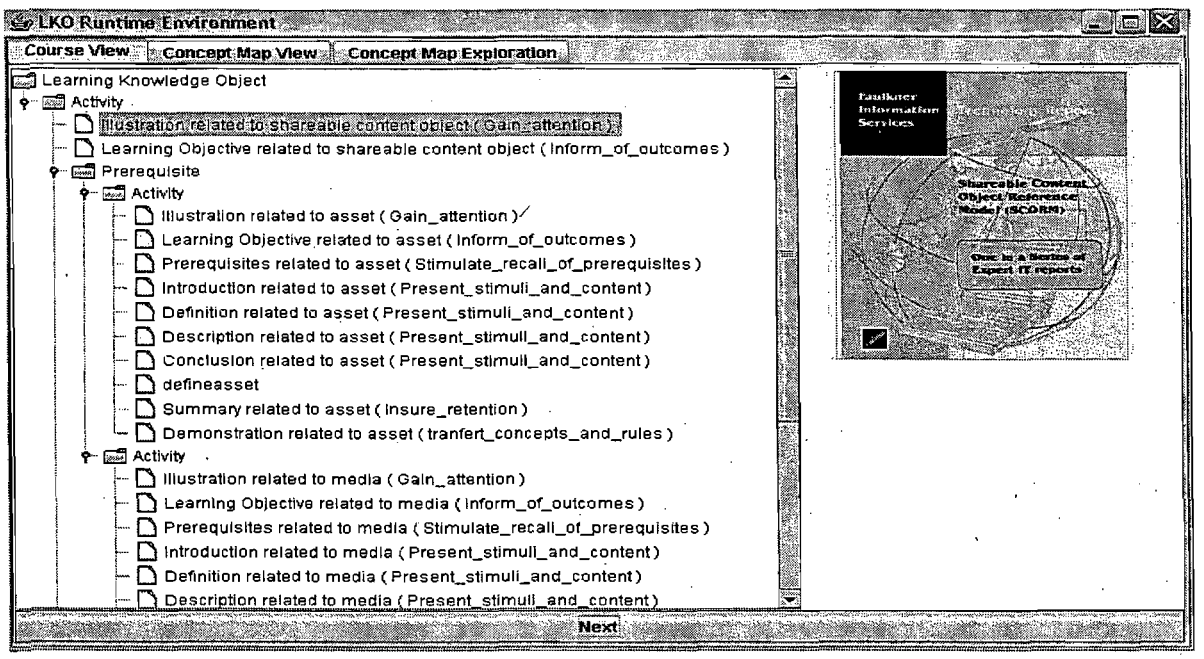


Figure 43. L'environnement d'exécution d'un LKO : déploiement de ressources didactiques

L'apprentissage didactique fourni par le LKO consiste à suivre le plan de cours généré et à résoudre les exercices présentés. En effet, un LKO peut également déployer des exercices afin de tester l'apprenant. Un exercice est constitué d'une ou plusieurs relations

correctes à choisir par un ensemble de relations erronées. Ces exercices peuvent être édités ou automatiquement générés en se basant sur l'ontologie du domaine et sur les liens entre concepts. Ces deux possibilités (édition et/ou génération) sont là encore spécifiées dans les règles SWRL reliées à la théorie pédagogique utilisée.

La génération d'un exercice utilise le contexte du concept concerné par l'habileté (sa carte de concepts) pour sélectionner une ou plusieurs relations correctes du domaine. Ensuite, une liste de relations erronées est établie en échangeant les labels des relations ou des concepts source et destination. L'apprenant doit ensuite pouvoir détecter la ou les relations correctes.

Une des faiblesses des objets d'apprentissage textuels actuels est qu'ils ne favorisent qu'un seul type d'apprentissage : un apprentissage didactique passif. L'apprenant ne peut que lire le contenu et résoudre des exercices. Le projet « The Knowledge Puzzle » permet en plus un apprentissage constructiviste (Novak & Cañas, 2006) en exploitant l'ontologie du domaine et les cartes de concepts. Chaque concept X des habiletés visées par la compétence à acquérir est extrait de l'ontologie accompagné de son contexte. Un contexte, dans ce cas, est représenté par la carte des concepts reliée à X. L'onglet «*Concept Map View*» offre une vue globale sur les concepts visés par la compétence et permet de montrer leurs contextes et leurs liens. De ce fait, elle permet de renforcer l'apprentissage actif (*meaningful learning*) (Novak & Cañas, 2006). Par exemple, la figure 43 montre le déploiement d'un LKO relié à la compétence « Définir le concept de SCO (*Shareable Content Object*) ». Supposons que cette compétence ait deux pré-requis : « définir le concept Asset » et « définir le concept Média ». Dans la vue « Carte de Concepts » (Figure 44), les différents concepts « *asset, media* et *shareable content object* » sont entourés de leurs contextes et reliés par des relations directes : *assets are electronic representation of media, asset is content object* et *Shareable Content Object is special kind of content object.*

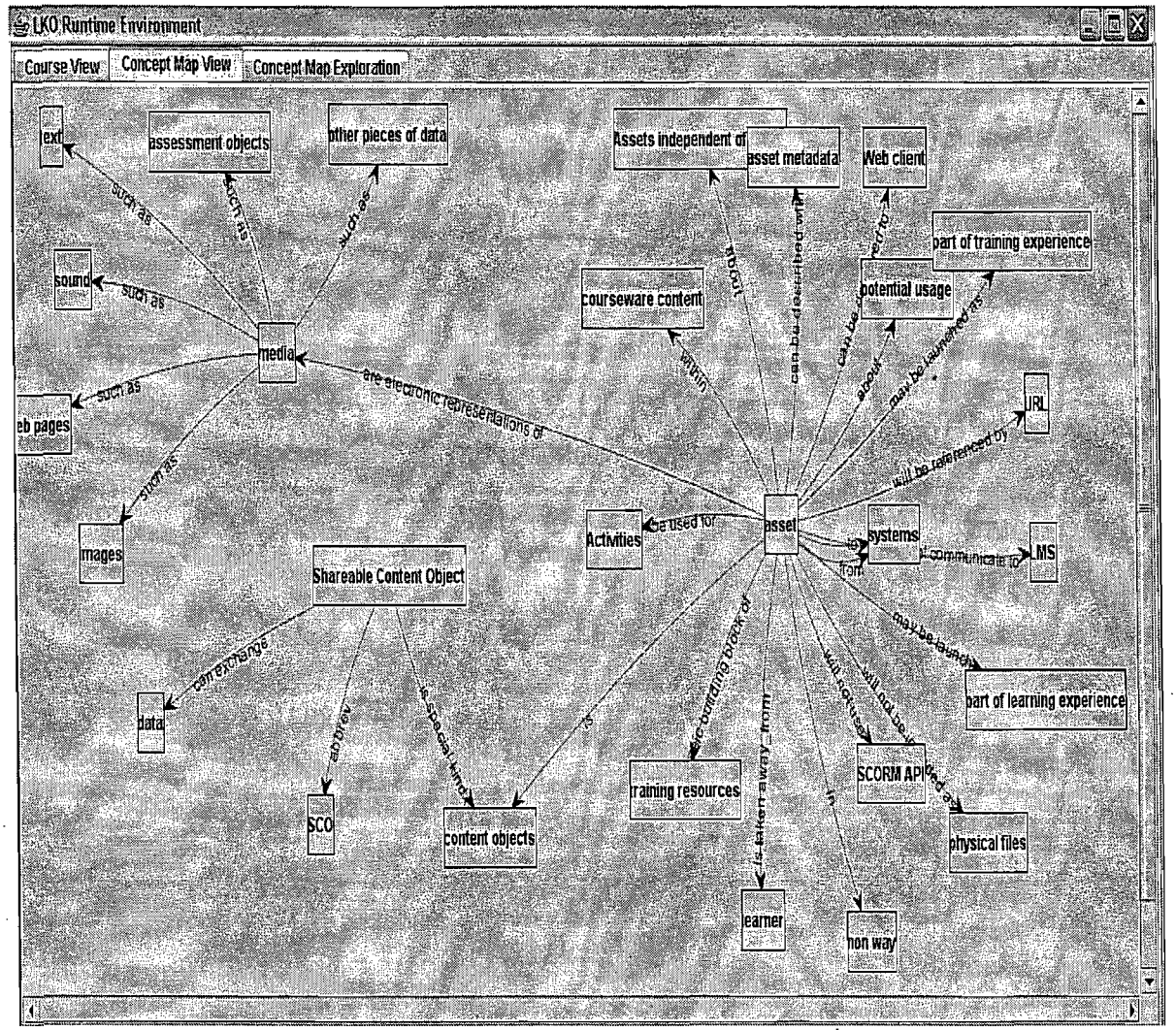


Figure 44. La vue « Carte de Concepts »

Enfin, l'exploration des cartes de concepts permet de compléter l'apprentissage en approfondissant la compréhension qu'a l'apprenant du concept et de son contexte. En effet, l'apprenant peut non seulement explorer les relations directes, mais également les relations indirectes et demander la génération du contexte d'un concept appartenant au contexte précédent et ainsi de suite. Par exemple, dans la figure 45, un menu contextuel offre la possibilité de naviguer dans le contexte du concept sélectionné (ici, le concept de LMS).

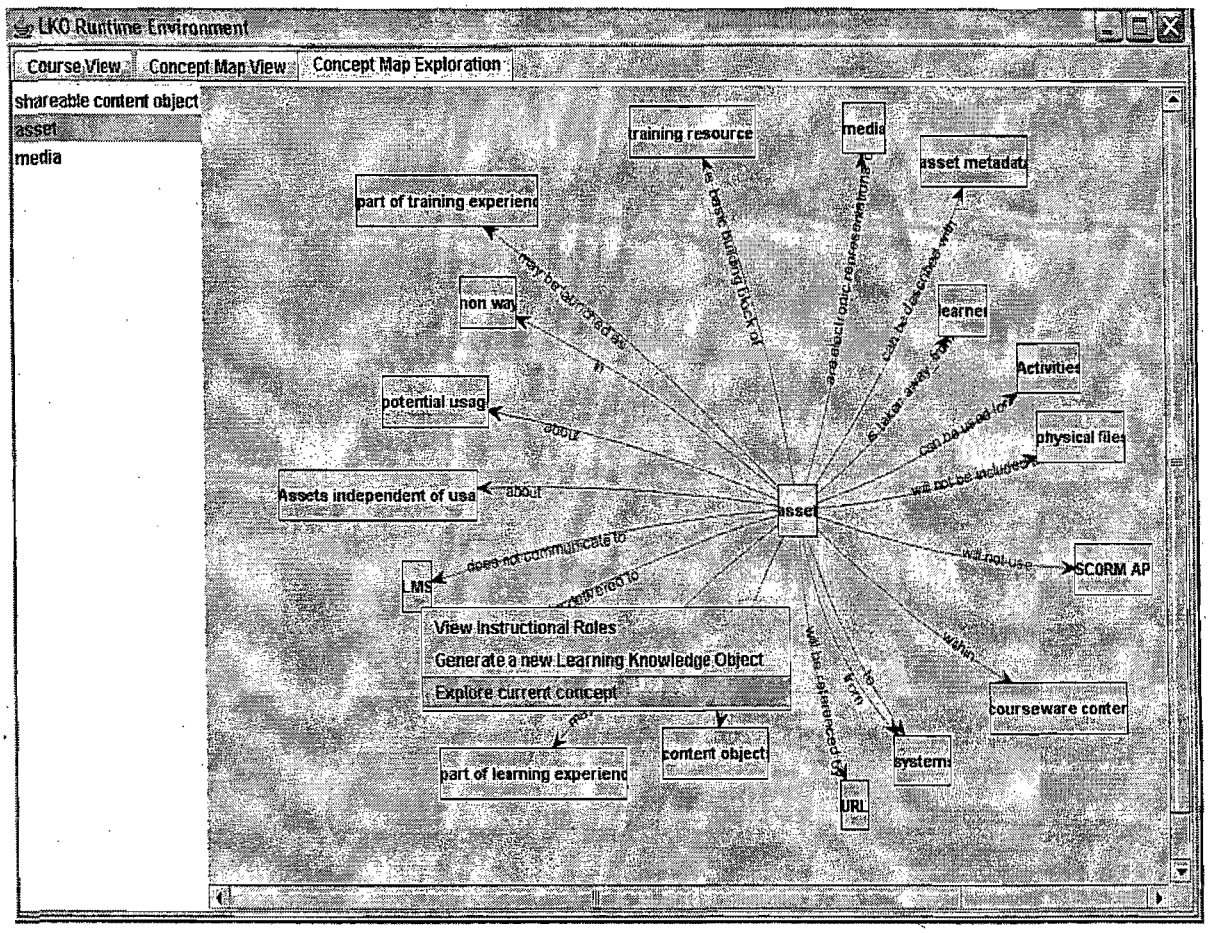


Figure 45. L'exploration des cartes de concepts

Une autre possibilité offerte dans le menu contextuel est d'explorer les différents rôles pédagogiques reliés à un concept donné. Cela peut être utile à l'apprenant aussi bien qu'au concepteur de cours car cela offre une vue complète sur les ressources disponibles autour d'un concept.

Comme le montre la figure suivante (Figure 46), le concept « *asset* » est associé à trois rôles pédagogiques : une illustration, une définition et un exemple. En cliquant sur un des rôles, il est possible d'en explorer le contenu ou même d'aller chercher le document source dont est extrait le rôle. Cela offre des ressources complémentaires d'apprentissage à

un apprenant proactif et cela offre des facilités au concepteur pour composer de nouveaux cours lorsque cette fonctionnalité est intégrée à un éditeur de cours.

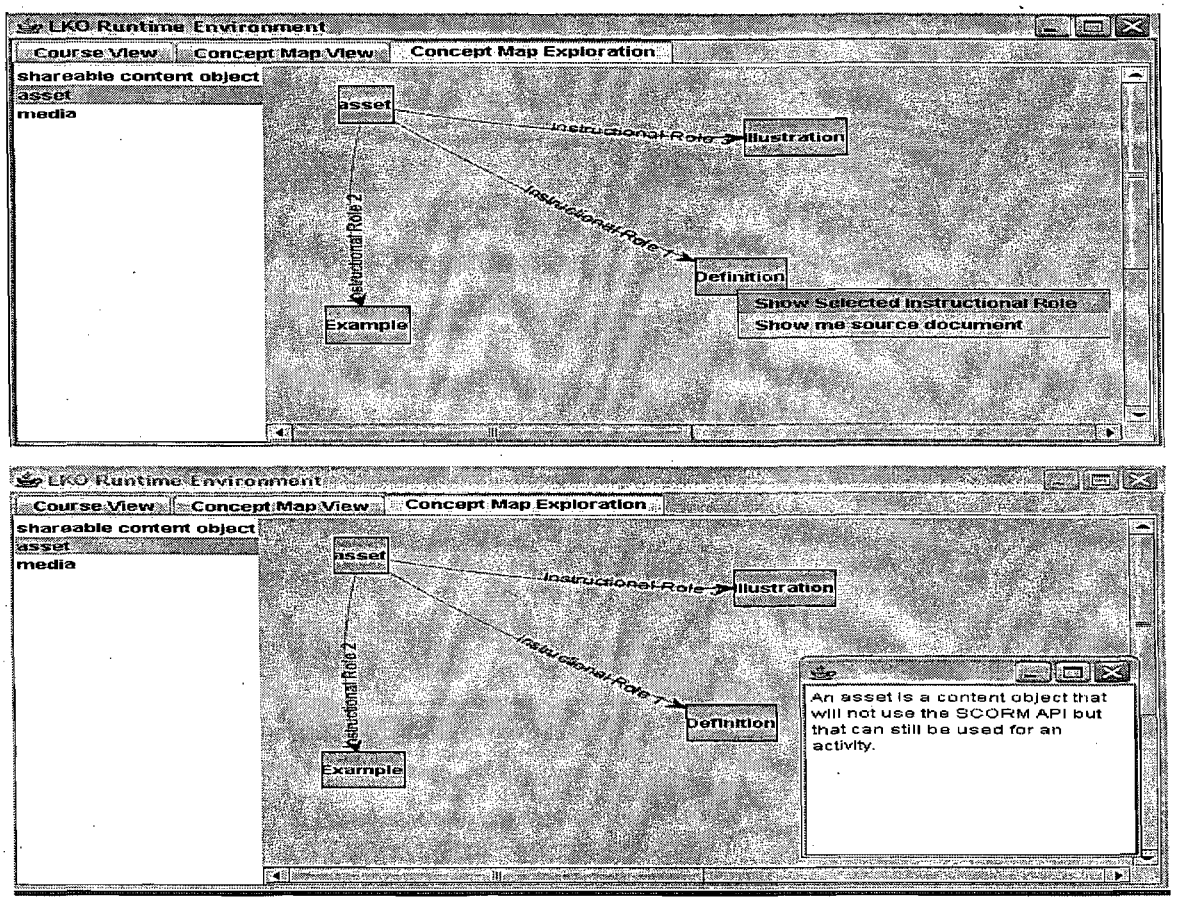


Figure 46. La vue consacrée aux rôles pédagogiques

Enfin, l'apprenant peut, à n'importe quel moment, demander la génération d'un LKO concernant un concept du contexte (à condition bien sûr que les ressources nécessaires aient été englobées dans l'agrégation de contenu au moment du déploiement).

Il est possible d'effectuer une recherche sur un concept du domaine. Une carte de concepts autour de ce concept est alors composée (ainsi qu'expliqué au chapitre 3) et il est possible de retrouver les phrases, paragraphes et les documents dont elle provient. Dans la figure 47, on peut voir l'interface de recherche de concepts. L'ontologie du domaine est chargée dans l'outil (sur la gauche) et il est possible de la parcourir afin de retrouver les

phrases dans lesquelles le concept apparaît. L'outil permet également de demander la génération de la carte de concepts du concept recherché.

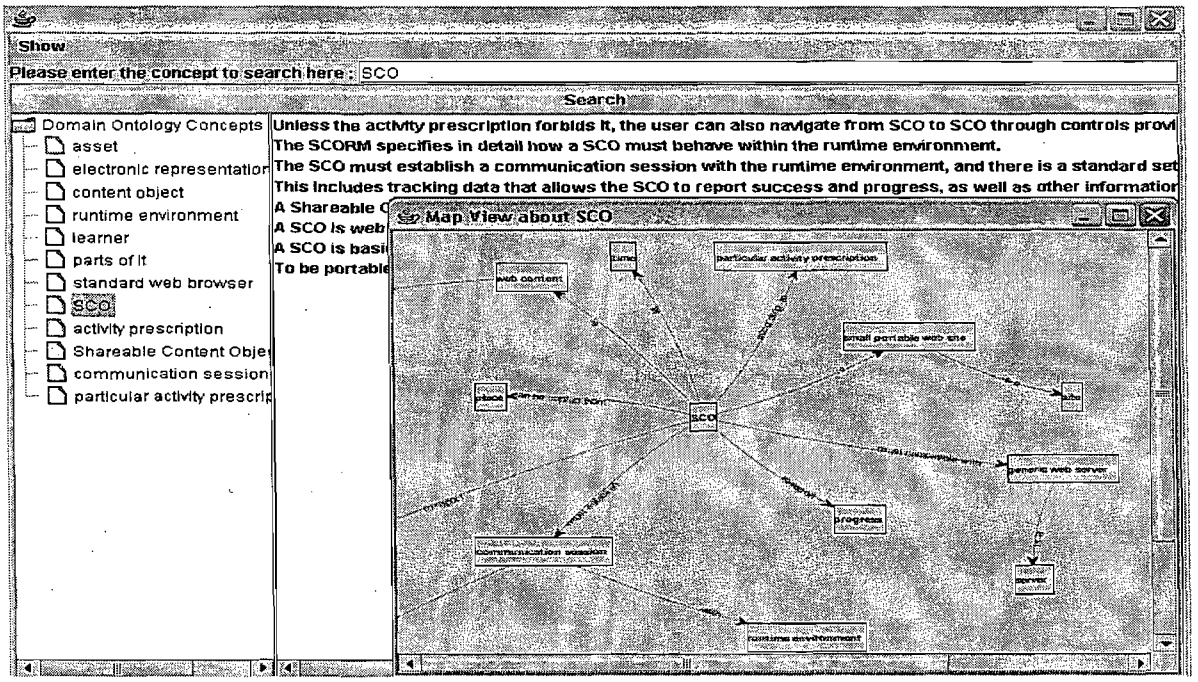


Figure 47. Interface de recherche de concepts

Enfin, le service de standardisation sert à exécuter les LKO dans différents environnements, notamment des environnements e-Learning standardisés.

6.3.2.3 Le service de standardisation

Les standards actuels du e-Learning ont montré l'importance de la réutilisation et de l'interopérabilité des objets d'apprentissage. Toutefois, ils n'ont pas mis en œuvre les ressources nécessaires à cette standardisation. Ni LOM ni IMS-LD ne désignent les rôles pédagogiques qui composent une ressource d'apprentissage. Ainsi, par exemple, LOM a une catégorie « *educational category* » sensée désigner le type pédagogique de l'objet décrit. Cette catégorie inclut entre autres les figures, les tables, les exercices, le texte narratif, etc. Outre que la liste est loin d'être exhaustive puisqu'elle occulte des rôles pédagogiques évidents tels que les définitions, ces valeurs, bien que désignant des

catégories par nature différentes, sont assemblées en une seule et unique liste confondant les genres (structurels, pédagogiques) (Ullrich, 2004). Aucune représentation ontologique n'est fournie pour désigner ces différentes catégories : les standards tels que SCORM et IMS-LD ont négligé, jusque maintenant, les représentations ontologiques du Web sémantique.

Le projet « The Knowledge Puzzle » produit des LKOs qui sont exécutables dans un environnement d'exécution SCORM (*SCORM Runtime Environment*), dans un environnement d'exécution IMS-LD (*IMS-LD Player*) ou dans n'importe quel système tutoriel. Pour cela, le LKO doit être déployé sous une forme acceptable par les trois types de systèmes. Un environnement d'exécution sous forme d'applet (*LKO Runtime Environnement*) dans une page Web est utilisé à cet effet. La couche de standardisation sert comme interface aux différents standards et environnements utilisables par un LKO. Elle comprend :

- un générateur SCORM de LKO (LKO2SCORM) qui génère des agrégations de contenu (*content package*) SCORM. Ce générateur encapsule l'applet LKO dans un patron SCORM standard (*standard SCORM Template*) ;
- un générateur IMS-LD (LKO2IMS-LD) qui génère une agrégation de contenu compatible avec IMS-LD ;
- un générateur de LKO sous forme de système tutoriel autonome (LKO2ITS).

Pour chaque standard, la même méthodologie est employée : une fois qu'un LKO est généré en fonction d'une théorie pédagogique particulière, il est possible d'exporter la structure générée sous forme d'un fichier OWL « lko.owl ». Ce fichier ou cette ontologie indique le scénario pédagogique à suivre (le plan), les différentes ressources nécessaires à chaque étape du scénario, les concepts du domaine concerné et leur contexte (c'est-à-dire leur carte de concepts associée). C'est cette ontologie qui est utilisée par l'applet LKO pour lancer l'objet de connaissances et d'apprentissage.

Pour que cette applet soit compatible avec SCORM, elle est exécutée dans une page HTML qui contient, ainsi qu'indiqué dans le standard, les appels de fonctions pour initialiser et terminer la communication entre le SCO généré et l'environnement d'apprentissage LMS (*Learning Management System*).

Un patron d'agrégation de contenu SCORM qui contient l'applet ainsi que des ressources SCORM telles que des scripts, des fichiers xsd, un manifeste etc. est alors utilisé comme modèle. Chaque fois qu'un nouveau LKO est généré, et qu'une demande d'exportation en format SCORM est effectuée, le fichier lko.owl contenant les données du LKOs est ajouté à ce modèle de même qu'un fichier « *manifest* » décrivant le LKO. Les ressources sont ensuite empaquetées dans un fichier zip. C'est ce fichier zip qui peut être importé par un environnement SCORM comme celui fourni par ADL (ADL, 2007) montré dans la figure suivante (Figure 48).

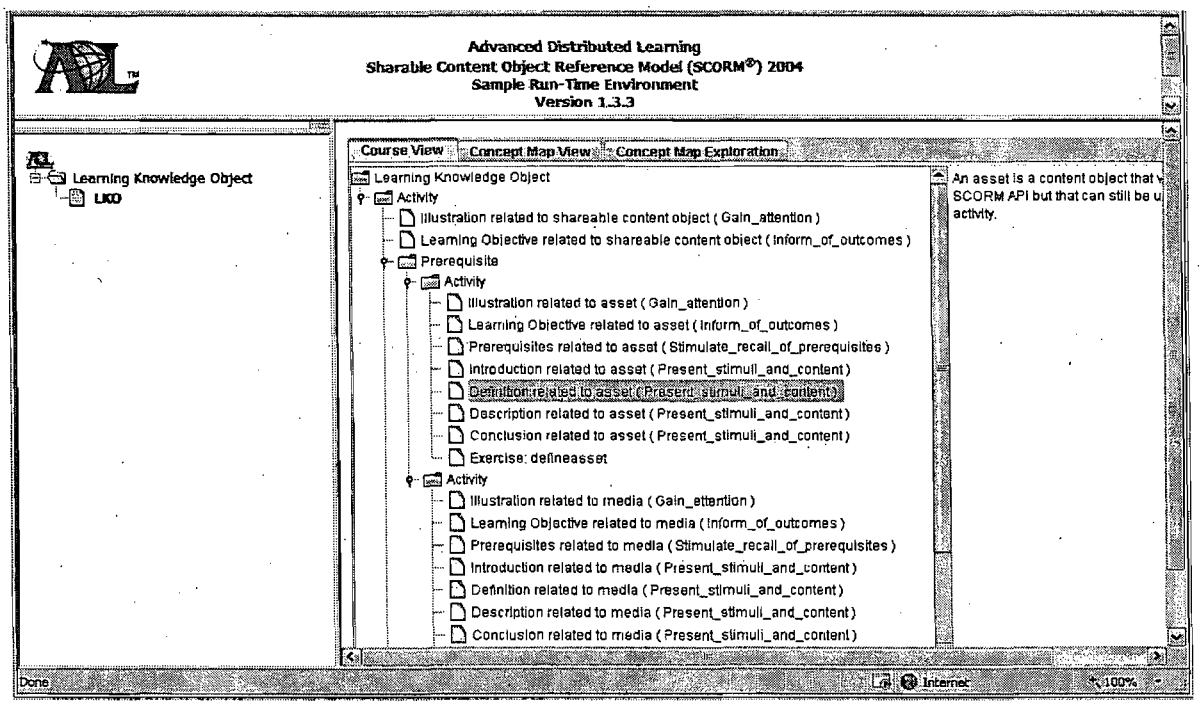
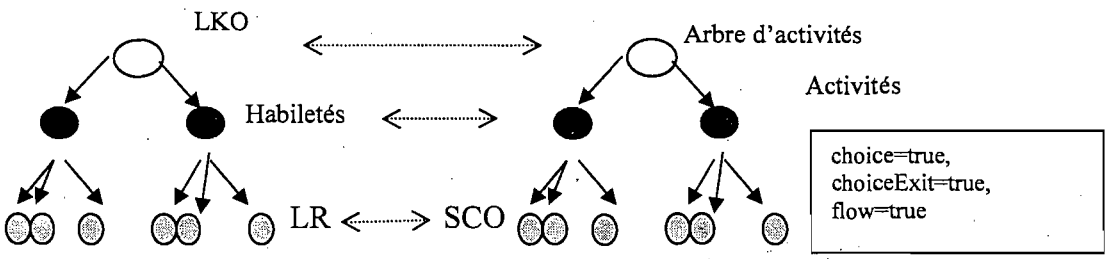


Figure 48. Exécution d'un LKO dans l'environnement d'exécution de SCORM

On peut faire le parallèle entre un LKO et une organisation de contenu SCORM. Chaque habileté à maitriser correspond à une activité qui suit une théorie donnée pour atteindre l'objectif d'apprentissage (Figure 49).



LKO : Learning Knowledge Object

SCO : Sharable Content Object

LR : Ressource d'apprentissage (rôle pédagogique ou résultat d'une méthode)

Figure 49. Correspondance conceptuelle d'un LKO dans le modèle SCORM

Un des problèmes auxquels doivent faire face les applets sont les problèmes de sécurité particulièrement si ces dernières doivent accéder à des ressources sur le disque dur (en l'occurrence le fichier lko.owl pour charger les données du LKO). Cela peut être résolu par des certifications d'authentification et des mécanismes de signatures de l'applet.

La même procédure est utilisée pour IMS-LD où un LKO est considéré comme une activité exécutable par un environnement IMS-LD comme le LDPlayer (Reload Editor and Player, 2007).

L'intérêt de ces standardisations est de permettre aux environnements e-Learning standards de bénéficier des caractéristiques des LKOs sans pour autant avoir à modifier ces standards. Toutefois, il est important de noter que l'interface de communication entre les LKO et l'environnement d'exécution SCORM est minimale et n'utilise pas le modèle de données SCORM. Une extension souhaitable serait donc d'établir une communication entre un LKO et un LMS, via le modèle de données de SCORM, de manière à transmettre des informations telles que le score, le temps écoulé, etc.

6.4 Validation des Objets d'Apprentissage et de Connaissance (LKO)

Dans un projet tel que « The Knowledge Puzzle » où de multiples problématiques sont abordées, la question qui se pose lorsque l'on veut valider l'approche, c'est quels aspects sont à évaluer ? On peut penser à évaluer l'ontologie du domaine, les cartes de concepts, les objets de connaissances et d'apprentissage (LKO), la standardisation des LKOs, l'intérêt pédagogique des LKOs, l'intérêt des théories pédagogiques pour guider le processus de composition ou même l'intérêt de spécifier des objectifs d'apprentissage sous forme de compétences à acquérir. Par ailleurs, cet aspect d'évaluation, quoique crucial dans toute discipline scientifique, devient plus délicat à manipuler lorsque des aspects cognitifs sont mis en œuvre comme c'est le cas dans les EIAH. En effet, des facteurs externes tels que l'interface de l'EIAH peuvent avoir une incidence lors de l'expérimentation.

Étant donné qu'une bonne partie de l'architecture repose sur la modélisation du domaine, et que celle-ci représente toute la partie d'acquisition automatique des connaissances à partir de textes, nous avons tout d'abord choisi de nous focaliser sur la validation de l'ontologie du domaine et des cartes de concepts. En effet, les formes d'apprentissage (didactique, constructiviste) reposent totalement sur le domaine, de même que la définition des compétences et des rôles pédagogiques. C'est cette validation que nous avons présentée dans le chapitre 4.

En ce qui concerne les objectifs d'apprentissage, leur formulation sous forme de compétences et l'utilisation de la taxonomie de Bloom (Bloom, 1956) ont déjà prouvé leur intérêt et leur efficacité dans le domaine de l'éducation (Nkambou, Frasson, & Gauthier, 2003). Par ailleurs, ainsi qu'indiqué précédemment, il est possible d'utiliser un tout autre formalisme de définition des compétences en autant qu'il reste indépendant du domaine. Ce point n'a donc pas été à nouveau validé dans le cadre de cette expérimentation.

Nous avons préalablement indiqué l'intérêt d'utiliser des théories pédagogiques afin de composer un objet d'apprentissage. L'utilité des théories dans les LKOs dépend des règles SWRL qui font le lien entre les étapes d'une théorie et les ressources d'apprentissage. Cet intérêt dépend de l'appréciation d'un expert, qui peut alors indiquer si ces règles reflètent correctement la théorie pédagogique. Cela ne dépend pas de l'architecture mise en place. Par exemple, si un concepteur humain indique qu'il faut fournir une description pour introduire un concept, alors que normalement, c'est une définition qui serait adéquate, alors la règle SWRL indiquera un rôle pédagogique « Description » (qu'il faudra ensuite manuellement corriger).

En plus de la validation de l'ontologie du domaine, nous avons indiqué la nécessité de valider les LKOs en tant qu'objets standards, c'est-à-dire de vérifier si ces objets peuvent être exécutés dans un environnement standard SCORM et IMS-LD. A cet effet, des outils ont été mis à la disposition des programmeurs sur le site de SCORM. ADL fournit un ensemble d'outils pour valider les objets d'apprentissage (dans notre cas, les LKOs) via une suite logicielle (*SCORM 2004 3rd Edition Conformance Test Suite Version 1.0 Beta (ST)*) :

- un environnement d'exécution dans lequel l'objet d'apprentissage doit pouvoir s'exécuter ;
- des outils pour valider les métadonnées générées, les agrégations de contenu, le fichier « *manifest* » ;
- des outils pour vérifier la conformité des environnements d'exécution avec les normes requises par SCORM pour les LMS (*Learning Management Systems*).

Dans notre cas, il n'y a pas de métadonnées qui accompagnent les LKOs puisqu'il n'y a pas de stockage de ces LKOs, ou même s'il y en a, elles ne sont pas d'une importance primordiale. L'important est surtout de vérifier que les LKOs sont exécutables dans l'environnement de SCORM 2004. La figure 50 montre que cela est possible, à travers le déploiement d'un LKO relatif au concept « *Shareable Content Object* ». Ce LKO offre, en plus du plan de formation didactique, l'accès aux cartes de concepts.

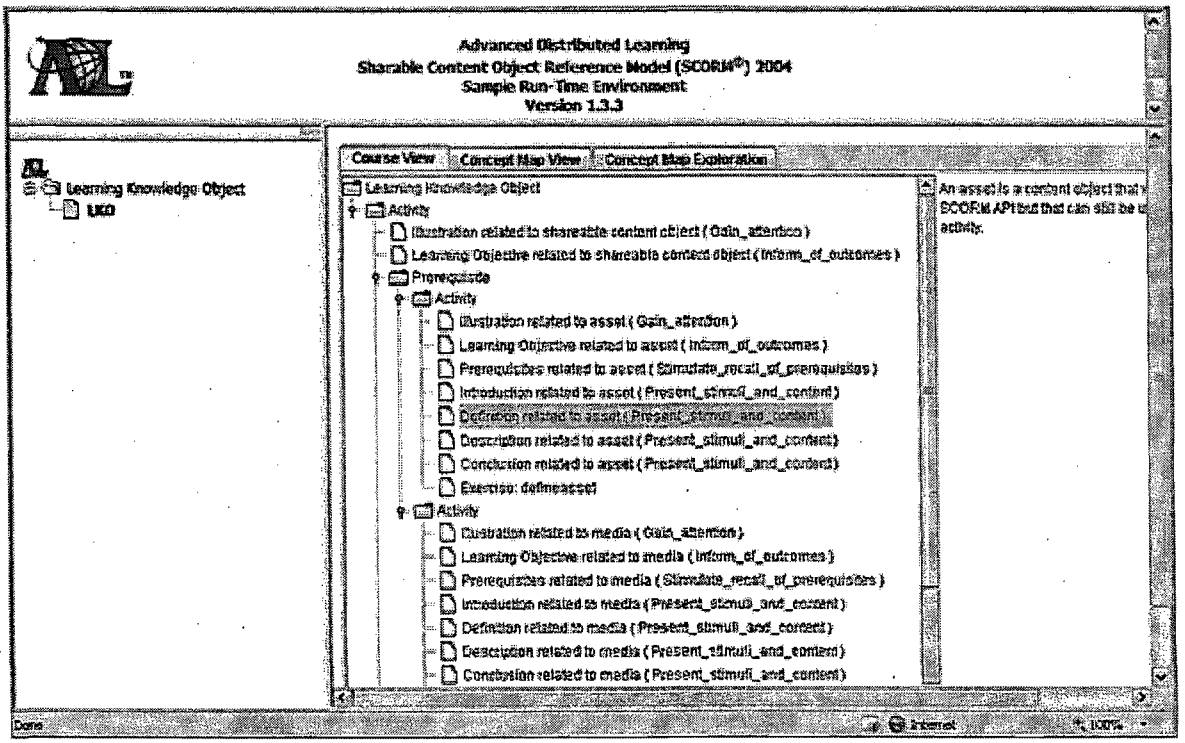


Figure 50. L'exécution d'un LKO dans le SCORM RTE 1.3.3

Par ailleurs, la même opération a été effectuée pour IMS-LD et il a été possible de lancer un LKO dans l'environnement RELOAD (Reload Project, 2004) (Reload Editor and Player, 2007).

La preuve de la possibilité du déploiement d'un LKO dans deux environnements standards est donc apportée. Le fait d'encapsuler ce LKO dans une applet n'est pas un inconvénient puisqu'il existe des possibilités de communication entre l'applet et le LMS (notamment dans le cas de SCORM).

6.5 En résumé

Nous avons présenté l'ensemble du projet « The Knowledge Puzzle » aussi bien en termes d'acquisition que d'exploitation des connaissances. La partie d'exploitation des connaissances repose sur une mémoire organisationnelle, qui stocke des fragments de

connaissances inter-reliés et structurellement guidés par des ontologies. Cette mémoire vise à offrir une alternative aux entrepôts d'objets d'apprentissage, qui souffrent de nombreuses limites, notamment l'aspect statique des ressources et leur aspect de boite noire. Cette mémoire a également comme objectif de représenter un pont entre les systèmes tutoriels intelligents et les systèmes d'apprentissage à distance via une basé de connaissances unifiée. Plutôt que de disposer d'agrégation de contenus d'apprentissage prédéfinies et annotées par un concepteur humain, la mémoire est petit à petit composée, de manière indépendante, d'un modèle du domaine, de rôles pédagogiques et de théories pédagogiques. Un peu à la manière d'un humain qui apprend en lisant des textes, la mémoire construit ces connaissances en les faisant émerger des textes « automatiquement » ou à l'aide d'un concepteur humain. Elle dispose également de buts, modélisés sous forme de compétences qui constituent des objectifs d'apprentissage. Enfin, la mémoire est dotée d'un ensemble de services tutoriels et d'interrogation des connaissances, qui permettent d'explorer son contenu et de déclencher certains mécanismes à la demande : composition et déploiement de ressources d'apprentissage, standardisation de ces ressources, exploration du domaine, génération d'exercices, etc. D'autres services pourraient être également offerts notamment des services d'explication du domaine, qui pourraient se baser sur des mécanismes de raisonnement pour définir des liens entre concepts, relier des parties de l'ontologie à certains problèmes à résoudre, bref, pour aider l'apprenant à résoudre certaines tâches et à mieux comprendre le contenu d'apprentissage.

L'apport de l'ingénierie ontologique au domaine des EIAH est donc multiple : globalement, elle permet une représentation formelle des connaissances, des mécanismes d'inférence et favorise la réutilisation, l'interopérabilité et les échanges via la notion de services tutoriels. Elle permet donc d'intégrer des mécanismes dits « intelligents » au processus de formation de manière à l'enrichir et à l'améliorer.

7 Discussion et conclusion

Cette thèse a présenté une architecture globale pour l'acquisition et l'exploitation des connaissances par un EIAH.

7.1 Les apports de la thèse

Au niveau acquisition des connaissances, nous avons implanté un modèle d'acquisition des connaissances à partir de textes, en nous inspirant principalement du domaine de la recherche d'information, du forage de données et du traitement de la langue naturelle. Notre approche, basée essentiellement sur des patrons lexico-syntaxiques, a pour caractéristique d'être non supervisée et indépendante du domaine. Contrairement aux approches supervisées en acquisition d'ontologies à partir de textes, notre méthode ne sait pas, a priori, quelles connaissances doivent être recherchées. Une objection possible serait qu'une telle approche peut ne pas forcément caractériser les spécificités d'un domaine particulier. Pour y remédier, on peut envisager d'ajouter une couche de patrons dépendants du domaine au dessus de ceux dépendant de la syntaxe, ce qui préserve l'indépendance et la flexibilité de notre proposition.

L'intérêt de notre modèle réside notamment dans son approche d'extraction en deux phases : une première phase permettant d'extraire des cartes de concepts à partir de textes, et une seconde de dériver une ontologie du domaine à partir des cartes de concepts. Une telle démarche, à notre connaissance, n'a pas été tentée jusque là. Elle permet d'intégrer deux représentations, une informelle (les cartes de concepts) et une formelle (l'ontologie du domaine), ce qui est d'un grand intérêt pour notre domaine d'application : la formation par ordinateur. En effet, les cartes de concepts ont prouvé leur intérêt dans le domaine de l'éducation. Certaines approches ont été proposées pour dériver des cartes de concepts à partir de textes mais les patrons utilisés et leurs représentations étaient moins riches. Ces cartes de concepts permettent de représenter des chemins d'information, c'est-à-dire de

schématiser la connaissance contenue dans les textes de manière à obtenir une vue globale et synthétique du contenu des textes. Ensuite, ces cartes de concepts sont converties en ontologie du domaine. Cela nécessite des réflexions sur la nature ontologique des concepts et relations. Sans avoir trouvé une réponse parfaite à ce sujet, nous avons tenté un début de réponse en utilisant des paramètres provenant de la théorie des graphes (degré sortant d'un concept, métriques concernant la richesse de description d'un concept, centralité d'un concept, etc.). Nous demeurons convaincus que d'autres paramètres doivent participer à cette nature ontologique. On pourrait notamment utiliser les métriques d'Alani et Brewster (Alani & Brewster, 2006), consacrées dans cette thèse à l'évaluation des ontologies, pour paramétrer la génération de l'ontologie.

Quel type d'ontologie obtenons-nous ? Une ontologie nette, exempte d'erreurs ? Certainement pas ! L'ontologie ne contient, par exemple, ni axiomes, ni disjonctions, et peu de classes définies, ces dernières étant pourtant nécessaires au processus d'inférence de nouvelles connaissances. Dans (Hendler, 2001), l'auteur écrit :

"The Semantic Web... will not primarily consist of neat ontologies that expert AI researchers have carefully constructed. I envision a complex web of semantics ruled by the same sort of anarchy that rules the rest of the web".

Dans cette citation, l'idée de base est qu'il ne faut pas s'attendre à ce que les ontologies générées soient parfaites. Malgré leurs imperfections, l'ingénieur d'ontologies doit essayer d'en tirer tout le bénéfice possible. Nous pouvons dire que nous obtenons un squelette d'ontologie du domaine, que le concepteur doit compléter et valider s'il veut disposer d'une base de connaissances complète. Toutefois, si le but est surtout d'indexer les textes dont est issue l'ontologie en question, alors moins de travail est requis : l'ingénieur ontologique doit notamment s'assurer de la plausibilité des concepts et relations générés. Notre modèle en trois couches (niveau textes, niveau cartes de concepts, niveau ontologie) permet d'ailleurs de raffiner cette indexation : on peut facilement naviguer des textes aux

cartes de concepts et ensuite des cartes de concepts vers l'ontologie du domaine et vice-versa.

Cette thèse a également proposé une méthodologie d'évaluation de l'ontologie du domaine en trois phases : l'analyse structurelle, l'analyse comparative et l'analyse sémantique. Cela nous a permis d'approfondir certaines métriques et nous a conduit à comparer notre outil de génération d'ontologies, TEXCOMON, avec un des outils les mieux connus de ces dernières années : TEXT-TO-ONTO. L'équipe qui l'a réalisé est d'ailleurs à l'origine de l'engouement pour l'acquisition automatique d'ontologies à partir de textes. Dans tous ces tests, nous avons pu constater que notre approche produisait de meilleurs résultats en termes de pertinence des concepts et relations générés.

Côté exploitation des connaissances, l'un des points forts de cette thèse est son application à un domaine concret : la formation, et l'établissement d'une approche commune aux EIAH via une base de connaissances intégrée, constituée d'une mémoire à base d'ontologies. Cette mémoire vise à remplacer les entrepôts d'objets d'apprentissage, que nous considérons comme dépassés et inappropriés pour répondre aux besoins actuels des EIAH : un besoin d'individualisation des formations, un besoin de représentation des connaissances du domaine, un besoin de formalisation d'objectifs d'apprentissage, un besoin de composition automatique de ressources d'apprentissage ciblées, etc. Tout cela ne peut être réalisé avec les structures hétérogènes existantes. La mémoire organisationnelle vise à représenter les connaissances contenues dans les objets d'apprentissage via notre méthode de génération de cartes de concepts et d'ontologies du domaine, via la détection de structures de documents, et via la représentation de rôles pédagogiques, fragments nécessaires à des mécanismes d'agrégation automatique. Les objectifs d'apprentissage sont représentés par une ontologie des compétences. La mémoire est également dotée d'une expertise pédagogique sous forme d'une ontologie des théories pédagogiques. Des services tutoriels permettent ensuite de composer des objets de connaissances et d'apprentissage, c'est-à-dire des ressources adaptées à un apprenant, possédant un modèle du domaine, actives (possédant des connaissances et des méthodes pour agir sur ces connaissances),

autonomes puisque autosuffisantes, et enfin temporaires (elles ne nécessitent pas forcément de stockage dans un entrepôt). Tout l'intérêt du modèle est dans ces caractéristiques que ne possédaient pas jusque là les objets d'apprentissage. L'autre point fort du projet est la possibilité de standardiser ces objets de connaissances et d'apprentissage, de manière à permettre leur intégration dans les environnements e-Learning standards sans nécessiter de changement dans les standards en question. Cette intégration confère à ces environnements les avantages cités ci-dessus et donc une capacité supérieure à ce qui existe actuellement dans le e-Learning.

7.2 Limites et perspectives

Dans cette section, nous abordons les limites de l'approche et de l'implantation proposées et définissons les perspectives du projet en acquisition et exploitation des connaissances. Notons, sur un plan global, que la suite logicielle (atelier) présentée dans cette thèse doit pouvoir être utilisée par la communauté éducative dans un avenir proche. Certaines améliorations sont toutefois requises avant de procéder à la mise en exploitation de l'atelier.

7.2.1 En acquisition des connaissances

La base de patrons lexico-syntaxiques utilisés par TEXCOMON est loin d'être complète et ne permet d'extraire que certaines représentations. Il serait intéressant d'identifier, conjointement avec des experts linguistes, de nouveaux patrons lexico-syntaxiques associés à des représentations sémantiques. Cette détection pourrait bénéficier de méthodes d'apprentissage automatique de patrons dans un corpus.

Par ailleurs, un patron lexico-syntaxique est actuellement modélisé sous forme de sous-arbre composé essentiellement d'un nœud racine et de liens entrants et sortants. Il serait nécessaire de modéliser des structures plus complexes incluant plusieurs nœuds et plusieurs liens : la prochaine étape serait de se pencher sur des travaux comme ceux de (Shasha, Wang, & Giugno, 2002) pour la recherche de patrons dans les graphes. Il serait

également intéressant d'étudier d'autres manières d'implanter le mécanisme de modélisation des patrons sous forme de règles. Cela nécessite un approfondissement de la notion de patron et de sa modélisation. Cela nécessite également de faire des recherches dans le domaine des grammaires en linguistique computationnelle : comment spécifier un patron et sa méthode de transformation sous forme d'un langage déclaratif ?

L'apprentissage automatique de patrons récurrents serait également une amélioration intéressante. Cela permettrait de détecter de nouvelles structures qui devraient être complétées par des méthodes de transformation sémantique.

D'autres traitements linguistiques seraient nécessaires en plus de ceux qui existent actuellement, notamment la détection d'anaphores et de coréférences plus poussées. Cela est d'autant plus important que de nombreuses connaissances sont occultées lorsqu'elles figurent dans des phrases dont le sujet principal est un pronom personnel. Il faudrait également pouvoir gérer les synonymes et repérer des classes de termes en se basant sur une approche plus approfondie que la seule lemmatisation.

Par ailleurs, le passage d'une carte de concepts à une ontologie du domaine peut être influencé par de nombreuses métriques telles que le degré de connexion des concepts ou les métriques définies dans (Alani & Brewster, 2006). Cet aspect doit être creusé et les relations entre terminologie, linguistique et ontologies doivent être beaucoup plus profondément traitées, nous pensons notamment aux réflexions de Roche dans (Roche, 2006a).

Notons que les services de génération d'une ontologie domaine semblent occulter la dimension consensuelle propre aux ontologies. Cela est effectivement le cas si l'on considère que le consensus doit émerger d'une validation de l'ontologie finale par une communauté donnée (ce qui n'a pas été clairement discuté dans ce travail). Cela n'est pas le cas si on utilise comme source de connaissances des textes ayant donné lieu à un certain consensus[f] sur le domaine de connaissances. Il n'en demeure pas moins que cette dimension nécessite un examen plus approfondi que ce qui a été accompli par cette thèse. A

ce niveau et à ce stade du travail, nous envisageons un entrepôt d'objets d'apprentissage qui soit progressivement enrichi et mis à jour par une communauté dans un domaine donné. A une certaine étape, nous estimons que cet entrepôt devrait être suffisamment riche pour permettre la génération d'une ontologie du domaine, telle que nous l'avons présentée dans cette thèse. Ensuite, l'analyse de l'ontologie résultante devrait permettre à la communauté de détecter des incohérences ou manques de connaissances, ce qui devrait donner lieu à la création de nouvelles ressources et à une nouvelle génération de l'ontologie. Après un certain cycle d'itérations, l'ontologie devrait atteindre une certaine maturité et être validée par la communauté. La prise en charge de la mise à jour de l'ontologie par des connaissances non prévues au départ devrait alors s'appuyer sur des mécanismes d'enrichissement d'une ontologie existante. En effet, cette thèse a traité le problème de la génération d'une ontologie sans considérer d'éventuels éléments du domaine déjà existants (sous forme d'ontologie, de thésaurus, de mots-clés, etc.) et en créant systématiquement une nouvelle ontologie. Il serait toutefois nécessaire d'élargir notre approche par la possibilité d'enrichir une ontologie existante à partir de nouvelles ressources. Cela nécessitera d'autres techniques que celles présentées dans cette thèse (évoquées toutefois dans l'état de l'art) comme par exemple la capacité de situer un nouveau concept dans la hiérarchie existante, ou d'apparier un nouveau concept avec un concept existant, etc.

7.2.2 En exploitation des connaissances

Au niveau de l'exploitation des connaissances, les LKOs pourraient bénéficier de nombreuses améliorations. Il faudrait notamment peaufiner les aspects suivants :

Tout d'abord, la modélisation des théories pédagogiques devrait s'inspirer de modèles plus complexes que celui adopté tels que ceux proposés dans (Mizoguchi, Hayashi, & Bourdeau, 2007). D'autre part, la modélisation des règles SWRL devrait être repensée dans un cadre plus large. S'il est facile de modéliser le niveau de connaissance et de description de Bloom, cela est-il aussi facile pour les niveaux subséquents ? Il serait

également intéressant d'investiguer d'autres taxonomies/terminologies pour exprimer les compétences de manière indépendante du domaine.

Ensuite, il faudrait approfondir le cadre de la composition automatique des ressources et réfléchir aux conditions et paramètres nécessaires pour réussir cette composition. En effet, nous n'ignorons pas les problèmes qui peuvent découler de l'assemblage de ressources trop hétérogènes (en terme de style, de cohésion, de liens entre les composants, etc.) et davantage de réflexion est nécessaire à ce niveau.

Par ailleurs, il faudrait enrichir les LKOs standardisés en SCORM par des éléments de données provenant du modèle de SCORM. Actuellement les communications entre le LMS et le LKO se limitent à l'initialisation et à la terminaison d'une session d'apprentissage, et tout le reste (progression, score, exercices, etc.) est géré par le LKO. Il faudrait que ce dernier puisse communiquer les résultats de l'apprenant au LMS.

7.2.3 En évaluation de la plateforme « *The Knowledge Puzzle* »

L'évaluation de la plateforme dans son ensemble et plus particulièrement de TEXCOMON doit être davantage creusée. En effet, nous souhaitons évaluer le taux de couverture du corpus par les ontologies générées de façon à connaître les connaissances manquantes. Par ailleurs, il serait intéressant d'effectuer l'inverse de ce qui a été fait dans l'évaluation en trois étapes, et de procéder à l'évaluation structurelle et comparative une fois l'analyse sémantique effectuée. L'entrée de ces deux évaluations serait alors des ontologies déjà corrigées par l'expert au travers de la suppression des classes et relations non pertinentes. Quels seront alors les résultats de l'analyse structurelle et surtout de la comparaison de TEXCOMON avec TEXT-TO-ONTO ?

En plus de TEXT-TO-ONTO, nous souhaitons comparer TEXCOMON avec d'autres outils disponibles notamment TEXT2ONTO (Cimiano & Völker, 2005), ONTOGEN (Fortuna, Grobelnik, & Mladenič, 2006a) et TOKO (Anjewierden & Efimova,

2006). Nous souhaitons également l'exécuter sur d'autres ressources documentaires et d'autres domaines.

Enfin, nous nous sommes focalisés dans cette thèse sur une évaluation analytique et technique des solutions proposées. Parallèlement à cette évaluation, de nombreux autres aspects déjà évoqués et reliés à l'intérêt pédagogique des LKOs devraient être testés de manière empirique avec un ensemble d'apprenants. De manière générale, il faudrait procéder à une évaluation de la facilité d'utilisation de l'atelier auprès d'enseignants-auteurs et effectuer une évaluation de l'impact des LKO sur des apprenants. Dans ce cadre, l'impact de l'interface et de la présentation des cartes de concept devra être clairement mesuré.

Bibliographie

Abel, M.-H., Benayache, A., Lenne, D., Moulin, C., Barry, C., & Chaput, B. (2004). Ontology-based Organizational Memory for e-learning. *Journal of Educational Technology & Society*, *7* (4), 98-111.

ADL. (2007). Consulté le 9 5, 2007, sur http://www.adlnet.org

Agichtein, Y. (2005). *Extracting Relations from Large Text Collections.* New York: Columbia University (Ph.D. Thesis).

Agichtein, Y., & Gravano, S. (2000). Snowball: Extracting relations from large plain-text collections. *Proceedings of the 5th ACM International Conference on Digital Libraries* (pp. 85 - 94). New York: ACM.

Agirre, E., Ansa, O., Hovy, E., & Martinez, D. (2000). Enriching Very Large Ontologies using the WWW. *Proceedings of the Workshop on Ontology Construction of the European Conference of A.I. (ECAI-00). 31.* Berlin: CEUR-WS.org.

Alani, H., & Brewster, C. (2006). Metrics for Ranking Ontologies. *Proceedings of the 4th International EON Workshop, 15th Int. World Wide Web Conference.* Edinburgh.

Algernon. (2007). Consulté le 11 19, 2007, sur Algernon: http://algernon-j.sourceforge.net/doc/algernon-protege.html

Anjewierden, A., & Efimova, L. (2006). Understanding weblog communities through digital traces: a framework, a tool and an example. *Proceedings of the International Workshop on Community Informatics (COMINF 2006). LNCS 4277*, pp. 279-289. Montpellier: Springer.

Aroyo, L., & Dicheva, D. (2004). The New Challenges for E-learning: The Educational Semantic Web. *Educational Technology & Society*, *7* (4), 59-69.

Artale, A., Franconi, E., & Guarino, N. (1996). Open Problems with Part-Whole Relations. *Proceedings of 1996 International Workshop on Description Logics (DL-96)* (pp. 70-73). Cambridge: AAAI Press.

Aussenac-Gilles, N., Biébow, B., & Szulman, N. (2000). Revisiting ontology design: a method based on corpus analysis. *Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management* (pp. 172-188). Berlin: Springer-Verlag.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007). Open Information Extraction from the Web. *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, (pp. 2670–2676). Hyderabad.

Baroni, M., & Bisi, S. (2004). Using cooccurrence statistics & the web to discover synonyms in a technical language. *Proceedings of the 4th International Conference on Language Resources and Evaluation. 5*, pp. 1725–1728. Lisbon: ELDA/LREC-2004.

Barritt, C., Lewis, D., & Wieseler, W. (1999). *Cisco.* Consulté le 12 28, 2007, sur http://www.cisco.com/warp/public/779/ibs/solutions/learning/whitepapers/el_cisco_rio.pdf

Benayache, A. (2005). *Construction d'une mémoire organisationnelle de formation et évaluation dans un contexte e-learning : le projet MEMORAe.* Compiègne: Université de technologie de Compiègne (Thèse de doctorat).

Bergsträßer, S., Rensing, C., Zimmermann, B., & Steinmetz, R. (2007). Building a Representation of Learning Resources to Support their Re-Purposing. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 3128-3136). Chesapeake: AACE.

Berland, M., & Charniak, E. (1999). Finding parts in very large corpora. *Proceedings of the 37th Annual Meeting of the Association For Computational Linguistics on Computational Linguistics* (pp. 57 - 64). Morristown: Association for Computational Linguistics.

Bernard, G. (2000). *La linguistique et l'intelligence artificielle.* Consulté le 11 14, 2007, sur http://www.ai.univ-paris8.fr/CSAR/Travaux/Ling_IA.pdf

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic Web. *Scientific American Magazine , 5* (1), 34–43.

Bisson, G., Nedellec, C., & Canamero, L. (2000). Designing clustering methods for ontology building - The Mo'K workbench. *Proceedings of the ECAI Ontology Learning Workshop* (pp. 13–19). Berlin: CEUR-WS.org.

Bloom, B. (1956). *Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain.* New York: Longman.

Bourdeau, J., Mizoguchi, R., Psyché, V., & Nkambou, R. (2004). Selecting Theories in an Ontology-Based ITS Authoring Environment. *Proceedings of the 7th International Conference (ITS 2004). LNCS 3220*, pp. 150-161. Maceiò: Springer.

Bourigault, D., & Aussenac-Gilles, N. (2003). Construction d'ontologies à partir de textes. *Actes de la 10ème conférence annuelle sur le Traitement Automatique des Langues (TALN 2003)*, (pp. 27-50). Batz-sur-Mer.

Bozsak, E. e., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., et al. (2002). KAON - Towards a large scale Semantic Web. *Proceedings of the Third International Conference on E-Commerce and Web Technologies (EC-Web)* (pp. 304-313). Aix-en-Provence: Springer.

Brachman, R. J., & Schmolze, J. G. (1985). An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science , 9* (2), 171-216.

Brooks, C., & McCalla, G. (2006). Towards flexible learning object metadata. *International Journal of Cont. Engineering Education and Lifelong Learning , 16* (1/2), 50-63.

Brooks, C., McCalla, G., & Winter, M. (2005). Flexible Learning Object Metadata. *Proceedings of the 3rd International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL 05) held in conjunction with the 12th International Conference on Artificial Intelligence in Education (AIED 2005)*, (pp. 1-8). Amsterdam.

Brusilovsky, P., Schwarz, E. W., & Weber, G. (1996). ELM-ART: An Intelligent Tutoring System on World Wide Web. *Proceedings of the 3rd International Conference on Intelligent Tutoring Systems. LNCS 1086*, pp. 261 - 269. Springer-Verlag: London.

Buffa, M., Dehors, S., Faron-Zucker, C., & Sander, P. (2005). Towards a Corporate Semantic Web Approach in Designing Learning Systems: Review of the TRIAL SOLUTION Project. *Proceedings of the International Workshop on Applications of Semantic Web Technologies for E-Learning, AIED2005.* Amsterdam.

Buitelaar, P., Cimiano, P., Grobelnik, M., & Sintek, M. (2005, October 3). Ontology Learning from Text. *Tutorial at ECML/PKDD 2005* . Porto, Portugal.

Buitelaar, P., Olejnik, D., & Sintek, M. (2004). OntoLT: A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. *Proceedings of the 1st European Semantic Web Symposium (ESWS)* (pp. 31-44). Heidelberg: Springer-Verlag.

Burgos, D., Arnaud, M., Neuhauser, P., & Koper, R. (2005, décembre). IMS Learning Design: la flexibilité pédagogique au service des besoins de l'e-formation. *La Revue de l'EPI* .

Burns, H. L., & Capps, C. G. (1988). Foundations of Intelligent Tutoring Systems: An Introduction. In *Foundations of Intelligent Tutoring Systems.* Lawrence Erlbaum Associates Publishers.

Byrd, R., & Ravin, Y. (1999). Identifying and extracting relations from text. *Proceedings of 4th International Conference on Applications of Natural Language to Information Systems.* Klagenfurt.

Cardinaels, K., Meire, M., & Duval, E. (2005). Automating metadata generation: the simple indexing interface. *Proceedings of the 14th international conference on World Wide Web* (pp. 548 - 556). New York: ACM.

Carrol, J., Minnen, G., & Briscoe, T. (1999). Corpus annotation for parser evaluation. *Proceedings of the EACL workshop on Linguistically Interpreted Corpora (LINC)*, (pp. 35-41). Bergen.

Cederberg, S., & Widdows, D. (2003). Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hyponymy Extraction. *Proceedings of the 7th Conference on Natural Language Learning (CoNLL-2003)*, (pp. 111-118). Edmonton.

Cimiano, P., & Staab, S. (2005). Learning Concept Hierarchies from Text with a Guided Agglomerative Clustering Algorithm. *Proceedings of the ICML 2005 Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods.* Bonn.

Cimiano, P., & Völker, J. (2005). Text2Onto – A Framework for Ontology Learning and Data-driven Change Discovery. *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB)* (pp. 227-238). Alicante: Springer.

Cimiano, P., & Wenderoth, J. (2007). Automatic Acquisition of Ranked Qualia Structures from the Web. *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*. Prague.

Cimiano, P., Hotho, A., & Staab, S. (2004a). Clustering ontologies from text. *Proceedings of the 4th International Conference on Language Resources and Evaluation*, (pp. 1721-1724). Artipol.

Cimiano, P., Hotho, A., & Staab, S. (2004b). Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text. *Proceedings of the 16th European Conference on Artificial Intelligence*, (pp. 435–439). Valencia.

Cimiano, P., Pivk, A., Schmidt-Thieme, L., & Staab, S. (2004). Learning Taxonomic Relations from Heterogeneous Sources of Evidence. In *Ontology Learning from Text: Methods, Evaluation and Applications*. IOS Press.

Clariana, R. B., & Koul, R. (2004). A Computer-Based Approach for Translating Text into Concept Map-like Representations. *Proceedings of the 1st International Conference on Concept Mapping*, (pp. 131-134). Pamplona.

Claveau, V. (2003). *Acquisition automatique de lexiques sémantiques pour la recherche d'information.* Thèse de doctorat, Université de Rennes 1, Rennes.

Cohen, J. D. (1995). Highlights: Language and domain independent automatic indexing terms for abstracting. *Journal of the American Society for Information Science , 46* (3), 162–174.

Collier, N., Nobata, C., & Tsujii, J. (2000). Extracting the Names of Genes and Gene Products with a Hidden Markov Model. *Proceedings of COLING 2000* (pp. 201-207). Morristown: Association for Computational Linguistics.

CopperCore project website. (2007). Consulté le 10 20, 2007, sur CopperCore: www.coppercore.org

Corcho, O., & Gómez-Pérez, A. (2000). Evaluating Knowledge Representation and Reasoning Capabilities of Ontology Specification Languages. *Proceedings of the ECAI 2000 Workshop on Applications of Ontologies and Problem-Solving Methods*. Berlin.

Covington, M. A. (2001). A fundamental algorithm for dependency parsing. *Proceedings of the 39th Annual ACM Southeast Conference* (pp. 95–102). New York: ACM.

Cruze, D. A. (1986). *Lexical Semantics*. Cambridge: Cambridge University Press.

Cunningham, D., Maynard, D., Bontcheva, K., & Tablan, V. (2002). GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*. Philadelphia.

De Marneffe, M.-C., MacCartney, B., & Manning, C. D. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2006)*. Genoa.

Dehors, S., Faron-Zucker, C., Giboin, A., & Stromboni, J.-P. (2005). Semi-automated Semantic Annotation of Learning Resources by Identifying Layout Features. *Proceedings of the International Workshop on Applications of Semantic Web Technologies for E-Learning, AIED2005*. Amsterdam.

Dekang, L. (1998). Automatic Retrieval and Clustering of Similar Words. *Proceedings of the 17th international conference on Computational linguistics* (pp. 768 - 774). Morristown: Association for Computational Linguistics.

Desmoulins, C., & Grandbastien, M. (2002). Ontologies pour la conception de manuels de formation à partir de documents techniques. *STE , 9* (3-4), 291-340.

Devedžić, V. (2004). Education and the Semantic Web. *International Journal of Artificial Intelligence in Education* , 39-65.

Di Eugenio, B., Fossati, D., Yu, D., Haller, S., & Glass, M. (2005). Aggregation improves learning:experiments in natural language generation for intelligent tutoring systems. *Proceedings of the 42nd Meeting of the Association for Computational Linguistics* (pp. 50-57). Morristown: Association for Computational Linguistic.

Dolog, P., & Nejdl, W. (2003). Challenges and Benefits of the Semantic Web for User Modeling. *Proceedings of AH2003 workshop at 12th World Wide Web Conference.* Budapest.

Dolog, P., & Schaefer, M. (2005). Learner Modeling on the Semantic Web. *Proceedings of PerSWeb-2005 Workshop: Personalization on the Semantic Web at User Modeling 2005: 10th International Conference.* Edinburgh.

Downey, D., Etzioni, O., & Soderland, S. (2005). A Probabilistic Model of Redundancy in Information Extraction. *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI)*, (pp. 1034-1041). Edinburgh.

Dublin Core Metadata Initiative. (1995). Consulté le 10 14, 2007, sur http://dublincore.org/

Educational Data Mining. (2007). *Proceedings of the Workshop of Educational Data Mining.* Marina Del Rey.

Eskridge, T., Hayes, P., Hoffman, R., & Warren, M. (2006). Formalizing the informal: A confluence of Concept Mapping and the semantic web. *Proceedings of the Second International Conference on Concept Mapping* (pp. 247-254). San Jose: Universidad de Costa Rica.

Etzioni, O., Kok, S., Soderland, S., Cafarella, M., Popescu, A.-M., Weld, D. S., et al. (2004). WebScale Information Extraction in KnowItAll (Preliminary Results). *Proceedings of the 13th international conference on World Wide Web* (pp. 100-110). New York: ACM.

Faure, D., & Nedellec, C. (1998). A Corpus-based conceptual clustering method for verb frames and ontology acquisition. *Proceedings of Adapting Lexical and Corpus Resources to Sublanguages and Applications, Workshop of the 1st International Conference on Language Resources and Evaluation (LREC)*, (pp. 1-8). Granada.

Faure, D., & Poibeau, T. (2000). First experiments of using semantic knowledge learned by ASIUM for information extraction task using INTEX. *Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence ECAI'00*, (pp. 7-12). Berlin.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* Cambridge: MIT Press.

Fortuna, B., Grobelnik, M., & Mladenič, D. (2006a). Background Knowledge for Ontology Construction. *Proceedings of the 15th International Conference on World Wide Web* (pp. 949-950). New York: ACM Press.

Fortuna, B., Grobelnik, M., & Mladenič, D. (2006b). System for semi-automatic ontology construction. *Demo at the 3rd European Semantic Web Conference ESWC-2006.* Budva.

Fortuna, B., Mladevic, D., & Grobelnik, M. (2005). Visualization of Text Document Corpus. *Informatica* , 497-504.

Fournier-Viger, P., Najjar, M., Mayers, A., & Nkambou, R. (2006). From Black-box Learning Objects to Glass-Box Learning Objects. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems (ITS 2006). LNCS 4053*, pp. 258-267. Berlin: Springer-Verlag.

Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., & Nevill-Manning, C. G. (1999). Domain-specific keyphrase extraction. *Proceedings of the 16th International Joint Conference on Artificial Intelligence* (pp. 668-673). San Francisco: Morgan Kaufmann Publishers.

Friedman-Hill, E. (2003). *Jess in Action: Java Rule-Based Systems* . Manning Publications Co.

Gagné, R. M., Briggs, L. J., & Wagner, W. W. (1992). *Principles of Instructional Design (4th Ed.).* Fort Worth: HBJ College Publishers.

Gandon, F. (2002). A Multi-Agent Architecture For Distributed Corporate Memories. *Proceedings of the 3rd International Symposium, From Agent Theory to Agent*

Implementation, at the 16th European Meeting on Cybernetics and Systems Research (EMCSR 2002), (pp. 623-628). Vienna.

Ganter, B., & Wille, R. (1999). *Formal Concept Analysis – Mathematical Foundations.* Heidelberg: Springer.

Gargouri, Y., Lefebvre, B., & Meunier, J. G. (2004). ONTOLOGICO : vers un outil d'assistance au développement itératif des ontologies. *Journées d'études sur Terminologie, Ontologie, et Représentation des connaissances (TERMINO'2004).* Lyon.

Gasevic, D., Jovanovic, J., & Devedzic, V. (2004). Ontologies for Creating Learning Object Content. *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information & Engineering Systems* (pp. 284-291). Wellington: Springer.

Girju, R. (2003). Automatic Detection of Causal Relations for Question Answering. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003), Workshop on "Multilingual Summarization and Question Answering - Machine Learning and Beyond"*. (pp. 76-83). Morristown: Association for Computational Linguistics.

Girju, R., & Moldovan, D. (2002). Text Mining for Causal Relations. *Proceedings of the International Florida Artificial Intelligence Research Society (FLAIRS 2002)* (pp. 360 -364). AAAI Press.

Girju, R., Badulescu, A., & Moldovan, D. I. (2006). Automatic Discovery of Part-Whole Relations. *Computational Linguistics , 32* (1), 83-135.

Girju, R., Badulescu, A., & Moldovan, D. (2003). Learning Semantic Constraints for the Automatic Discovery of Part-Whole Relations. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language* (pp. 1 - 8). Morristown: Association for Computational Linguistics.

Godard, D. (2006). Compositionalité: questions linguistiques. Dans *Sémanticlopédie: dictionnaire de sémantique.* GDR Sémantique & Modélisation, CNRS.

Graesser, A. C., VanLehn, K., Rosé, C. P., Jordan, P. W., & Harter, D. (2001). Intelligent Tutoring Systems with Conversational Dialogue. *AI Magazine , 22* (4), 39-52.

Grefenstette, G. (1993). Evaluation techniques for automatic semantic extraction: Comparing syntactic and window-based approaches. *Proceedings of the Workshop on Acquisition of Lexical Knowledge From Texts.* Columbus: SIGLEX/ACL.

Hahn, U., & Romacker, M. (2000). Content management in the SYNDIKATE system - How technical documents are automatically transformed to text knowledge bases. *Data & Knowledge Engineering , 35*, 137-159.

Hammouda, K., & Kamel, S. M. (2005). Data Mining in e-Learning. Dans *E-Learning Networked Environments and Architectures: A Knowledge Processing Perspective* (pp. 374-404). New York: Springer-Verlag.

Harris, Z. S. (1968). *Mathematical Structures of Language.* New York: John Wiley & Sons.

Hayes, P., Eskridge, T., Saavedra, R., Reichherzer, T., Mehrotra, M., & Bobrovnikoff, D. (2005). Collaborative Knowledge Capture in ontologies. *Proceedings of the 3rd International Conference on Knowledge Capture (K-CAP'2005)* (pp. 99-106). New York: ACM Press.

Hearst, M. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. *Proceedings of the Fourteenth International Conference on Computational Linguistics*, (pp. 539–545). Nantes.

Hendler, J. (2001). Agents and the semantic web. *IEEE Intelligent Systems* , 30–36.

IMS. (1997). Consulté le 12 22, 2007, sur Welcome to IMS Global Learning Consortium Inc.: http://www.imsglobal.org/specifications.html

IMS ePortfolio Specification. (2007). Consulté le 9 5, 2007, sur www.imsglobal.org/ep/index.html

IMS Learner Information Package Specification. (2001, 03). Consulté le 12 20, 2007, sur http://www.imsglobal.org/profiles/

IMS-LD. (2007). Consulté le 9 5, 2007, sur http://www.imsglobal.org/learningdesign/index.html

Ingénierie pédagogique et agrégation des objets d'apprentissage. (2007). Consulté le 11 20, 2007, sur LORNET: http://www.lornet.org/Publications/Th%C3%A8me2Ing%C3%A9nieriep%C3%A9dagogiqueetagr%C3%A9gationdes/tabid/325/Default.aspx

Iwanska, L. M., Mata, N., & Kruger, K. (2000). Fully automatic acquisition of taxonomic knowledge from large corpora of texts. In *Natural Language Processing and Knowledge Processing* (pp. 335–345). MIT/AAAI Press.

Java Universal Network/Graph Framework. (2007). Consulté le 11 2007, sur JUNG: http://jung.sourceforge.net/applet/index.html

Jordan, P., Makatchev, M., & VanLehn, K. (2004). Combining Competing Language Understanding Approaches in an Intelligent Tutoring System. *Proceeding of the 7th International Conference on Intelligent Tutoring Systems. LNCS 3220*, pp. 346-357. Maceiò: Springer.

Jovanovic, J., Gasevic, D., & Devedzic, V. (2006a). Dynamic Assembly of Personalized Learning Content on the Semantic Web. Dans *The Semantic Web: Research and Applications* (pp. 545-559). Berlin / Heidelberg: Springer.

Jovanovic, J., Gasevic, D., & Devedzic, V. (2006b). Ontology-based Automatic Annotation of Learning Content. *International Journal on Semantic Web and Information Systems , 2* (2), 91-119.

Jurafsky, D., & Martin, J. H. (2000). *Speech and Language Processing: An introduction to natural language processing,computational linguistics, and speech recognition.* Prentice-Hall.

KAON. (2007). Consulté le 11 12, 2007, sur KAON: http://kaon.semanticweb.org/

Kavalec, M., & Svatek, V. (2005). A Study on Automated Relation Labelling in Ontology Learning. In *Ontology Learning from Text: Methods, Evaluation and Applications* (pp. 44–58). IOS Press.

Keenoy, K., Poulovassilis, A., Christophides, V., Rigaux, P., Papamarkos, G., Magkanaraki, A., et al. (2004). Personalisation Services for Self E-learning Networks. *Proceedings of the 4th International Conference on Web Engineer, (ICWE'04)*, (pp. 215-219). Munich.

Klein, D., & Manning, C. (2003). Accurate Unlexicalized Parsing. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, (pp. 423-430).

Knight, C., Gasevic, D., & Richards, G. (2006). An Ontology-Based Framework for Bridging Learning Design and Learning Content. *Educational Technology & Society , 9* (1), 23-37.

Koohang, A., & Harman, K. (2006). *Learning Objects: Theory, Praxis, Issues, and Trends.* Informing Science.

Koper, R. (2003). Combining Reusable Learing Resources and Services with Pedagogical Purposeful Units of Learning. *Reusing Online Resources: a sustainable approach to E-learning* (pp. 46-59). Kogan Page.

Krdzavac, N., Gasevic, D., & Devedzic, V. (2004). Description Logics Reasoning in Web-based Education Environments. *Proceedings of the Workshop on Adaptive Hypermedia and Collaborative Web-based Systems (at he 4th International Conference on Web Engineering)* (pp. 219-226). Munich: Rinton Press.

Kumar, A. (2006). Using Enhanced Concept Map for Student Modeling in a Model-Based Programming Tutor. *Proceedings of the 19th International FLAIRS Conference on Artificial Intelligence (FLAIRS 2006) Special Track on Intelligent Tutoring Systems* (pp. 527-532). Melbourne Beach: AAAI Press.

Kumar, A., & Kahle, D. J. (2006). VUE: A concept mapping tool for digital content. *Proceedings of the 2nd International Conference on Concept Mapping* (pp. 323-326). San José: University of Costa Rica.

L'allier, J. J. (1997, 04). Consulté le 12 30, 2007, sur http://www.im.com.tr/framerefer.htm

Li, Y., & Huang, R. (2006). Dynamic Composition of Curriculum for Personalized E-Learning. *Proceedings of the 14th International Conference on Computers in Education (ICCE2006)* (pp. 569 - 576). IOS Press.

Lin, D. (1998). Dependency-based evaluation of MINIPAR. *Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation.* Granada.

LOM. (2002, 07 15). *LTSC Home Page.* Consulté le 11 20, 2007, sur http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

Lougheed, P., Bogyo, B., Brokenshire, D., & Kumar, V. (2005). Formalizing Electronic Portfolios in the SPARC ePortfolio Tool. *Proceeding of SW-EL '05: Applications of Semantic Web Technologies for E-learning Workshop at K-CAP '05 International Conference*, (pp. 9-18). Eindhoven.

LT4eL. (2008). Consulté le 02 20, 2008, sur http://www.lt4el.eu/

Maedche, A., & Staab, S. (2000a). Discovering Conceptual Relations from Text. *Proceedings of the 14th European Conference on Artificial Intelligence* (pp. 321-325). Amsterdam: IOS Press.

Maedche, A., & Staab, S. (2000b). Mining Ontologies from Text. *Proceedings of the 12th International Conference EKAW 2000* (pp. 169-189). Berlin / Heidelberg: Springer .

Maedche, A., & Staab, S. (2001). Ontology Learning for the Semantic Web. *IEEE Intelligent Systems , 16* (2), 72-79.

Maedche, A., & Staab, S. (2000c). Semi-Automatic Engineering of Ontologies from Text. *Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering (SEKE'2000)*, (pp. 231-239). Chicago.

Medelyan, O., & Witten, I. H. (2006). Thesaurus Based Automatic Keyphrase Indexing. *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries* (pp. 296-297). New York: ACM .

Meire, M., Ochoa, X., & Duval, E. (2007). SAmgI: Automatic Metadata Generation v2.0. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007* (pp. 1195-1204). Vancouver: AACE.

Memmi, D. (2000). *Le modèle vectoriel pour le traitement des documents.* Consulté le novembre 20, 2007, sur http://www.ieml.org/IMG/pdf/vectoriel.pdf

Minsky, M. (1975). A framework for representing knowledge. *The Psychology of Computer Vision* , 211-277.

Mizoguchi, R. (2004). Le rôle de l'ingénierie ontologique dans le domaine des EIAH. *Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation , 11*, 231-246.

Mizoguchi, R., Hayashi, Y., & Bourdeau, J. (2007). Inside Theory-Aware and Standards-Compliant Authoring System. *Proceedings of The 5th International Workshop on Ontologies and Semantic Web for E-Learning*, (pp. 1-18). Marina del Rey.

Morin, E. (1999). *Extraction de liens sémantiques entre termes à partir de corpus de textes techniques.* Nantes: Université de Nantes (Thèse de doctorat).

Navigli, R., & Velardi, P. (2004). Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. *Computational Linguistics , 30* (2), 151-179.

Ng, A., Hatala, M., & Gasevic, D. (2006). Ontology-based Approach to Learning Objective Formalization. Dans *Competencies in Organizational E-Learning: Concepts and Tools.* Idea Group.

Nkambou, R., Frasson, C., & Gauthier, G. (2003). CREAM-Tools: An Authoring Environment for Knowledge Engineering in Intelligent Tutoring Systems. Dans *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective, adaptative, interactive, and intelligent educational software* (pp. 269-308). Dordrecht: Kluwer Academic Publishers.

Novak, J. D., & Cañas, A. J. (2006). *The Theory Underlying Concept Maps and How to Construct Them.* Florida Institute for Human and Machine Cognition. Pensacola, FI, USA: Florida Institute for Human and Machine Cognition.

OWL Web Ontology Language Overview. (2004, 02 10). Consulté le 12 2007, sur http://www.w3.org/TR/owl-features/

Paquette, G. (2007). An Ontology and a Software Framework for Competency Modeling and Management. *Educational Technology & Society , 10* (3), 1-21.

Paquette, G. (2004). Instructional Engineering for Learning Objects Repositories Networks. *Proceedings of International Conference on Computer Aided Learning in Engineering Education (CALIE 04)*. Grenoble.

Park, Y. (2004). GlossOnt: A Concept-focused Ontology Building Tool. *Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning* (pp. 498-506). Menlo Park, Calif: AAAI Press.

Poesio, M., & Almuhareb, A. (2005). Identifying Concept Attributes Using A Classifier. *Proceedings of the ACL Workshop on Deep Lexical Acquisition* (pp. 18-27). Ann Arbor: Association for Computational Linguistics.

Popescu, A.-M., Yates, A., & Etzioni, O. (2004). Class Extraction from the World Wide Web. *Proceedings of the AAAI-04 Workshop on Adaptive Text Extraction and Mining*, (pp. 68-73). San Jose.

Porter, M. (1980). An algorithm for suffix stripping. *Program; Automated Library and Information Systems , 14* (3), 130-137.

Potter, S. (2001). *A Survey of Knowledge Acquisition from Natural Language*. Consulté le 9 5, 2007, sur http://www.aiai.ed.ac.uk/project/akt/work/stephenp/TMA%20of%20KAfromNL.pdf

Priestley, M. (2001). DITA XML: a reuse by reference architecture for technical documentation. *Proceedings of the 19th annual international conference on Computer documentation* (pp. 152–156). ACM Press.

Protégé. (2007). Consulté le 9 5, 2007, sur http://protege.stanford.edu/

Protégé OWL API. (2007). Consulté le 11 20, 2007, sur http://protege.stanford.edu/plugins/owl/api/

Psyché, V., Bourdeau, J., Nkambou, R., & Mizoguchi, R. (2005). Making Learning Design Standards Work with an Ontology of Educational Theories. *Proceedings of the 12th International Conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology.* (pp. 539-546). Amsterdam, The Netherlands: IOS Press.

Pustejovsky, J. (1995). *The Generative Lexicon.* Cambridge: MIT Press.

Quillian, M. R. (1968). *Semantic memory in semantic information processing.* Cambridge: MIT Press.

RACER. (2007). Consulté le 11 20, 2007, sur http://www.racer-systems.com/site/index.phtml

RDF/XML Syntax Specification. (2004, 02 10). Consulté le 09 05, 2007, sur http://www.w3.org/RDF/

Reinberger, M.-L., & Daelemans, W. (2004). Unsupervised Text Mining for Ontology Extraction: An Evaluation of Statistical Measures. *Proceedings of the 4th International Conference on Language Resources and Evaluation* (pp. 491-494). Paris: ELRA/ELDA.

Reload Editor and Player. (2007). Consulté le 9 5, 2007, sur http://www.reload.ac.uk/ldplayer.html

Reload Project. (2004). Consulté le 10 20, 2007, sur http://www.reload.ac.uk

Resnik, P. (1993). *Selection and Information: A Class-Based Approach to Lexical Relationships.* University of Pennsylvania.

Rinaldi, F., Dowdall, J., Hess, M., Kaljuran, K., Koit, M., Vider, K., et al. (2002). Terminology as Knowledge in Answer Extraction. *Proceedings of the 6th International Conference on Terminology and Knowledge Engineering*, (pp. 107–113). Nancy.

Robison, H. R. (1970). Computer-detectable Semantic Structures. *Information Storage and Retrieval , 6*, 273-288.

Roche, C. (2006a). How words map concepts. *Proceedings of the 10th IEEE on International Enterprise Distributed Object Computing Conference Workshops* (p. 5). Washington: IEEE Computer Society.

Roche, C. (2006b). *L'ontologie–comme–principe–terminologique.* Consulté le 11 20, 2007, sur http://ontology.univsavoie.fr/condillac/fr/activites/publications/Ontologie_principe_terminologique.pdf

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval.* McGraw-Hill.

Sánchez, D., & Moreno, A. (2006). Discovering Non-taxonomic Relations from the Web. *Proceedings of the 7th International Conference on Intelligent Data Engineering and Automated Learning – IDEAL 2006* (pp. 629-636). Berlin / Heidelberg: Springer.

Schank, R. (1972). Conceptual Dependency: A Theory of Natural Language Understanding. *Cognitive Psychology , 3* (4), 532-631.

Schank, R., & Abelson. (1977). *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures.* Hillsdale, NJ: L. Erlbaum.

Schutz, A., & Buitelaar, P. (2005). RelExt: A Tool for Relation Extraction from Text in Ontology Extension. *Proceedings of the 4th International Semantic Web Conference* (pp. 593-606). Springer .

SCORM. (2007). Consulté le 9 5, 2007, sur Sharable Content Object Reference Model: http://www.adlnet.gov/scorm/index.cfm

Searle, J. (1975). Indirect speech acts. *Syntax and Semantics* , 59–82.

SemCor. (2002). Consulté le 12 28, 2007, sur http://www.cs.unt.edu/~rada/downloads.html#semcor

Shamsfard, M., & Barforoush, A. A. (2003). The State of the Art in Ontology Learning:A Framework for Comparison. *The Knowledge Engineering Review , 18* (4), 293-316.

Shasha, D., Wang, J. T., & Giugno, R. (2002). Algorithmics and Applications of Tree and Graph Searching. *Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (pp. 39-52). ACM Press.

Shen, D., Zhang, J., Zhou, G., Su, J., & Tan, C.-L. (2003). Effective Adaptation of Hidden Markov Model-based Named Entity Recognizer for Biomedical Domain. *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine* (pp. 49–56). Morristown: Association for Computational Linguistics.

Siemer, J., & Angelides, M. C. (1998). A comprehensive method for the evaluation of complete intelligent tutoring systems. *Decision Support Systems* , 85-102.

Sipser, M. (2005). Regular Languages. Dans *Introduction to the Theory of Computation* (pp. 31–90). Boston: PWS Publishing.

Sleator, D., & Temperley, D. (1993). Parsing English with a Link Grammar. *Proceedings of the 3rd International Workshop on Parsing Technologies.* Tilburg, the Netherlands.

Soderland, S., & Mandhani, B. (2007). Moving from Textual Relations to Ontologized Relations. *Proceedings of the AAAI 2007 Spring Symposium Series on Machine Reading* (pp. 85-90). Menlo Park, California: AAAI Press.

Sosnovsky, S., Dolog, P., Henze, N., Brusilovsky, P., & Nejdl, W. (2007). Translation of Overlay Models of Student Knowledge for Relative Domains Based on Domain Ontology Mapping. *Proceedings of the 13th International Conference on Artificial Intelligence in Education.* IOS Press.

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine.* Reading, MA: Addison-Wesley.

Srikant, R., & Agrawal, R. (1997). Mining Generalized Association Rules. *Future Generation Computer Systems , 13* (2-3), 161-180.

Stanford. (2007). *EnglishGrammaticalRelations (Stanford JavaNLP API)*. Consulté le 11 20, 2007, sur

http://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/trees/EnglishGrammaticalRelations.html

Stevenson, M., & Greenwood, M. A. (2006). Comparing Information Extraction Pattern Models. *Proceedings of the Workshop "Information Extraction Beyond The Document" held in conjunction with 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics* (pp. 12–19). Australia: Association for Computational Linguistics.

Stojanovic, L., Staab, S., & Studer, R. (2001). eLearning based on the Semantic Web. *Proceedings of WebNet2001 - World Conference on the WWW and Internet.* Orlando.

Suraweera, P., Mitrovic, A., & Martin, B. (2004). The Use of Ontologies in ITS Domain Knowledge Authoring. *Proceedings of the 2nd International Workshop on Applications of Semantic Web for E-learning held at the ITS 2004 Conference* (pp. 41-49). Maceio: University of Canterbury.

Tesnière, L. (1959). *Elements de syntaxe structurale.* Paris: Klincksieck.

Text-To-Onto. (2007). Consulté le 6 1, 2007, sur http://sourceforge.net/projects/texttoonto

The Stanford NLP Group. (2007). Consulté le 10 10, 2007, sur http://nlp.stanford.edu/software/lex-parser.shtml

Turney, P. D. (2000). Learning algorithms for keyphrase extraction. *Information Retrieval , 2* (4), 303–336.

Turney, P. D. (2001). Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. *Proceedings of the 12th European Conference on Machine Learning* (pp. 491-502). London: Springer-Verlag.

Tuso, G., & Longmire, W. (2000). Competency-Based Systems and the Delivery of Learning Content. In *Learning Without Limits* (pp. 33-38). Informania Inc.

UIMA Java Framework. (2007). Consulté le 10 10, 2007, sur UIMA Java Framework: http://uima-framework.sourceforge.net/

Ullrich, C. (2004). Description of an instructional ontology and its application in web services for education. *Proceedings of the Workshop on Applications of Semantic Web Technologies for E-learning*, (pp. 17-23). Hiroshima.

Ullrich, C. (2005). The learning-resource-type is dead, long live the learning- resource-type! *Learning Objects and Learning Designs , 1* (1), 7-15.

Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., et al. (2006). Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web , 4*, 14–28.

Valerio, A., & Leake, D. (2006). Jump-Starting Concept Map Construction With Knowledge Extracted from Documents. *Proceedings of the 2nd International Conference on Concept Mapping* (pp. 296-303). San José: Universidad de Costa Rica.

Van Elst, L., & Abecker, A. (2002). Domain Ontology Agents for Distributed Organizational Memories. Dans *Knowledge Management and Organizational Memories* (pp. 147-158). Kluwer Academic Publishers.

Van Hage, W. R., Kolb, H., & Schreiber, G. (2006). A Method for Learning Part-Whole Relations. *Proceedings of the 5th International Semantic Web Conference* (pp. 723-735). Berlin / Heidelberg: Springer.

Verbert, K., & Duval, E. (2004). Towards a Global Component Architecture for Learning Objects: A Comparative Analysis of Learning Object Content Models. *Proceedings of the ED-MEDIA 2004 World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 202-208). AACE.

Verbert, K., Jovanovic, J., Duval, E., & Gaševic, D. (2006). Ontology-based Learning Content Repurposing: the ALOCoM Framework. *International Journal on E-Learning , 5* (1), 67-74.

Verbert, K., Klerkx, J., Meire, M., Najjar, J., & Duval, E. (2004). Towards a Global Component Architecture for Learning Objects: An Ontology Based Approach. *Proceedings of OTM Workshops 2004* (pp. 713–722). Berlin: Springer-Verlag.

Véronis, J. (2004). HyperLex: lexical cartography for information retrieval. *Computer speech and language , 18* (3), 223-252.

Vintar, S., Buitelaar, P., Ripplinger, B., Sacaleanu, B., Raileanu, D., & Prescher, D. (2002). An Efficient and Flexible Format for Linguistic and Semantic Annotation. *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, (pp. 1658-1662). Las Palmas.

Volker, J., Hitzler, P., & Cimiano, P. (2007). Acquisition of OWL DL Axioms from Lexical Resources. *Proceedings of the 4th European Semantic Web Conference (ESWC'07)* (pp. 670-685). Berlin: Springer-Verlag.

Volker, J., Vrandecic, D., Sure, Y., & Hotho, A. (2007). Learning Disjointness. *Proceedings of the 4th European Semantic Web Conference (ESWC'07)* (pp. 175-189). Berlin: Springer-Verlag.

Voorhees, R. A. (2001). *Competency-Based Learning Models: A Necessary Future.* New York: John Wiley & Sons, Inc.

Wagner, E. D. (2002). Steps to Creating a Content Strategy for Your Organization. *The e-Learning Developers' Journal* .

Wiley, D. A. (2000). *Connecting Learning Objects to Instructional Design Theory: A Definition,a Metaphor, and a Taxonomy.* Consulté le 11 20, 2007, sur http://www.elearning-reviews.org/topics/technology/learning-objects/2001-wiley-learning-objects-instructional-design-theory.pdf

Winograd, T. (1972). Understanding Natural Language. *Cognitive Psychology , 3* (1), 1-191.

Winston, M. E., Chaffin, R., & Herrmann, D. (1987). A Taxonomy of Part-Whole Relations. *Cognitive Science , 11*, 417-444.

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. (2005). Kea: Practical automatic keyphrase extraction. In *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific* (pp. 129-152). London: Information Science Publishing.

Yamada, I., & Baldwin, T. (2004). Automatic discovery of telic and agentive roles from corpus data. *Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation (PACLIC)*, (pp. 115–126). Tokyo.

Zavitsanos, E., Paliouras, G., & Vouros, G. (2006). *Ontology Learning and Evaluation: A survey.* Consulté le 11 10, 2007, sur http://www.ontosum.org/static/Publications

Zouaq, A., Nkambou, R., & Frasson, C. (2007a). A Framework for the Capitalization of e-Learning Resources. *Proceedings of ED-MEDIA--World Conference on Educational Multimedia, Hypermedia & Telecommunications (Ed-media 2007)* (pp. 1241-1247). AACE.

Zouaq, A., Nkambou, R., & Frasson, C. (2007b). An Integrated Approach for Automatic Aggregation of Learning Knowledge Objects. *Interdisciplinary Journal of Knowledge and Learning Objects (IJKLO) , 3*, 135-162.

Zouaq, A., Nkambou, R., & Frasson, C. (2006b). An Ontology-Based Solution for Knowledge Management and eLearning Integration. *Proceedings of the 8th International Conference on Intelligent Tutoring System (ITS). LNCS 4053*, pp. 716-718. Berlin: Springer-Verlag.

Zouaq, A., Nkambou, R., & Frasson, C. (2007c). Building Domain Ontologies from Text for Educational Purposes. *Proceedings of the 2nd European Conference on Technology-enhanced Learning* (pp. 393-407). Berlin: Springer-Verlag.

Zouaq, A., Nkambou, R., & Frasson, C. (2007d). Document Semantic Annotation for Intelligent Tutoring Systems: a Concept Mapping Approach. *Proceedings of the 20th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2007)* (pp. 380-385). AAAI Press.

Zouaq, A., Nkambou, R., & Frasson, C. (2006a). The Knowledge Puzzle: An Integrated Approach of Intelligent Tutoring Systems and Knowledge Management. *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2006)* (pp. 575-582). IEEE Computer Society Press.

Zouaq, A., Nkambou, R., & Frasson, C. (2007e). Towards Learning Knowledge Objects. *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AIED 2007)* (pp. 674-676). IOS Press.

Zouaq, A., Nkambou, R., & Frasson, C. (2007f). Une architecture d'acquisition et d'exploitation des connaissances pour les EIAH. *Proceedings of the third International Conference on « Environnement Informatique pour l'Apprentissage Humain » (EIAH 2007)*, (pp. 131-136). Lausanne.

Zouaq, A., Nkambou, R., & Frasson, C. (2007g). Using a Competence Model to Aggregate Learning Knowledge Objects. *Proceedings of the 7th IEEE International Conference on Advanced Learning Technologies (ICALT 2007)* (pp. 836-840). IEEE Computer Society Press.

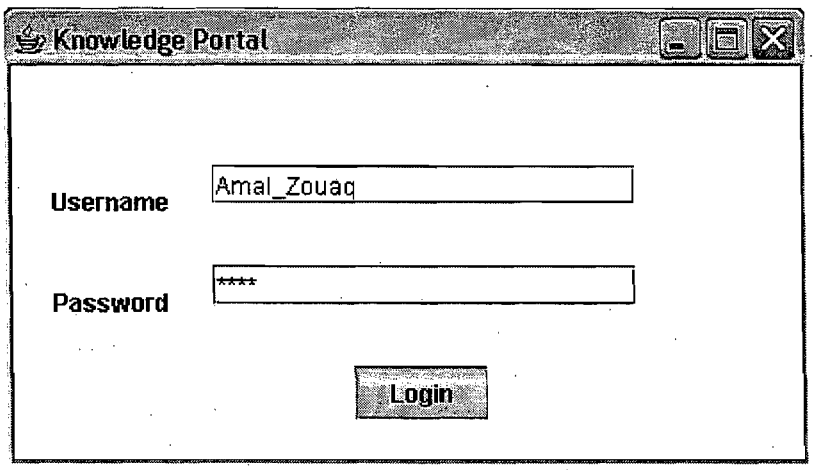
Annexe A : Manuel d'utilisation de TEXCOMON

Il existe quelques desideratas sur les données en entrées et quelques remarques à formuler :

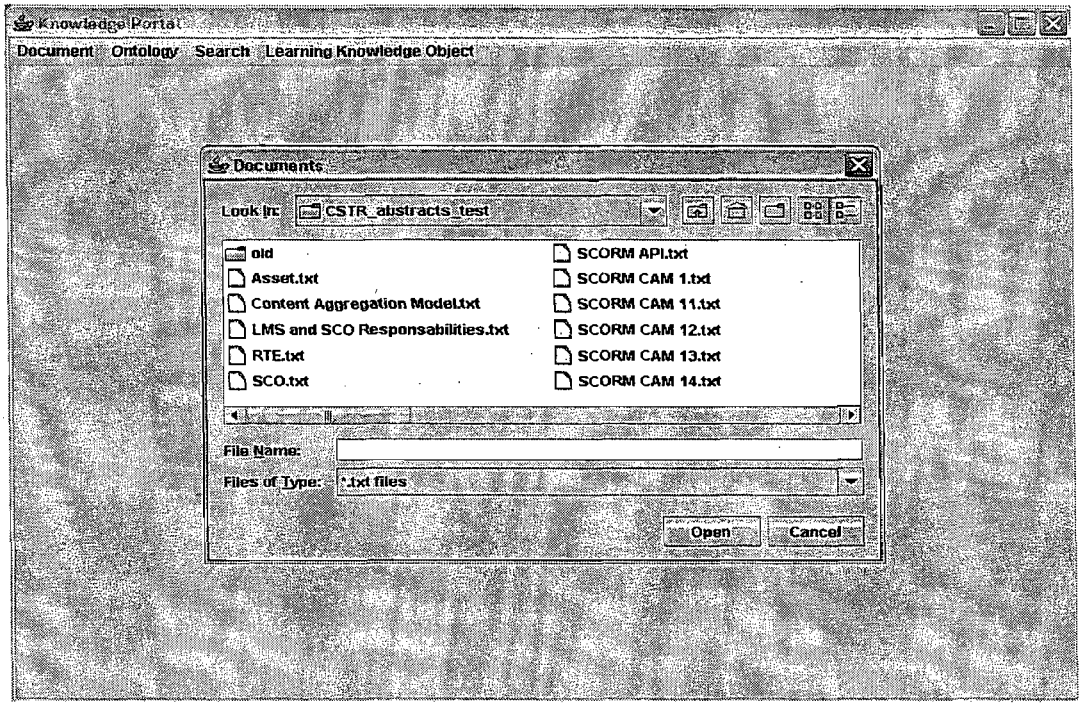
1. Pour le moment, les documents en entrée doivent être de type txt.
2. Les différents paragraphes doivent être séparés par une ligne blanche.
3. Nous travaillons actuellement avec la version 1.5 de l'analyseur de Stanford.
4. Nous avons noté un problème avec les mots contenant un tiret dans le module de traitement des dépendances de l'Université de Stanford. Par exemple : dans « *web-based content* », le « *-based* » ne sera pas présent dans la carte grammaticale de concepts. Nous avons reporté ce problème aux développeurs de Stanford, qui nous ont confirmé que cela était un bug et qu'il était réparé dans la dernière version de l'analyseur (version 1.6) (Toutefois nous ne l'avons pas vérifié).

Voici les étapes à suivre pour générer une ontologie du domaine dans TEXCOMON :

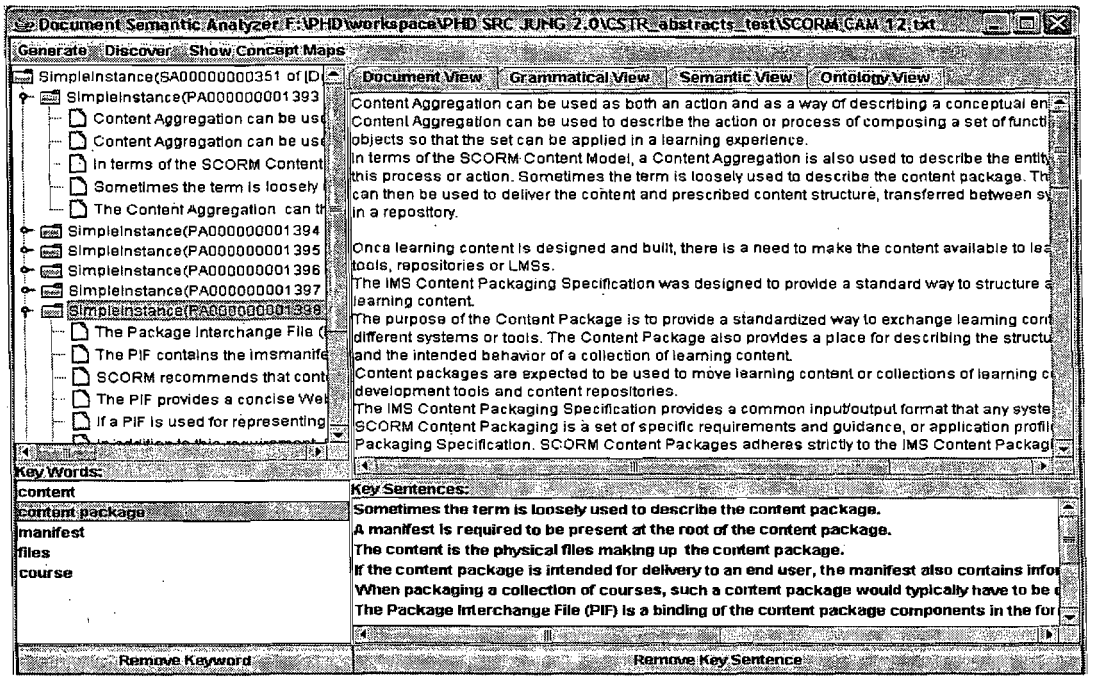
1. S'identifier en tant qu'usager du système



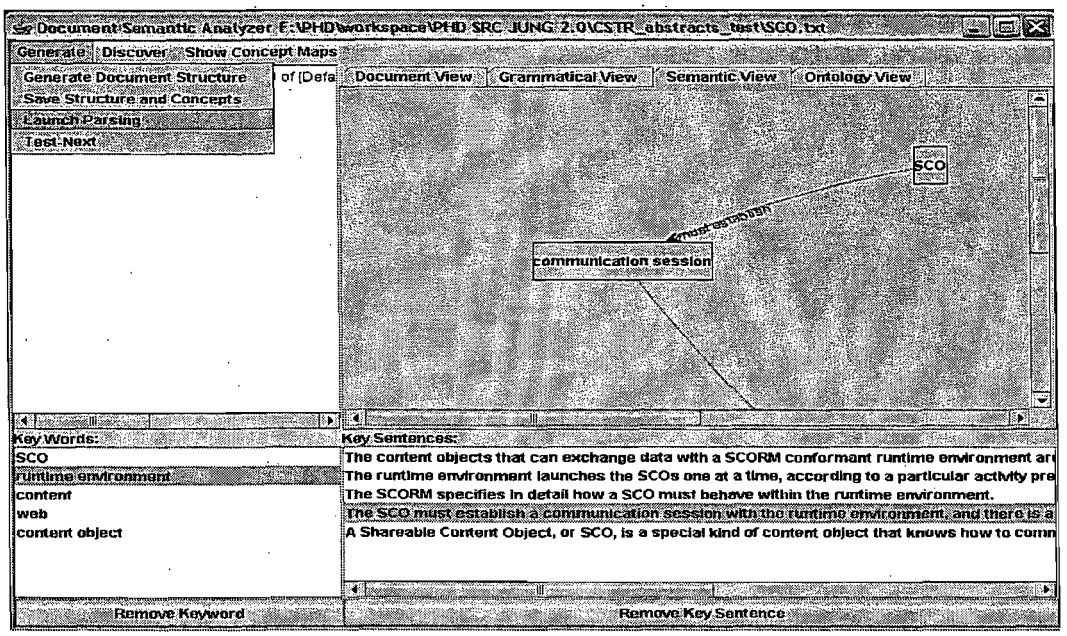
2. Créer un corpus de documents txt concernant un même domaine ;



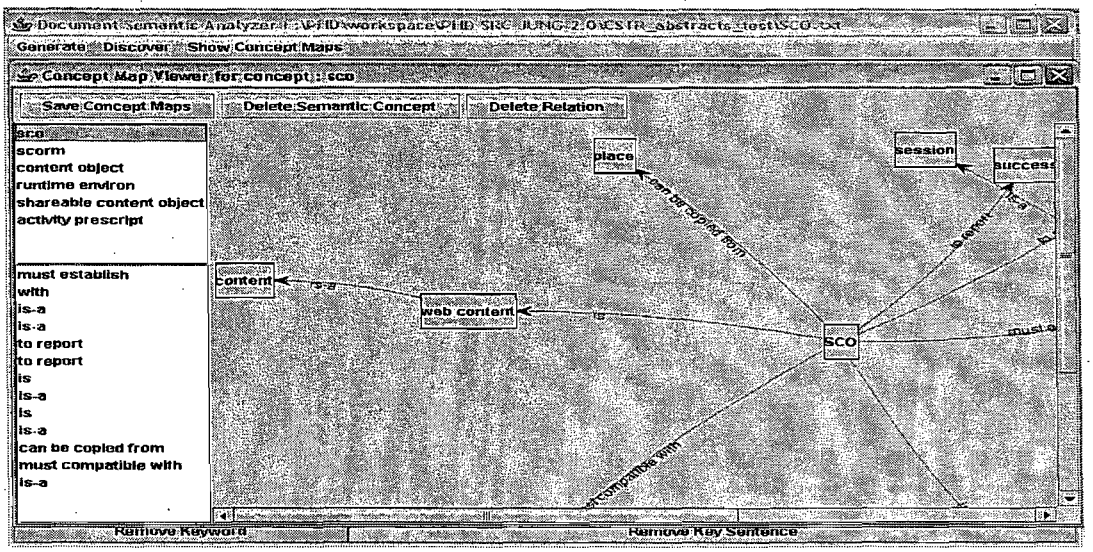
3. Pour chaque document, déterminer la structure du document, ses mots-clés, ses phrases-clés en exécutant l'outil « *Knowledge Extractor* ». Ces mots-clés et phrases-clés peuvent être mis à jour (suppression, ajout) par un opérateur humain.



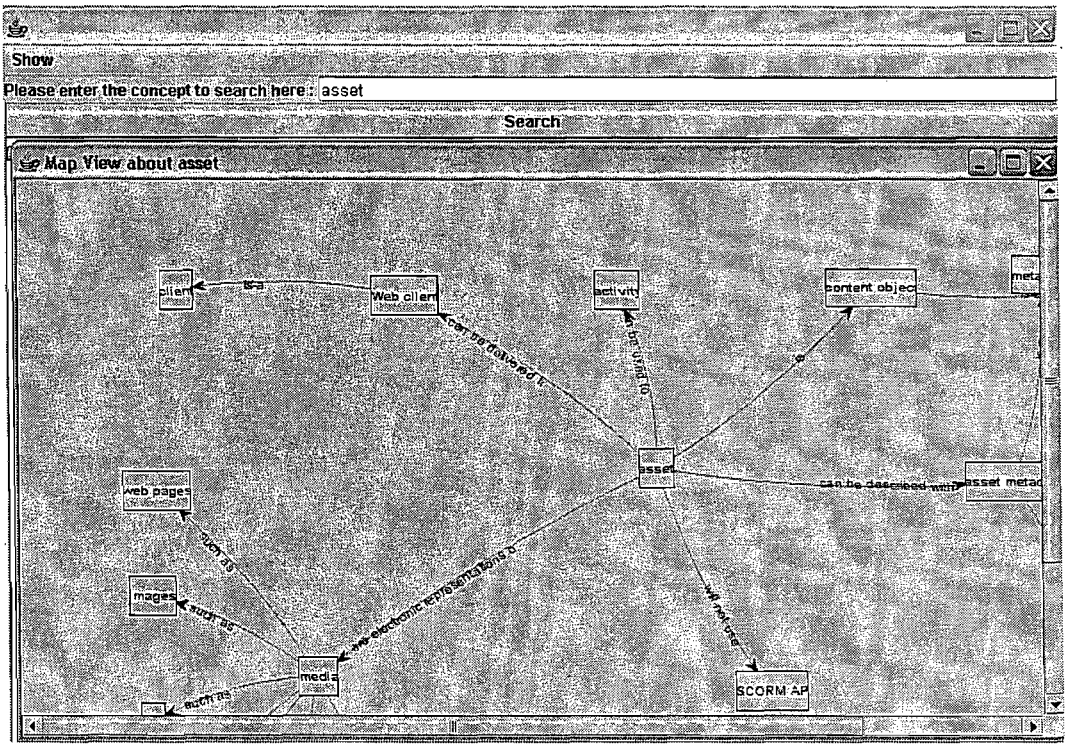
4. Toujours dans le même outil, cliquer sur le menu « *Generate* » pour demander l'analyse des phrases-clés. Cette fonctionnalité fait appel à l'analyseur de Stanford et à son module de dépendances typées, puis à la fonction de recherche des patrons lexico-syntaxiques et à leur transformation en représentations sémantiques.



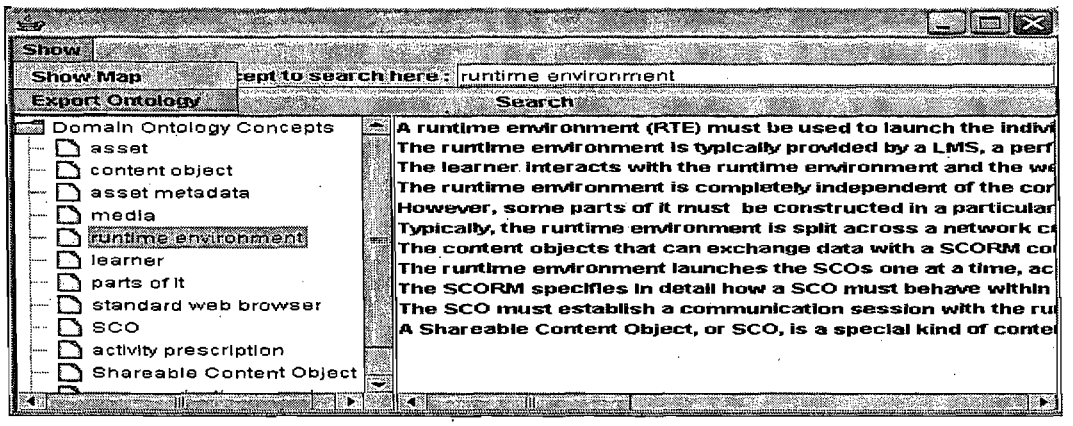
5. Une fois l'analyse du document terminée, il est possible de visualiser les cartes de concepts générées (menu « *Show Concept Maps* »). Ces dernières sont regroupées autour des concepts les plus importants du document. Il est possible de les mettre à jour avant de les sauvegarder.



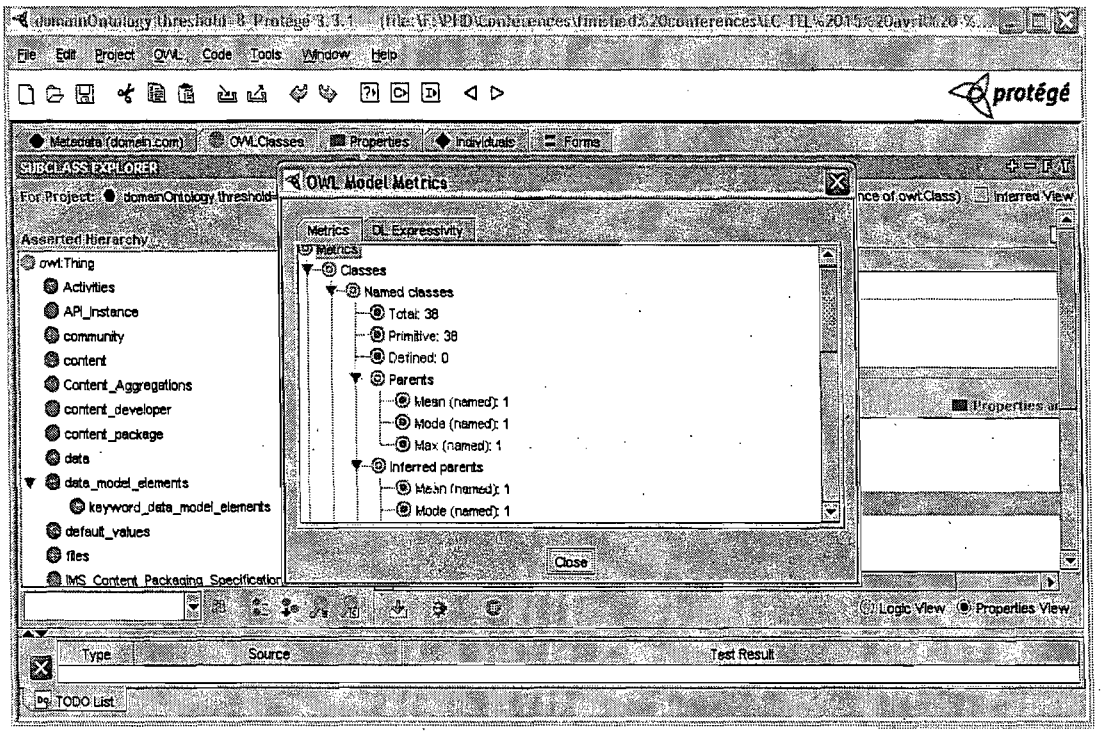
6. Recommencer le processus pour tous les documents du corpus.
7. Une fois la phase d'analyse complétée, il est possible d'effectuer des recherches sur les concepts générés et de demander l'affichage de leurs cartes de concepts. C'est à ce niveau que l'on fixe le paramètre « *out-degree* » qui permet de déterminer quels concepts et relations se retrouveront effectivement dans l'ontologie du domaine.



8. Cliquer sur le menu *Show* et sur le sous-menu de génération de l'ontologie du domaine (*Export Ontology*) ;



9. Ouvrir l'ontologie générée dans un éditeur d'ontologie, comme Protégé par exemple. Il est possible alors de l'explorer, de connaître ses métriques (nombre de classes, de propriétés, etc.) et son expressivité.



10. On peut faire varier le paramètre « *out-degree* », comparer les ontologies résultant de cette variation et conserver celle qui nous convient le plus.

Annexe B : L'environnement Protégé

Nous avons utilisé l'environnement Protégé (Protégé, 2007) pour l'édition et la mise à jour de l'architecture ontologique de la mémoire organisationnelle. Protégé est un éditeur qui permet de construire des ontologies et leur adjoint une interface graphique qui permet de créer des instances de classes. Divers Plugins enrichissent l'environnement Protégé et permettent de manipuler des règles (SWRLTab), d'afficher l'ontologie sous forme graphique (OntoViz, OWLViz, TGVizTab), d'intégrer des moteurs de règles (JessTab, Algernon) et autres fonctionnalités.

Protégé offre également un service de validation du langage utilisé par l'ontologie et permet de déterminer si cette dernière est exprimée en OWL-Lite, OWL-DL ou OWL-Full. Dans notre cas, nous générons des ontologies du domaine en OWL-DL ou en OWL-Lite en fonction du corpus dont nous disposons (Figure 51).

Protégé dispose d'une librairie Java « *Protégé-OWL API* » qui permet de convertir le schéma ontologique en objets java. Nous l'avons utilisé pour la manipulation et les requêtes sur les ontologies à partir du projet « *The Knowledge Puzzle* ». Cette API fournit des méthodes pour charger des modèles OWL, formuler des requêtes sur ces modèles et inférer de nouvelles connaissances en effectuant des mécanismes de raisonnement. En effet, il est possible de raisonner avec des ontologies en utilisant un moteur d'inférence. Nous avons préalablement indiqué qu'il est possible d'utiliser Jess (Friedman-Hill, 2003) ou Algernon (Algernon, 2007) pour des tâches de raisonnement. Par exemple, Jess nous a permis d'exécuter les règles SWRL définies en conjonction avec l'ontologie des compétences et l'ontologie des théories pédagogiques.

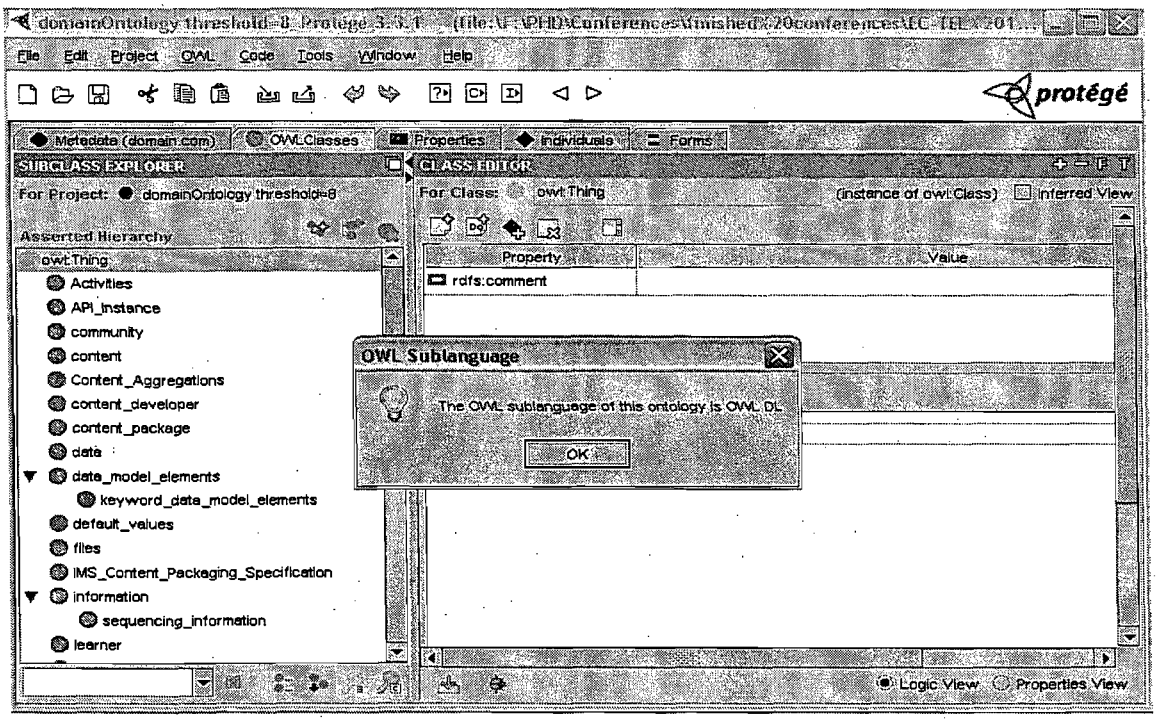


Figure 51. Langage de l'ontologie générée

Il existe aussi des moteurs d'inférence basés sur les logiques de description notamment RACERPRO (RACER, 2007). Bien que commercial, il est possible d'obtenir une licence à des fins de recherche en université. C'est d'ailleurs au travers de RACERPRO qu'il nous a été possible :

- De tester **la consistance** des ontologies générées : le moteur de raisonnement se base sur la description des classes (leurs conditions) pour déterminer s'il leur est possible de posséder des instances. Dans le cas contraire, les classes sont dites inconsistantes. Il est ainsi possible de tester la consistance d'une ontologie dans son ensemble (Figure 52. Interface de test de la consistance de l'ontologie générée) ou en sélectionnant un concept particulier (Figure 53).

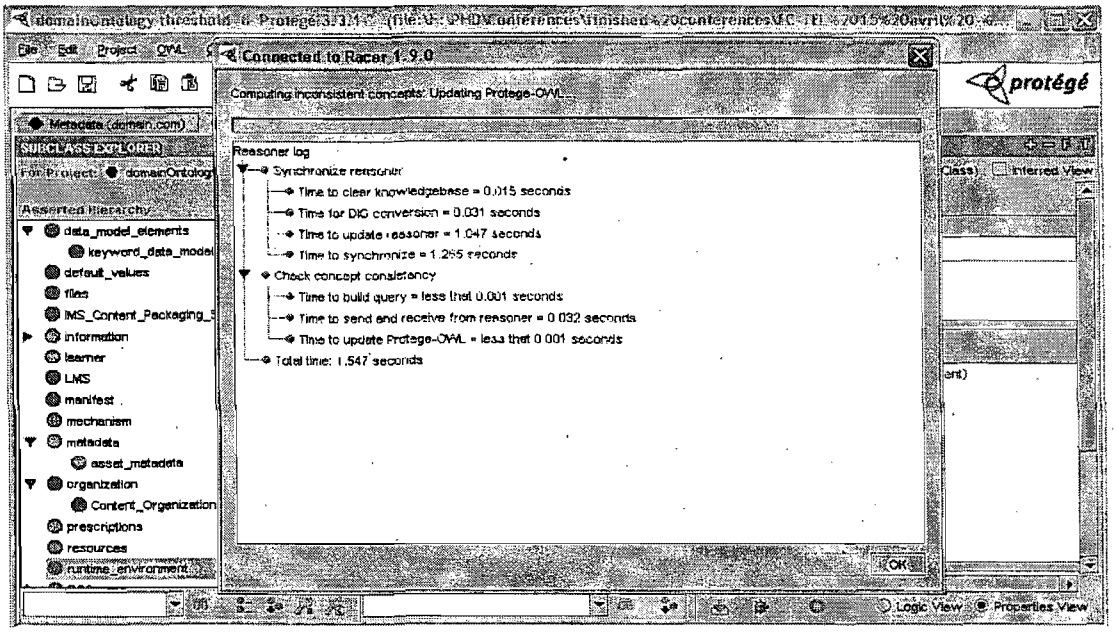


Figure 52. Interface de test de la consistance de l'ontologie générée

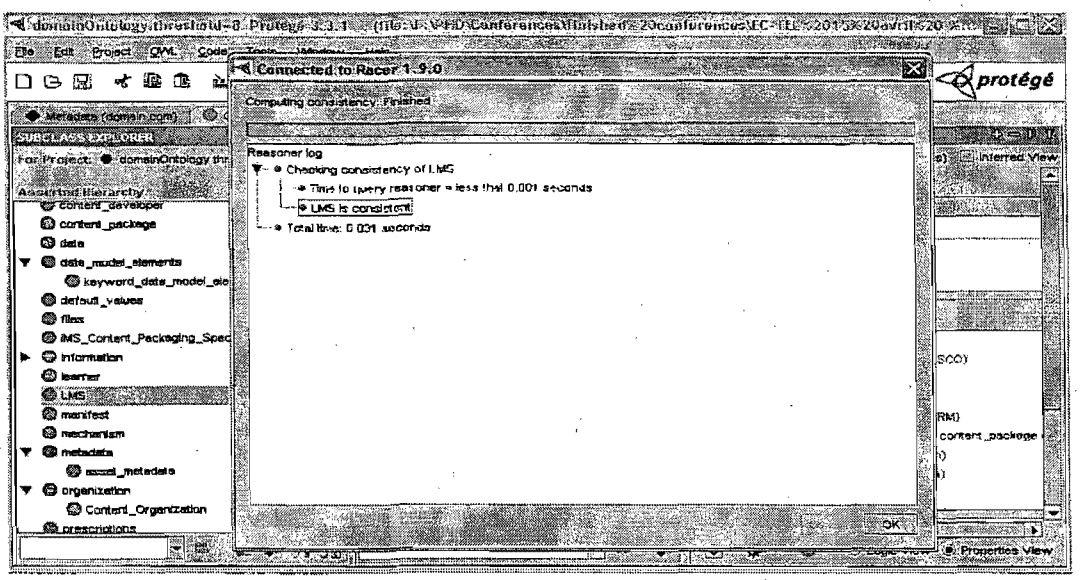


Figure 53. Test de la consistance du concept LMS

- De **classifier la hiérarchie des classes** (la taxonomie), c'est-à-dire d'inférer une nouvelle hiérarchie (s'il y a lieu) à partir des définitions de classes. Lorsqu'une classe est re-classifiée, c'est-à-dire que ses superclasses ont changé, cela est

clairement indiqué dans la hiérarchie inférée. Pareillement, lorsqu'une classe est jugée inconsistante, elle est entourée d'un rond rouge. Notons que le moteur d'inférence doit avoir des **classes définies ou complètes** (qui ont des conditions nécessaires et suffisantes) pour pouvoir inférer une nouvelle hiérarchie (Figure 54).

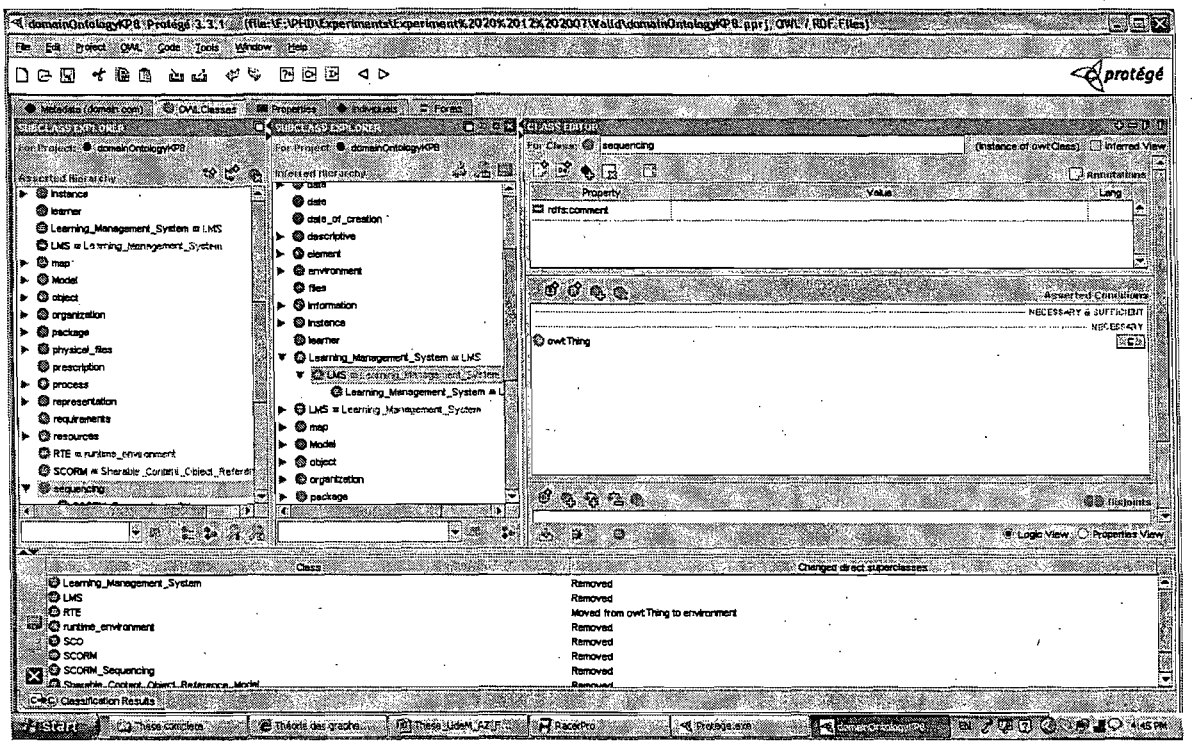


Figure 54. Classification de l'ontologie générée à l'aide de RacerPro

Par exemple, RacerPro a déplacé la classe *RTE* et l'a classifié comme sous-classe de la classe « *Environment* ».

Notons enfin que Protégé offre la possibilité d'utiliser les interfaces graphiques Protégé des classes au sein même de l'application, épargnant ainsi au programmeur d'avoir à les recréer. On dispose donc sans effort d'interfaces d'édition des différentes classes et de leurs instances, comme par exemple un éditeur de compétences.

Annexe C : Explication des différentes relations grammaticales

La documentation fournie par l'université de Stanford contient l'explication des différentes catégories grammaticales. Les relations suivantes sont extraites intégralement de la documentation Java de (Stanford, 2007):

PREDICATE

`public static final GrammaticalRelation` **`PREDICATE`**

The "predicate" grammatical relation. The predicate of a clause is the main VP of that clause; the predicate of a subject is the predicate of the clause to which the subject belongs.

Example:
"Reagan died" → `pred`(Reagan, died)

AUX_MODIFIER

`public static final GrammaticalRelation` **`AUX_MODIFIER`**

The "auxiliary" grammatical relation. An auxiliary of a clause is a non-main verb of the clause.

Example:
"Reagan has died" → `aux`(died, has)

AUX_PASSIVE_MODIFIER

`public static final GrammaticalRelation` **`AUX_PASSIVE_MODIFIER`**

The "passive auxiliary" grammatical relation. A passive auxiliary of a clause is a non-main verb of the clause which contains the passive information.

Example:
"Kennedy has been killed" → auxpass(killed, been)

COPULA

public static final GrammaticalRelation **COPULA**

The "copula" grammatical relation. A copula is the relation between the complement of a copular verb and the copular verb.

Examples:
"Bill is big" → cop(big, is)
"Bill is an honest man" → cop(man, is)

CONJUNCT

public static final GrammaticalRelation **CONJUNCT**

The "conjunct" grammatical relation. A conjunct is the relation between two elements connected by a conjunction word.

Example:
"Bill is big and honest" → conj(big, honest)

COORDINATION

public static final GrammaticalRelation **COORDINATION**

The "coordination" grammatical relation. A coordination is the relation between an element and a conjunction.

Example:
"Bill is big and honest." → cc(big, and)

PUNCTUATION

`public static final GrammaticalRelation` **PUNCTUATION**

The "punctuation" grammatical relation. This is used for any piece of punctuation in a clause, if punctuation is being retained in the typed dependencies.

Example:
"Go home!" → `punct`(Go, !)

ARGUMENT

`public static final GrammaticalRelation` **ARGUMENT**

The "argument" grammatical relation. An argument of a VP is a subject or complement of that VP; an argument of a clause is an argument of the VP which is the predicate of that clause.

Example:
"Clinton defeated Dole" → `arg`(defeated, Clinton), `arg`(defeated, Dole)

SUBJECT

`public static final GrammaticalRelation` **SUBJECT**

The "subject" grammatical relation. The subject of a VP is the noun or clause that performs or experiences the VP; the subject of a clause is the subject of the VP which is the predicate of that clause.

Examples:
"Clinton defeated Dole" → `subj`(defeated, Clinton)
"What she said is untrue" → `subj`(is, What she said)

NOMINAL_SUBJECT

`public static final GrammaticalRelation` **NOMINAL_SUBJECT**

The "nominal subject" grammatical relation. A nominal subject is a subject which is an noun phrase.

Example:
"Clinton defeated Dole" → `nsubj`(defeated, Clinton)

NOMINAL_PASSIVE_SUBJECT

`public static final GrammaticalRelation` **`NOMINAL_PASSIVE_SUBJECT`**

The "nominal passive subject" grammatical relation. A nominal passive subject is a subject of a passive which is an noun phrase.

Example:
"Dole was defeated by Clinton" → `nsubjpass`(defeated, Dole)

CLAUSAL_SUBJECT

`public static final GrammaticalRelation` **`CLAUSAL_SUBJECT`**

The "clausal subject" grammatical relation. A clausal subject is a subject which is a clause.

Examples: (subject is "what she said" in both examples)
"What she said makes sense" → `csubj`(makes, said)
"What she said is untrue" → `csubj`(untrue, said)

CLAUSAL_PASSIVE_SUBJECT

`public static final GrammaticalRelation` **`CLAUSAL_PASSIVE_SUBJECT`**

The "clausal passive subject" grammatical relation. A clausal passive subject is a subject of a passive verb which is a clause.

Example: (subject is "that she lied")
"That she lied was suspected by everyone" → `csubjpass`(suspected, lied)

COMPLEMENT

`public static final GrammaticalRelation COMPLEMENT`

The "complement" grammatical relation. A complement of a VP is any object (direct or indirect) of that VP, or a clause or adjectival phrase which functions like an object; a complement of a clause is an complement of the VP which is the predicate of that clause.

Examples:
"She gave me a raise" → comp(gave, me), comp(gave, a raise)
"I like to swim" → comp(like, to swim)

OBJECT

`public static final GrammaticalRelation OBJECT`

The "object" grammatical relation. An object of a VP is any direct object or indirect object of that VP; an object of a clause is an object of the VP which is the predicate of that clause.

Examples:
"She gave me a raise" → obj(gave, me), obj(gave, raise)

DIRECT_OBJECT

`public static final GrammaticalRelation DIRECT_OBJECT`

The "direct object" grammatical relation. The direct object of a VP is the noun phrase which is the (accusative) object of the verb; the direct object of a clause is the direct object of the VP which is the predicate of that clause.

Example:
"She gave me a raise" → dobj(gave, raise)

INDIRECT_OBJECT

`public static final GrammaticalRelation INDIRECT_OBJECT`

The "indirect object" grammatical relation. The indirect object of a VP is the noun phrase which is the (dative) object of the verb; the indirect object of a clause is the indirect object of the VP which is the predicate of that clause.

Example:
"She gave me a raise" → `iobj`(gave, me)

PREPOSITIONAL_OBJECT

`public static final GrammaticalRelation PREPOSITIONAL_OBJECT`

The "prepositional object" grammatical relation. The object of a preposition is the head of a noun phrase following the preposition. (The preposition in turn may be modifying a noun, verb, etc.) We here define cases of VBG quasi-prepositions like "including", "concerning", etc. as instances of pobj.

Example:
"I sat on the chair" → `pobj`(on, chair)

PREPOSITIONAL_COMPLEMENT

`public static final GrammaticalRelation PREPOSITIONAL_COMPLEMENT`

The "prepositional complement" grammatical relation. The prepositional complement of a preposition is the head of a sentence following the preposition.

Examples:
"We have no useful information on whether users are at risk" &arr; `pcomp`(on, are)
"They heard about you missing classes." &arr; `pcomp`(about, missing)

ATTRIBUTIVE

`public static final GrammaticalRelation` **ATTRIBUTIVE**

The "attributive" grammatical relation. The attributive is the complement of a verb such as "to be, to seem, to appear".

CLAUSAL_COMPLEMENT

`public static final GrammaticalRelation` **CLAUSAL_COMPLEMENT**

The "clausal complement" grammatical relation. A clausal complement of a VP or an ADJP is a clause with internal subject which functions like an object of the verb or of the adjective; a clausal complement of a clause is the clausal complement of the VP or of the ADJP which is the predicate of that clause. Such clausal complements are usually finite (though there are occasional remnant English subjunctives).

Example:
"He says that you like to swim" → `ccomp`(says, like)
"I am certain that he did it" → `ccomp`(certain, did)

XCLAUSAL_COMPLEMENT

`public static final GrammaticalRelation` **XCLAUSAL_COMPLEMENT**

The "xclausal complement" grammatical relation. An xcomp complement of a VP or an ADJP is a clausal complement with an external subject. These xcomps are always non-finite. (Only "TO-clause" are recognized.)

Examples:
"I like to swim" → `xcomp`(like, swim)
"I am ready to leave" → `xcomp`(ready, leave)

COMPLEMENTIZER

`public static final GrammaticalRelation` **COMPLEMENTIZER**

The "complementizer" grammatical relation. A complementizer of a clausal complement is the word introducing it.

Example:
"He says that you like to swim" → `complm`(like, that)

MARKER

`public static final GrammaticalRelation` **`MARKER`**

The "marker" grammatical relation. A marker of an adverbial clausal complement is the word introducing it.

Example:
"U.S. forces have been engaged in intense fighting after insurgents launched simultaneous attacks" → `mark`(launched, after)

RELATIVE

`public static final GrammaticalRelation` **`RELATIVE`**

The "relative" grammatical relation. A relative of a relative clause is the head word of the WH-phrase introducing it.

Examples:
"I saw the man you love" → `rel`(love, that)
"I saw the man whose wife you love" → `rel`(love, wife)

REFERENT

`public static final GrammaticalRelation` **`REFERENT`**

The "referent" grammatical relation. A referent of NP is a relative word introducing a relative clause modifying the NP.

Example:
"I saw the book which you bought" → `ref`(book, which)

EXPLETIVE

`public static final GrammaticalRelation` **EXPLETIVE**

The "expletive" grammatical relation. This relation captures an existential there.

Example:
"There is a statue in the corner" → `expl`(is, there)

ADJECTIVAL_COMPLEMENT

`public static final GrammaticalRelation` **ADJECTIVAL_COMPLEMENT**

The "adjectival complement" grammatical relation. An adjectival complement of a VP is a adjectival phrase which functions like an object of the verb; an adjectival complement of a clause is the adjectival complement of the VP which is the predicate of that clause.

Example:
"She looks very beautiful" → `acomp`(looks, very beautiful)

MODIFIER

`public static final GrammaticalRelation` **MODIFIER**

The "modifier" grammatical relation. A modifier of a VP is any constituent that serves to modify the meaning of the VP (but is not an ARGUMENT of that VP); a modifier of a clause is an modifier of the VP which is the predicate of that clause.

Examples:
"Last night, I swam in the pool" → `mod`(swam, in the pool), `mod`(swam, last night)

ADV_CLAUSE_MODIFIER

`public static final GrammaticalRelation` **ADV_CLAUSE_MODIFIER**

The "adverbial clause modifier" grammatical relation. An adverbial clause modifier of a VP is a clause modifying the verb (temporal clauses, consequences, conditional clauses, etc.)

Examples:
"The accident happened as the night was falling" → `advcl`(happened, falling)
"If you know who did it, you should tell the teacher" → `advcl`(tell, know)

PURPOSE_CLAUSE_MODIFIER

`public static final GrammaticalRelation PURPOSE_CLAUSE_MODIFIER`

The "purpose clause modifier" grammatical relation. A purpose clause modifier of a VP is a clause headed by "(in order) to" specifying a purpose. Note: at present we only recognize ones that have "in order to" as otherwise we can't give our surface representations distinguish these from xcomp's. We can also recognize "to" clauses introduced by "be VBN".

Example:
"He talked to the president in order to secure the account" → `purpcl`(talked, secure)

TEMPORAL_MODIFIER

`public static final GrammaticalRelation TEMPORAL_MODIFIER`

The "temporal modifier" grammatical relation. A temporal modifier of a VP or an ADJP is any constituent that serves to modify the meaning of the VP or the ADJP by specifying a time; a temporal modifier of a clause is an temporal modifier of the VP which is the predicate of that clause.

Example:
"Last night, I swam in the pool" → `tmod`(swam, night)

RELATIVE_CLAUSE_MODIFIER

`public static final GrammaticalRelation RELATIVE_CLAUSE_MODIFIER`

The "relative clause modifier" grammatical relation. A relative clause modifier of an NP is a relative clause modifying the NP. The link points from the head noun of the NP to the head of the relative clause, normally a verb.

Examples:
"I saw the man you love" → `rcmod`(man, love)
"I saw the book which you bought" → `rcmod`(book, bought)

ADJECTIVAL_MODIFIER

`public static final GrammaticalRelation ADJECTIVAL_MODIFIER`

The "adjectival modifier" grammatical relation. An adjectival modifier of an NP is any adjectival phrase that serves to modify the meaning of the NP.

Example:
"Sam eats red meat" → `amod`(meat, red)

NUMERIC_MODIFIER

`public static final GrammaticalRelation NUMERIC_MODIFIER`

The "numeric modifier" grammatical relation. A numeric modifier of an NP is any number phrase that serves to modify the meaning of the NP.

Example:
"Sam eats 3 sheep" → `num`(sheep, 3)

NUMBER_MODIFIER

`public static final GrammaticalRelation NUMBER_MODIFIER`

The "compound number modifier" grammatical relation. A compound number modifier is a part of a number phrase or currency amount.

Example:
"I lost $ 3.2 billion" → number($, billion)

QUANTIFIER_MODIFIER

public static final GrammaticalRelation **QUANTIFIER_MODIFIER**

The "quantifier phrase modifier" grammatical relation. A quantifier modifier is an element modifying the head of a QP consitutent.

Example:
"About 200 people came to the party" → quantmod(200, About)

NOUN_COMPOUND_MODIFIER

public static final GrammaticalRelation **NOUN_COMPOUND_MODIFIER**

The "noun compound modifier" grammatical relation. A noun compound modifier of an NP is any noun that serves to modify the head noun. Note that this has all nouns modify the rightmost a la Penn headship rules. There is no intelligent noun compound analysis.

Example:
"Oil price futures" → nn(futures, oil), nn(futures, price)

APPOSITIONAL_MODIFIER

public static final GrammaticalRelation **APPOSITIONAL_MODIFIER**

The "appositional modifier" grammatical relation. An appositional modifier of an NP is an NP that serves to modify the meaning of the NP. It includes parenthesized examples

Examples:
"Sam, my brother, eats red meat" → appos(Sam, brother)
"Bill (John's cousin)" → appos(Bill, cousin)

ABBREVIATION_MODIFIER

public static final GrammaticalRelation **ABBREVIATION_MODIFIER**

The "abbreviation appositional modifier" grammatical relation. An abbreviation modifier of an NP is an NP that serves to abbreviate the NP.

Example:
"The Australian Broadcasting Corporation (ABC)" → abbrev(Corporation, ABC)

PARTICIPIAL_MODIFIER

public static final GrammaticalRelation **PARTICIPIAL_MODIFIER**

The "participial modifier" grammatical relation. A participial modifier of an NP or VP is a VP[part] that serves to modify the meaning of the NP or VP.

Examples:
"truffles picked during the spring are tasty" → partmod(truffles, picked)
"Bill picked Fred for the team demonstrating his incompetence" → partmod(picked, demonstrating)

INFINITIVAL_MODIFIER

public static final GrammaticalRelation **INFINITIVAL_MODIFIER**

The "infinitival modifier" grammatical relation. A participial modifier of an NP is an S/VP that serves to modify the meaning of the NP.

Example:
"points to establish are ..." → infmod(points, establish)

ADVERBIAL_MODIFIER

`public static final GrammaticalRelation ADVERBIAL_MODIFIER`

The "adverbial modifier" grammatical relation. An adverbial modifier of a word is an RB or ADVP that serves to modify the meaning of the word.

Examples:
"genetically modified food" → `advmod`(modified, genetically)
"less often" → `advmod`(often, less)

NEGATION_MODIFIER

`public static final GrammaticalRelation NEGATION_MODIFIER`

The "negation modifier" grammatical relation. The negation modifier is the relation between a negation word and the word it modifies.

Examples:
"Bill is not a scientist" → `neg`(scientist, not)
"Bill doesn't drive" → `neg`(drive, n't)

MEASURE_PHRASE

`public static final GrammaticalRelation MEASURE_PHRASE`

The "measure-phrase" grammatical relation. The measure-phrase is the relation between the head of an ADJP/ADVP and the head of a measure-phrase modifying the ADJP/ADVP.

Example:
"The director is 65 years old" → `measure`(old, years)

DETERMINER

`public static final GrammaticalRelation DETERMINER`

The "determiner" grammatical relation.

Examples:
"The man is here" → `det`(man,the)
"Which man do you prefer?" → `det`(man,which)

PREDETERMINER

`public static final GrammaticalRelation` **`PREDETERMINER`**
The "predeterminer" grammatical relation.

Example:
"All the boys are here" → `predet`(boys,all)

PRECONJUNCT

`public static final GrammaticalRelation` **`PRECONJUNCT`**
The "preconjunct" grammatical relation.

Example:
"Both the boys and the girls are here" → `preconj`(boys,both)

POSSESSION_MODIFIER

`public static final GrammaticalRelation` **`POSSESSION_MODIFIER`**
The "possession" grammatical relation.

Examples:

"their offices" → `poss`(offices, their)

"Bill 's clothes" → `poss`(clothes, Bill)

POSSESSIVE_MODIFIER

`public static final GrammaticalRelation` **`POSSESSIVE_MODIFIER`**
The "possessive" grammatical relation.

Example:

"John's book" → `possessive`(John, 's)

PREPOSITIONAL_MODIFIER

`public static final GrammaticalRelation` **`PREPOSITIONAL_MODIFIER`**

The "prepositional modifier" grammatical relation. A prepositional modifier of a verb, adjective, or noun is any prepositional phrase that serves to modify the meaning of the verb, adjective, or noun.

Examples:
"I saw a cat in a hat" → `prep`(cat, in)
"I saw a cat with a telescope" → `prep`(saw, with)
"He is responsible for meals" → `prep`(responsible, for)

PHRASAL_VERB_PARTICLE

`public static final GrammaticalRelation` **`PHRASAL_VERB_PARTICLE`**

The "phrasal verb particle" grammatical relation. The "phrasal verb particle" relation identifies phrasal verb.

Example:
"They shut down the station." → `prt`(shut, down)

SEMANTIC_DEPENDENT

`public static final GrammaticalRelation` **`SEMANTIC_DEPENDENT`**

The "semantic dependent" grammatical relation has been introduced as a supertype for the controlling subject relation.

CONTROLLING_SUBJECT

`public static final GrammaticalRelation` **`CONTROLLING_SUBJECT`**

The "controlling subject" grammatical relation.

Example:
"Tom likes to eat fish" → `xsubj`(eat, Tom)

AGENT

`public static final GrammaticalRelation` **AGENT**

The "agent" grammatical relation. The agent of a passive VP is the complement introduced by "by" and doing the action.

Example:
"The man has been killed by the police" → `agent`(killed, police)