An Integrated Approach for Automatic Aggregation of Learning Knowledge Objects

Amal Zouaq
University of Montreal, Canada
zouaq@iro.umontreal.ca

Roger Nkambou
University of Quebec at Montreal, Canada
nkambou.roger@uqam.ca

Claude Frasson
University of Montreal, Canada
frasson@iro.umontreal.ca

Abstract

This paper presents the Knowledge Puzzle, an ontology-based platform designed to facilitate domain knowledge acquisition from textual documents for knowledge-based systems. First, the Knowledge Puzzle Platform performs an automatic generation of a domain ontology from documents' content through natural language processing and machine learning technologies. Second, it employs a new content model, the Knowledge Puzzle Content Model, which aims to model learning material from annotated content. Annotations are performed semi-automatically based on IBM's Unstructured Information Management Architecture and are stored in an Organizational memory (OM) as knowledge fragments. The organizational memory is used as a knowledge base for a training environment (an Intelligent Tutoring System or an e-Learning environment). The main objective of these annotations is to enable the automatic aggregation of **Learning Knowledge Objects (LKOs)** guided by instructional strategies, which are provided through SWRL rules. Finally, a methodology is proposed to generate SCORM-compliant learning objects from these LKOs.

Keywords: Learning Knowledge Object, Ontology, Semantic Annotation, Organizational Memory, reusability, SCORM

Introduction

The aim of this paper is to show the importance of documents as vehicles of knowledge and the necessity to be able to extract this knowledge for training purposes. In fact, a large number of existing documents reflect the expertise available within communities of practice and documents constitute 80 to 85% of the information stored by many companies (Uren et al., 2006).

We believe that the ability to reuse document content could represent a great opportunity to capture tacit and explicit domain knowledge. This is a key issue in competence development (Zouaq, Frasson & Nkambou, 2006). Training could benefit from document reuse by creating learning objects from documents fragments. This will avoid building learning objects from scratch, which is a very expensive and time-consuming operation. This can also help knowledge dissemination within a community.

The research about learning objects pursues the same goals of knowledge reusability and dissemination. In fact, the philosophy behind learning objects is based on the ability to reuse them. Currently, research investigates how this could be effectively implemented. Semantic web and ontologies can bring an answer to this need and are starting to be widely used in e-Learning communities. Ontologies can serve as a structure to index document content from different points of view and can help formalize it. Most of research projects in learning objects focused on standard learning object metadata to foster their reusability. We believe that the document itself must be appropriately annotated with ontologies and not only its metadata. A semantic model of the

document can be created through the generation of a document concept map. Natural language processing and machine learning provide techniques that are used for text mining and that can help accomplish this objective. The incremental union of document concept maps represents the actual content of the documents knowledge base hence constituting “de facto” a domain ontology.

Moreover, despite the fact that the research in the learning object area is very active, most of the projects seem to forget that the final objective of learning objects is to promote LEARNING. Koohang (2004) stated that learning object’s ultimate purpose is to enhance and facilitate learning. But without an appropriate instructional strategy, we believe that learning objects cannot meet their goal. Learning objects should present their content to the learner in the most pedagogical and effective way. They must facilitate the act of learning. The use of instructional theories to effectively construct a learning object must be envisaged in the dynamic composition of a learning object. This could facilitate the comprehension of the learning object structure, hence easing its indexing and its reusability as parts or as a whole. However, we do not argue that learning objects must be constrained by a unique instructional theory. They must be envisaged as knowledge structures that can be automatically adapted to fit a particular instructional theory.

This paper is organized as follows: First, this paper presents learning object dynamic composition in a nutshell and introduces the Knowledge Puzzle architecture from a functional point of view. It describes our method to generate domain ontologies from textual documents. It also presents a new content model to describe learning knowledge objects (LKO) and their metadata and to provide resources compliant with current e-Learning standards (SCORM, LOM). Second, the paper shows how knowledge objects are created from ontology-based annotations of documents. This enables to formalize and re-purpose key knowledge and store it in an Organizational Memory (OM). The paper also presents the adopted ontological model. Third, the paper shows how Learning Knowledge Objects are aggregated based on the OM content and on instructional theories formalized as SWRL rules. Finally, the paper explains how SCORM-compliant content is generated from the Learning Knowledge Objects.

Learning Object Dynamic Composition and Reuse in a Nutshell

Learning objects and learning object composition and reuse have become very active research issues. In fact, the importance of learning objects derives from the ability to reuse effectively appropriate chunks of knowledge. This ability necessitates a global model based not only on learning object description but also on learning objectives definition.

However, on which basis can learning objects be effectively described? According to Malaxa and Douglas (2005), discovery and reuse of learning objects is based on the availability of human-created metadata or semantic annotation. The main challenges that face manual metadata creation are the high cost of production and the errors that such a process involves. This constitutes a serious barrier to the successful use of metadata to facilitate reuse and sharing.

Therefore, the first issue that must be tackled when creating metadata is the ability of reducing this overhead through the (semi) automation of semantic annotations. Research projects such as AMG (Cardinaels, Meire, & Duval, 2005) or Tangram (Jovanović, Gasevic & Devedzic, 2006) proposed an alternative to the manual generation of metadata based on automatic frameworks. Such approaches must, however, still mature before becoming extensively used.

Another trend in the field of semantic (manual and/or automatic) annotation is concretized by the use of ontologies, which are an angular stone in the semantic web vision (Berners-Lee, & Lassila, 2001). But again, the problem of the manual creation of these ontologies arises. Moreover, the majority of ontology-based approaches focuses on the use of ontologies for metadata creation and

neglects the real content of the document or learning object. To be more precise, such ontologies describe the world outside the document, like the language of the document, the author, and the keywords and do not model its real content. Gasevic, Jovanović and Devedzic (2004) and Devedzic (2004) emphasize the need to describe a learning object through a domain ontology for effective indexing and reuse. However, the same difficulties face a manual ontology construction and suppose the extensive and continuous contribution of domain experts. This paper aims to propose a semi-automatic approach for domain ontology generation and document indexing. The authors present an approach that is domain-independent and that can be used for many kinds of documents (learning objects, domain documents, training manuals, technical manuals, etc.).

Another issue faces the problem of automatic composition of learning objects: building a description of their content or context is not enough. There must be a framework that triggers this automatic composition (Lytras & Sicilia, 2005). Competency-based models establish a correspondence between learning needs and learning resources through an analysis of the learner profile, of the available resources and of the required learning objective (Sicilia, 2005; Tuso & Longmire, 2000). Such a model can be represented as a competency ontology as described in this paper.

To summarize, automatic composition of learning objects must rely on an integrated approach of knowledge, competency and training management. The state of the art in this area underlines the need for the automation of the whole process. This paper aims to contribute to such a goal through the presentation of the **Knowledge Puzzle Project**, which is an integrated platform of knowledge management and training (e-Learning, Intelligent Tutoring Systems). Figure 1 summarizes the Knowledge Puzzle Functional Architecture.

Throughout this paper, we explain each step of the functional architecture (Figure 1), we give some useful references about the state of the art in each area, and we present the adopted approach.

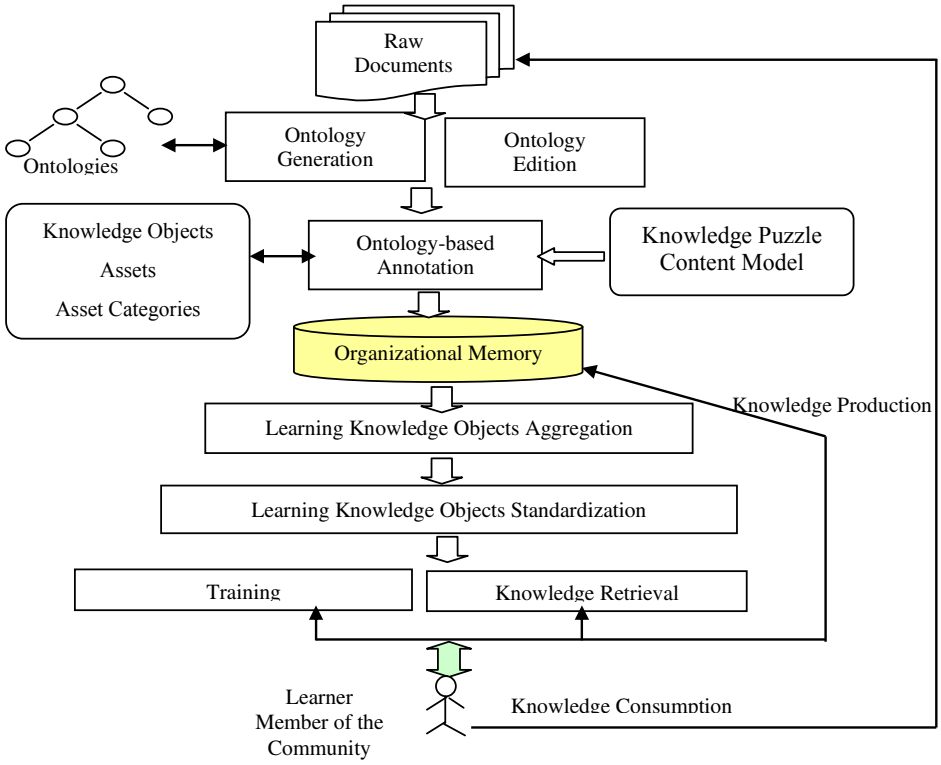


Figure 1: The Knowledge Puzzle Platform Functional View

Finally, one interesting point about the work described here is that it allows bringing together two communities that are generally separated: the e-Learning and the Intelligent Tutoring Systems communities. In fact, a more fine-grained representation of learning object content allows both communities to share the same content while probably deploying different instructional strategies. Fournier-Viger, Najjar, Mayers, and Nkambou (2006) talked about "glass-box learning objects" that explicitly connect the description of learners' cognitive processes to learning objects. Similarly, this paper argues for the necessity to explicitly state **learning object content** in terms of fine-grained knowledge pieces through natural language processing. This opens the door to the deployment of classic intelligent tutoring systems mechanisms such as diagnosis, model tracing and other tutoring services, thanks to the domain ontology.

So, the first issue to tackle to obtain a semantic representation of document content is the issue of ontology learning (or generation) from text.

Ontology Generation from Text

Ontology generation from text is becoming an important area of ontology engineering. As evoked before, manual generation of ontologies is a very time-intensive and error-prone process. Automatic ontology generation requires the use of natural language processing (NLP) technologies and text mining strategies. Although far from being perfect due to language ambiguities, NLP helps to discover interesting structures in textual documents.

State of the Art in Ontology Generation

Many interesting papers exist regarding ontology learning from text and explain the general process of such a generation (Buitelaar, Cimiano & Magnini, 2005; Maedche & Staab, 2000, 2004). Buitelaar et al. (2005) describe the ontology generation layers as consisting of six extraction layers of growing complexity: terms, synonyms, concepts, taxonomy, relations and rules. Other systems implement specific algorithms and present case studies and interesting results such as ASIUM (Faure & Nedellec, 1999), TextToOnto (Maedche & Staab, 2000), Ontolearn (Navigli, Velardi, & Gangemi, 2003), InfoSleuth (Hwang, 1999), OntoLT (Buitelaar, Olejnik & Sintek, 2004) and GlossOnt (Park, 2004).

For example, Faure and Nedellec (1999) implemented a system called **ASIUM** that uses an unsupervised method based on syntactic parsing to acquire sub-categorization frames of verbs and ontologies.

In **TextToOnto**, a collection of domain documents is annotated with NLP tools to extract a number of occurring terms. An association rules algorithm then finds correlations in the co-occurrence of classes of terms, and the system identifies possible relations between these terms. Finally, the system represents these terms and relations as classes in the ontology (Maedche & Staab, 2000).

In **InfoSleuth**, human experts provide a small number of high-level concepts or seeds to the system that are used to automatically collect relevant documents from the web. Then the system extracts phrases containing seed words, generates corresponding concept terms and stores them in the ontology. InfoSleuth extracts several kinds of relations such "is-a", and "assoc-with". A human expert is consulted to verify the correctness of the concept (Hwang, 1999).

OntoLT is an interesting project because it provides a plug-in for the Protégé ontology development environment. It defines a number of linguistic patterns to map Protégé classes and slots to annotated texts (Buitelaar et al., 2004).

In **GlossOnt**, the author proposes a semi-automatic method for building *partial* ontologies which focuses on a particular domain concept at a time and which represents only domain concepts and relationships regarding the target concept. The proposed method takes a target concept from the

user and searches knowledge sources about the target concept, such as domain glossaries and web documents. It then extracts ontological concepts and relationships that are relevant to the target concept (Park, 2004).

Finally, Haase and Volker (2005) present **Text2Onto**, a framework to generate consistent OWL ontologies from learned ontology models by representing the uncertainty of the knowledge in the form of annotations. These annotations capture the confidence about the correctness of the ontology elements. They generate ontologies based on a Learned Ontology Model (LOM), which is then transformed into a standard logic-based ontology language.

The Knowledge Puzzle Approach for Domain Ontology Generation

The aim of the Knowledge Puzzle Approach is to be able to reuse document content for information retrieval and training purposes. The extraction of document content can be obtained through the automatic construction of a **concept map** for each document identifying the important concepts and relations between them. Indeed, several projects related to training have studied the construction of concept maps as a knowledge elicitation technique (Novak & Cañas, 2006) and showed their usefulness in training.

Our approach relies first on obtaining seed words to begin the mining process. This is performed with the use of a machine-learning algorithm that extracts keywords from documents. Then the system builds a semantic concept map by collecting the sentences containing the extracted keywords and parsing them through a statistical NLP parser. A set of lexico-semantic patterns is then applied to the grammatical categories obtained through the parsing process, and results into a semantic concept map. Particularly, triples of the form <concept-verb-concept> are extracted as well as other types of relations expressing time, place, etc. Indeed, verbs express central semantic relations between concepts and specify the interaction between their subjects and objects. According to a number of researchers syntactic dependency relations correspond closely to semantic relations between the entities (Gamallo, Gonzalez, Agustini, Lopes, & de Lima, 2002; Maedche & Staab, 2000; Park 2004). We agree with the fact that domain ontologies rarely model verbs as relations between concepts (Schutz & Buitelaar, 2005). When this is done, the system knows in advance what kind of verbs or knowledge must be mined. The approach adopted in this paper tends to be unsupervised in the sense that it does not have a set of predefined verbs or relations to discover, which is usually done in the text-mining field. The resulting semantic concept map offers a view about the content in the form of concepts and relationships between them. These relations can serve us in the training process to deploy multiple pedagogical strategies. For example, it can serve to give a conceptual overview of a subject area or to make connections between two learned concepts thus enlightening a tacit link, etc.

The whole process builds a document concept map that enables to index document content. The union of all the document concept maps constitutes the domain ontology. As Park (2004) stated, this approach is more feasible than methods that try to build a *full* ontology from a collection of documents. In fact, the system intends to focus on a small number of domain concepts that are the document keywords and identify target concepts and relationships in documents in a more focused manner. This approach can produce more up-to-date ontologies because a document collection within a community is rapidly evolving and new documents can easily be processed.

Keyword extraction

Like the project InfoSleuth (Hwang, 1999), the Knowledge Puzzle project relies on seed words to begin the mining process. Unlike InfoSleuth where a human expert provides these keywords, the

Knowledge Puzzle uses a machine-learning algorithm named Kea-3.0 (Frank, Paynter, Witten, Gutwin, & Nevill-Manning, 1999) to discover document's keywords.

Kea-3.0 is a key phrase (one or more words) extraction algorithm developed by members of the New Zealand Digital Library Project. It is composed of two main phases. During the training phase, the algorithm acquires a Naïve Bayesian model from a set of training documents with their author-supplied keywords. Kea extracts n-grams of a predefined length (e.g. 1 to 3 words) that do not start or end with a stop word. Each document to be analyzed is converted to text format and all its candidate phrases are extracted and converted to their canonical form. For each candidate phrase Kea-3.0 computes 3 feature values:

- **The TFxIDF feature** which describes the specificity of a term for this document under consideration, compared to all other documents in the corpus.
- **The first occurrence feature** is computed as the percentage of the document preceding the first occurrence of the term in the document.
- **The frequency of a phrase feature, which** computes this frequency in the set of key phrases that occur in the training data.

The model uses these features to calculate the probability that each candidate phrase is a key phrase. The most probable candidates are output in ranked order and constitute the document key phrases. A human is consulted to ascertain the quality of the extracted keywords and can modify them as necessary. The Kea-3.0 algorithm was chosen because it performs as state-of-the-art keywords extractors and has been integrated in the GATE architecture for Natural Language Processing, which proves its value. Recently, a new version was made available that enhances the algorithm with a domain thesaurus to guide the extraction process. However, this presupposes the existence of such a thesaurus, which is not the case in the Knowledge Puzzle Platform.

Once keywords are determined, sentences containing them are collected and analyzed through natural language processing.

Semantic concept map extraction

Natural language processing enables to decompose a set of sentences into a structured representation. A number of natural language processing tools have been developed and among them the probabilistic parsers (Charniak, 2000; Collins, 1999; Klein & Manning, 2003). These parsers differ from the others in the sense that they are trained with hand-parsed sentences and try to produce the most probable analysis of new sentences (De Marneffe, MacCartney, & Manning, 2006).

The Stanford Parser is a treebank-trained statistical parser able to generate parses with high accuracy (De Marneffe et al., 2006). The Knowledge Puzzle Platform uses the Stanford typed dependency parsing module to parse the candidate interesting sentences (sentences containing keywords extracted by Kea-3.0). Typed dependencies represent dependencies between individual words and are labelled with grammatical relations, such as subject, direct object or noun compound modifier. More details can be found in (De Marneffe et al., 2006) to describe this component.

The Knowledge Puzzle Platform includes a Graphical Concept Map Editor that enables to view the results of the typed dependency parses described above (Figure 2). The set of candidate key phrases is in the left and the typed dependency structure for the selected sentence is in the right pane.

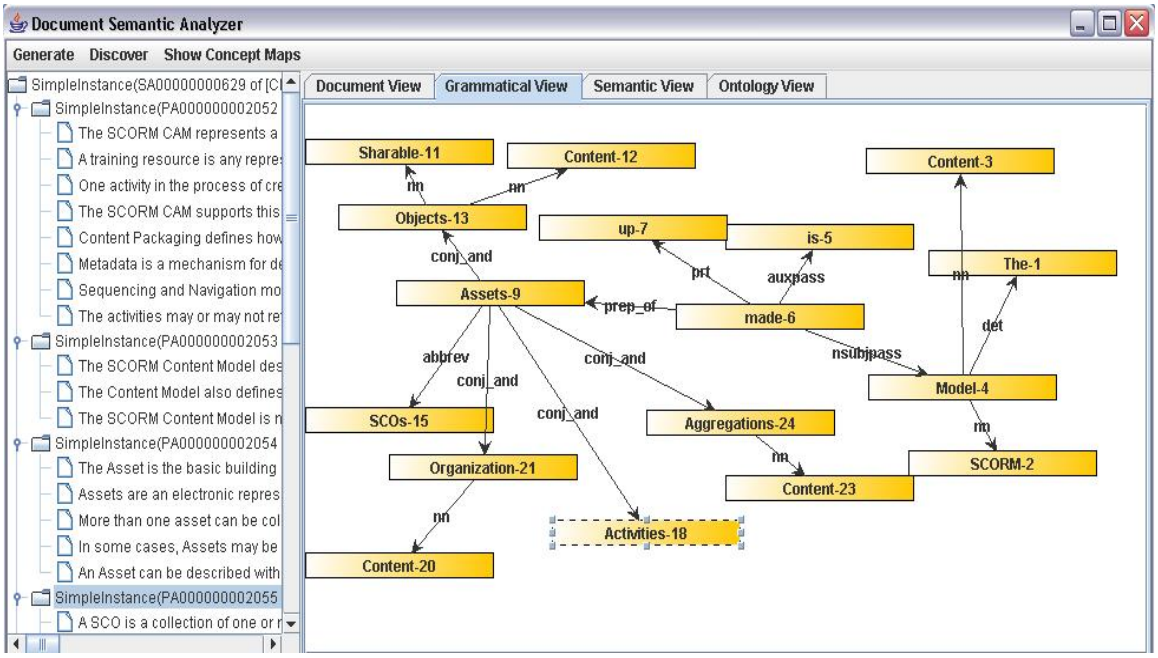


Figure 2: A grammatical concept map

Once a grammatical analysis is performed and results in a typed dependency structure, a semantic analysis takes this representation and interprets it semantically using lexico-semantic patterns. These patterns serve to build concepts and relations from individual words. Examples of such patterns include:

- Aggregation of words linked by specific grammatical relations to form concepts:

  C(w)= w+ noun compound modifier(w, w1) + adjectival modifier(w, w2)

  C(w)= w+ noun compound modifier(w, w1)

- Aggregation of words to form relations :
  - To convert a node verb into a semantic verbal relation
  - To identify verbs and their auxiliary (for active and passive forms)
  - To contract some nodes and their grammatical relation for example: a node verb and a preposition (e.g. is inserted into) thus creating a single verbal relation
  - R(v)= auxiliary(v, v1) + v
  - Where: w, w1, w2 are words
  - C is a function to define a concept
  - R is a function to define a relation

Other patterns serve:

- To delete some words such as determiners (the) or “that” and “which” nodes
- To search acronyms from dependent relations (“dep”)
- To conserve some relations as they are, for example conjunctions and prepositions
- To classify relationships into formal relations such as the relation of hyponymy “is-a”, or meronymy “is-composed-of”, or attribute “has”.

The application of these lexico-semantic patterns builds a semantic view of the previous structure as indicated in Figure 3.

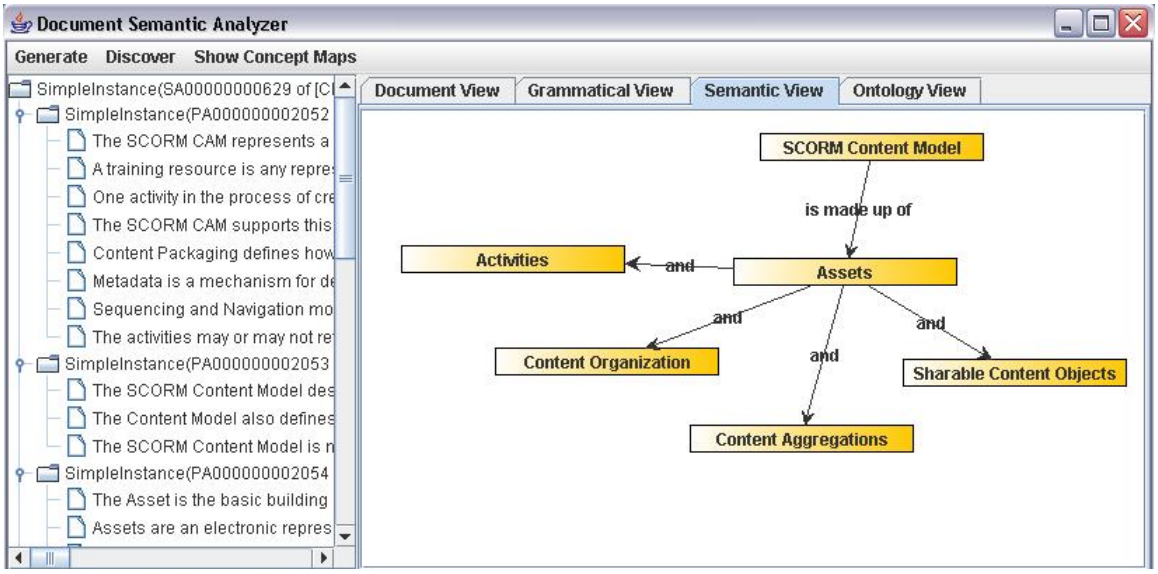


Figure 3: A semantic concept map

For example, for the sentence "The SCORM Content Model is made up of Assets, Sharable Content Objects (SCOs), Activities, Content Organizations and Content Aggregations", the system obtains the concepts: **SCORM Content Model**, **Assets**, **Sharable Content Objects, Activities, Content Organizations** and **Content Aggregations**.

Two kinds of semantic relations are also extracted:

- The **verbal relation "is made up of"** between SCORM Content Model and the all the other concepts.
- The **conjunction relation** "and" between Assets and all the other concepts.

The union of the different document concept maps represents the domain ontology. Editing tools are provided to modify the concept maps. Concepts and relations are stored as instances of the Concept and Relation class.

The generation of domain ontology allows document content modeling. However, this is not enough to create learning materials and to index relevant document's portions. In fact, there is a need to formalize document content through a learning content model: The Knowledge Puzzle Content Model.

The Knowledge Puzzle Content Model

A number of learning content models already exist such as the SCORM content aggregation model (SCORM, 2007), the AICC specification (AICC, 2007) or the ALOCOM Model (Verbert & Duval, 2004). However, few of them support the semantic web technologies (Ontology-based content and metadata) and use it to sustain the definition of content objects. Verbert and Duval (2004) studied six content models, compared them to their Abstract Learning Object Content Model (ALOCOM) and showed that they could map on their abstract model.

Most of the content models enable a three-level to an n-level decomposition of learning objects. For instance, SCORM, which is the most widespread standard content model, decomposes the learning resources into assets, sharable content objects and content aggregations. The 2004 ver-

sion of SCORM adds two other components to the content model: Activities and Content Organizations. In this discussion, we would like to focus on assets (fragments in the ALOCOM model) and sharable content objects (content objects in the ALOCOM model). An Asset, which is the **most** basic form of a learning resource, can be represented by a text, an image, a JavaScript file, a web page, an HTML Fragment, etc. A Sharable Content Object constitutes the second level of aggregation. It is a collection of one or more assets that represents a single executable learning resource. This resource can communicate with a Learning Management System (LMS) through the use of the SCORM Runtime Environment.

In fact, the Content Aggregation Model defines components (assets and SCOs) that seem to represent the same concept. Only SCOs can be exploited through the Runtime Environment but both components are packaged and annotated in the same way. A system for scanning automatically SCO's content can retrieve basic information about their asset composition, their main subject as well as other metadata. However, the real content of the asset and the pedagogical reason of its presence in the SCO as well as the design motivations of these resources (Asset, SCO) are **inaccessible.** It is assumed that a human expert provides their instructional framework, which is left implicit (Ullrich, 2004). We also agree with Ullrich's critic (Ullrich, 2004) about the Learning Object Metadata (LOM, 2007) and the learning design IMS-LD: even if LOM's educational category allows for a description of resources from an instructional perspective, the instructional objects are limited (Exercise, Simulation, Questionnaire, exam, experiment, problem statement, self assessment, and lecture). This list does not encompass categories such as Definition, Question, Answer, Example, Explanation, etc. Similarly, IMS LD describes ordered learning activities and the roles of the involved parties but it does not represent the instructional functions of the learning resources.

So the claim presented here is that two problems arise from the available content models and standards: the real semantic content is unknown as well as the instructional content of learning resources. For example, when considering a web page asset, the following questions remain unanswered with current content models:

- What is the content of the web page?
- What is the pedagogical design behind it?
- What is the instructional role of each of its component?

An educational metadata standard should indicate how to specify instructional aspects of a learning resource such as instructional theories and metadata. This could help to find learning resources for a "just-in-time, just-enough" learning aim more tailored to learner's needs and profile. Ontologies could help formalize such a framework. However, they are poorly represented in current content models and especially in most used e-Learning standards: SCORM and IMS-LD. In fact, among the studied of content models, ALOCOM seems to be the only one that uses ontologies and semantic web languages (Verbert & Duval 2004).

To try to overcome these shortcomings, this paper presents a new content model, the Knowledge Puzzle Content Model, which contributes to give answers to the above questions. It enables to describe a learning object and its components at a very fine-grained level:

- **Assets:** In SCORM, assets can be "anything" ranging from a single image to a whole web page. In this paper, assets describe the structural elements of a document: paragraphs, sentences, sections, images, tables, etc.
- **Asset Categories** describe the instructional role of an asset. Figure 4 shows the various asset categories.

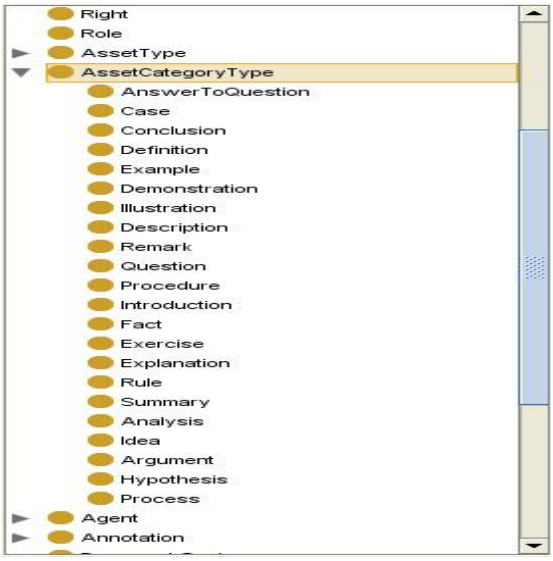


Figure 4: Asset Categories

Asset and Asset categories are created through automatic and manual annotation processes hence constituting semantic annotations of documents. These semantic annotations create Knowledge Objects linked to the annotated documents. This is explained in the following section.

Content Annotation, Edition, and Indexation

The usage of documents and their content has always been an important part of knowledge management. However, Uren et al (2006) stress that semantic annotations of documents can bring new capabilities such as semantic search and interoperability. Indeed, semantic annotation formally identifies concepts and relations between concepts in documents and thus enables a certain knowledge elicitation. Semantic annotation has traditionally taken the form of metadata such as title, author and keywords but neglected the real content of documents. The advent of the Semantic Web and domain ontologies changed the situation by providing unified domain models for document representation. Uren et al (2006) underline that this improvement comes at the cost of increased efforts for authoring content. They suggest that ergonomic authoring environments should be provided to the user to support document content analysis, semantic annotation and ontology engineering from documents. This paper's aim is to contribute to such an environment.

State of the Art

The state of the art in the domain of semantic annotation can be divided into multiple categories, such as semantic annotation platforms (using formal representation) versus non-semantic ones (using non-formal representation), manual versus automatic, frameworks versus tools, and generic platforms versus specific ones.

Annotation Frameworks such as Annotea (Kahan, Koivunen, Prud'Hommeaux, & Swick, 2001) and CREAM (Handschuh & Staab, 2003) are generic frameworks that can be used to create specific annotation tools. There are also many specific tools that produce annotations such as KIM (Popov, Kirayakov, Ognyanoff, Manov & Kirilov, 2004) or MnM (Vargas-Vera et al., 2002). The KIM platform is particularly interesting because it uses NLP and text mining techniques for automatic annotation, indexing, and retrieval of documents (Popov et al., 2003; Popov et al., 2004). KIM is based on the General Architecture for Text Engineering framework (GATE) and produces metadata in the form of named entities (people, places etc.), which are defined in the KIMO ontology.

Dehors, Faron-Zucker, Giboin, and Strombon (2005) present a semi-automatic annotation of learning resources approach based on document layout features. Their basic assumption is that any pedagogical document of reasonable quality holds an underlying model. The method requires a close collaboration between the teacher and an ontologist to decide on the model of the document according to a certain pedagogical strategy. Final annotations are expressed in the form of pedagogical roles.

Automatic Metadata Generation (AMG) framework (Cardinaels et al., 2005) aims at providing an automatic metadata generation system in the form of a web service that generates IEEE Learning Object Metadata. Metadata is derived from the learning object itself by content analysis (Object-based indexers), such as keyword extraction and language classification, but also from the learning object context, which is the learning management system in which the learning object is deployed (Context-based indexers).

Finally, Tangram (Jovanović et al., 2006) is an integrated learning environment for the domain of Intelligent Information Systems that uses an ontology-based approach to automatically annotate learning objects. Tangram enables an automatic metadata generation for learning objects' components through content-mining algorithms and heuristics. It also uses a content structure ontology (based on ALOCOM (Verbert, Klerkx, Meire, Najjar, & Duval, 2004)) in order to decompose a learning object into smaller content units.

The Knowledge Puzzle Approach: An Ontology-based Annotation

The use of ontologies as the backbone of the annotation process for learning object metadata is more and more adopted in e-Learning communities (Aroyo & Dicheva, 2004). Many uses of annotations have been described and many tools for annotating content have been implemented (Marshall, 1998). However, to the best of our knowledge, few annotation platforms for training materials aimed to annotate learning object content and integrated like we do natural language processing (NLP) in the annotation process.

In the Knowledge Puzzle approach, the ontology-based annotations aim to index documents to facilitate their retrieval but also to create a pool of knowledge objects that can serve as a knowledge base for training purposes (Intelligent Tutoring System, learning object composition). The typology of annotations must be designed according to the training objective and must carry out a pedagogical dimension. So the Knowledge Puzzle's indexing strategy must be based on multiple facets:

- Indexing according to content (domain ontology)
- Indexing according to structure (Document Structure Ontology)
- Indexing according to pedagogical role (Instructional Role Ontology)

We developed an ontological model composed of five ontologies in the Protégé Ontology Editor (Knublauch, Fergerson, Fridman Noy, & Musen, 2004) and we used the Web Ontology Language (OWL) to express the ontologies (Protégé OWL). Besides the fact that Protégé (Protégé, 2007) is a good, intuitive and widely used tool, it also provides an open-source Java API that enables to access the ontological model and to use Protégé Forms from a java environment. This feature is very interesting because the graphical interfaces are updated automatically to reflect the changes in the ontological schema.

Indexing according to content: Domain Ontology (DO)

Domain Ontology is organized around the notion of concept. As previously said, it represents the union of the entire document concept maps obtained through the automatic ontology generation described in the first section.

A concept can be linked to another one by a number of relations modeled through the Relation Class. Knowledge objects, assets and asset categories are related to domain concepts either in their content or in their metadata (description, key concepts, etc.). Figure 5 shows a partial view of the generated domain ontology around the notion of "assets".

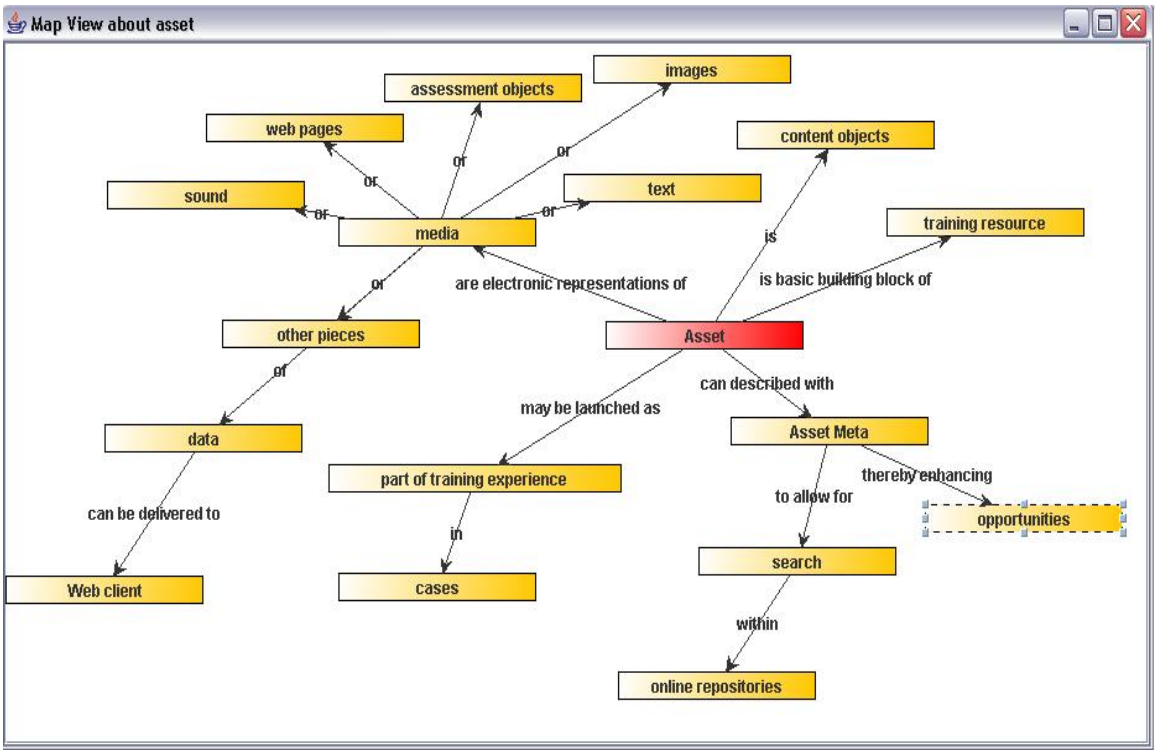


Figure 5: A concept map around the concept of assets

Indexing according to structure: Document Structure Ontology (DSO)

This ontology describes the relevant structural types that can be found in a document (assets), such as sentences, paragraphs, sections, images, tables, figures, etc. Annotation of a document using these assets is called a Structural Annotation and is performed through the Knowledge Extractor (Figure 6). The annotation process transforms a document into a Knowledge Object linked to a structural annotation and assigns metadata such as format, author, and keywords (discovered with the Kea-3.0 algorithm as previously explained).

For the moment, only two asset types are annotated automatically: sentences and paragraphs. To perform this annotation, we use IBM's Unstructured Information Management Architecture (UIMA, 2007). UIMA is an integrated solution that analyzes large volumes of unstructured information to discover, organize and deliver relevant knowledge to the application end-user. The semantic analysis results are processed into Common Analysis Structures (CAS) and then stored in the document structure ontology. Structural annotation allows the indexing of a document according to its structure.

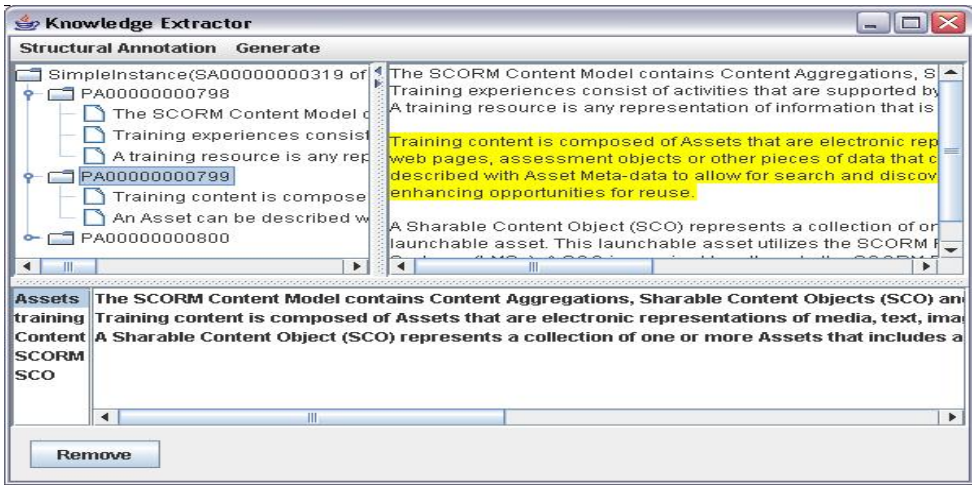


Figure 6: The Knowledge Extractor

Indexing according to pedagogical role: Instructional Role Ontology (IRO)

Because the Knowledge Puzzle's objective is to reuse document content in a training context, indexing must not be limited to content and structure, but must also be performed according to instructional roles.

The instructional Role Ontology models the asset categories introduced in the Knowledge Puzzle Content Model. Questions, Definitions, and Examples are among the possible types.

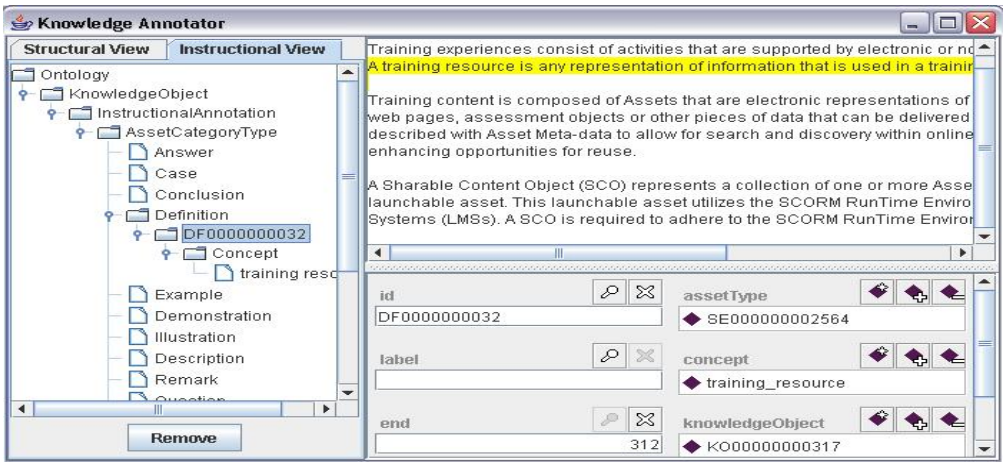


Figure 7: The Knowledge Annotator

This type of annotation is called an Instructional Annotation. This is done manually, through the Knowledge Annotator (Figure 7), by simple drag and drops from document content to the tree representing the asset categories. The annotator has to drop the selected portion of text under the desired asset category, resulting in the creation of an instance of this asset category. Then a domain ontology concept must be attached to the new asset category either by another drag and drop or by selecting it in the asset category property window.

Up to now, this paper only described the ontologies necessary to index a document from three points of view: content, structure and pedagogy. However, this indexing must not be performed in isolation. The Knowledge Puzzle Platform must also model the community in which and for which the indexing takes place. Moreover, because it is used in a training context, it must model

the competencies of the community and link them to the other ontologies. This is done through the organization ontology and the competence ontology.

Organization Ontology (OO)

The organization ontology describes the targeted community and its structure in term of actors, tasks and processes. In the case of a company for example, it describes its divisions, its employees, their roles and other entities (such as human and software agents, places, meetings).

Each class of the organization ontology is also described by a number of concepts of the domain ontology. For example, a member of the community is linked to a set of domain concepts depending on his role, his projects and his interests.

Competence Ontology (CO)

Nash (2005) states that "before using a learning object, learning objectives, desired learner outcomes (performative and measurable), range of content and learner level, and instructional strategies must be in place". Due to the final aim of the Knowledge Puzzle, which is training, a competence is linked to a learning objective. In CREAM, Nkambou, Frasson, and Gauthier (2003) describe a learning objective as the set of abilities or skills to be mastered by a student after a pedagogical activity. The metadata standards such as LOM and SCORM do not provide the means to represent competencies and abilities in an exploitable way. Learning objectives or competencies must be linked to learning objects or parts of them to enable their reusability efficiently.

The abilities are classified, in the Knowledge Puzzle System, according to the Bloom's Taxonomy of Educational Objectives (Bloom, 1956) which is largely used in education in general and which enables the definition of competencies at a very detailed level. The Bloom's Taxonomy uses action verbs to qualify the ability involved in a competence. Bloom defined six levels of intellectual behaviour important in learning and associated a set of verbs to each level. Figure 8 shows the different levels of the taxonomy with examples of verbs for each level.

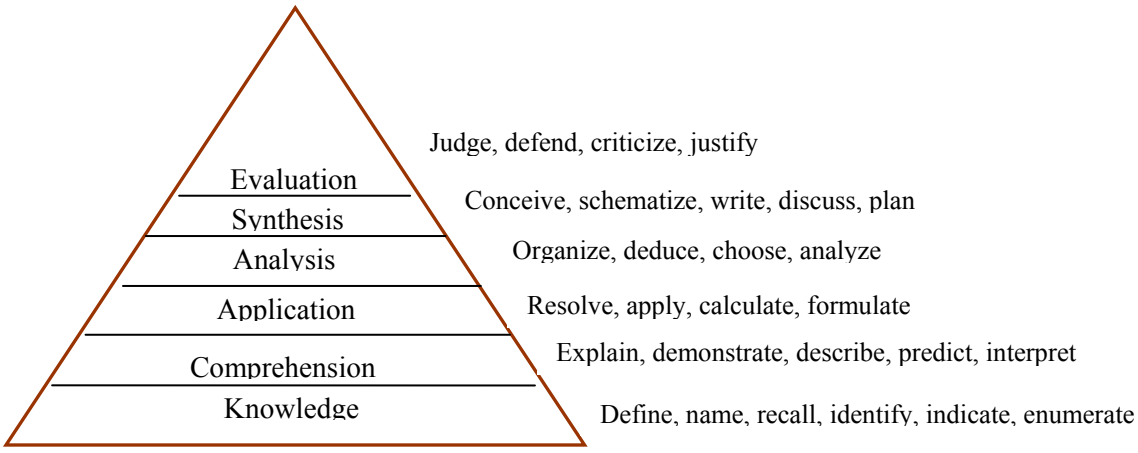


Figure 8: Bloom Taxonomy with Examples of Verbs for each Level

A competence is a set of skills defined on domain concepts. An example of a competence is: "Learn what is SCORM". The set of skills and concepts associated to this competence are: "**define** SCORM" and "**describe** SCORM components". The abilities in the example are indicated in bold and correspond to the levels of acquisition and comprehension. SCORM and SCORM Components are, in the example, concepts of the domain ontology.

Competencies are associated with RDCEO compliant metadata, which also refer to domain ontology concepts. According to IMS (IMS, 2007), the Reusable Definition of Competency or Edu-

cational Objective (RDCEO) specification provides a means to create a common competency model. These competencies can be used to represent learning prerequisites or learning outcomes.

Until now, this paper talked about the adopted way to annotate documents to create knowledge objects, assets, and asset categories, and it introduced different kinds of ontologies. All these structures must be stored in a knowledge base called an Organizational Memory.

Organizational Memory (OM)

We believe that this work could be beneficial for organizations as well as other types of communities. The term "Organizational Memory" is usually more directed towards enterprise communities such as KnowMore (Abecker, Bernardi, & Sintek, 1999) and CoMMA (Gandon, 2002). Few studies tried to merge e-Learning with OM (Abel et al., 2004) and Knowledge Management (Schmidt, 2005). We believe that an OM represents an alternative to the learning object repository (LOR) view. In fact, LORs are static pools of learning resources organized in predefined structures, regardless of the learners' knowledge, preferences, learning styles, etc. An organizational memory is viewed as a knowledge prosthesis in which knowledge objects are stored and retrieved to fulfill the actual need: information retrieval or training through the dynamic aggregation of learning Knowledge Objects.

Figure 9 depicts the content of an organizational memory dedicated to training. In fact, an OM is mainly composed of three layers: the document layer, the annotation layer that contains knowledge objects, assets and assets categories and finally the ontology layer that structure the system's knowledge. An instructional layer in the form of instructional theories linked to SWRL rules can be added to guide learning knowledge objects aggregation. This will be explained more thoroughly in the following section.

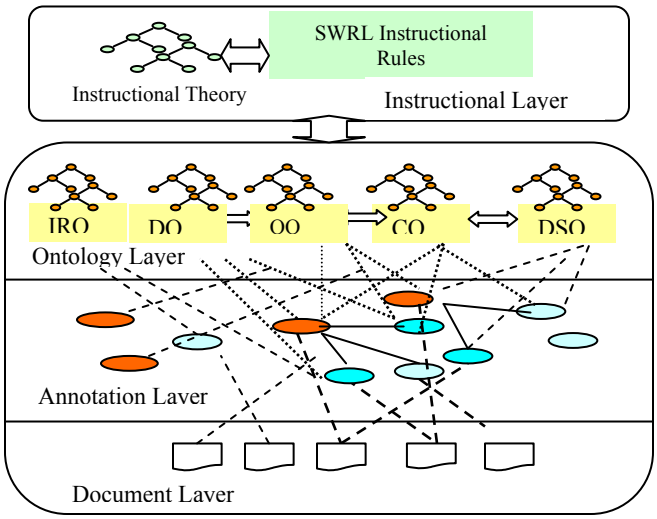


Figure 9: Organizational Memory

Semantic tools able to search its content efficiently must accompany the OM. We developed an ontology navigator to fetch the ontological content and to update it (Figure 10). Moreover, the system is able to search for a concept with different views (sentences, paragraphs, documents) and within different instructional roles (definition, example, explanation, etc.). It can also generate a concept map around a given concept gradually constructed from the annotation of the various documents.

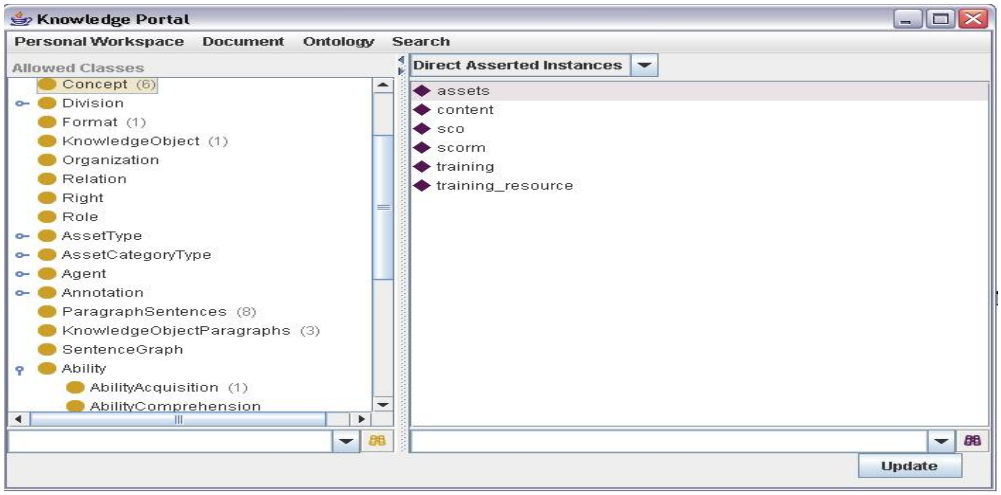


Figure 10: An Ontology Navigator

Automatic Aggregation of Learning Knowledge Objects

State of the Art

Learning objects are a very active research field. Many projects try to deal with learning objects composition and annotation. Some projects that employ automatic approaches to create learning materials and metadata are presented below.

The SeLeNe (Keenoy et al., 2004) project takes DocBook documents as input and transforms them into learning objects. SeLeNe offers services for the sharing and collaborative creation of learning resources and relies also on semantic metadata describing educational material. SeLeNe generates learning structures called trails that are represented as RDF Sequences of learning objects forming the trail. Automatically generated trails are derived from semantic links (related to, part of, has prerequisite) between learning objects, inferred from the information contained in the learning object metadata.

The Trial-Solution (Buffa, Dehors, Faron-Zucker, & Sander, 2005) project, whose domain knowledge is undergraduate mathematics, aims to disaggregate existing electronic books into elementary learning resources, to edit these resources to refine the slicing and to annotate the resources with metadata. The Trial-solution project uses a Slicing Information Technology (SIT) to disaggregate structured format documents such as LaTeX or well-styled Word documents into semantic units (slices) thus constituting a learning resources repository. Then it generates appropriate metadata for these resources containing the pedagogical role of the resource contents like a definition, keywords to specify the topics that the resource contents address and relations with other resources like “references”, “requires”, etc. Metadata is then used to compose personalized documents tailored to the learner‘s knowledge and needs.

The aim of the IMAT project (De Hoog et al., 2002) is to reuse technical manuals as training material using automated document analysis (PDF documents) combined with ontology indexing techniques. The stored and indexed fragments correspond to the logical document structure (sections, tables, images, items, etc.) determined by converting a set of layout objects to a single hierarchical logical structure object. A fragmented document can be indexed according to different points of view: general and syntactical (fragment ontology), semantic (description ontology), instructional and domain.

Verbert et al (2004) propose an ontology that enables a formal definition of Learning Object content structure. This ontology is made to facilitate re-purposing of learning objects at different levels of granularity and thus enables the splitting and the aggregation of learning objects. The authors realized an implementation with OpenOffice.org and MS PowerPoint presentations that allows decomposing the documents into clear segments (slides, paragraphs, lists, list items, images, diagrams and tables). These segments are then categorized into content objects using text patterns and annotated using the AMG framework (Cardinaels et al., 2005). The aggregation process is used when an author wants to build a new learning object. He can search through the learning object content structures (at the content and fragment levels) and reuse the retrieved components.

Finally, Tangram (Jovanović et al., 2006) is an integrated learning environment for the domain of Intelligent Information Systems. It enables an automatic metadata generation for learning objects' components and is grounded on ALOCOM (Verbert et al., 2004). The learner selects the part of the domain ontology that he is interested in. Then the system verifies the learner's knowledge, the required prerequisites and generates an annotated tree of links between concepts. When the learner makes his choice among the concepts, an automatic generation of learning objects is launched, based essentially on learning object metadata (subject, hierarchical relations, ordering relations) and the learner model (preferences, learning style, learning history).

The Knowledge Puzzle Approach for the Creation of Learning Knowledge Objects

The problem of creating learning materials can be solved either by using an authoring environment or by reusing existing resources. These resources can be either dedicated to training (pedagogical material) but can also take the form of domain documents such as reports or notes. In fact, working with pre-existing content is cost-efficient and communities of practice have a lot of electronic documents that can be re-used.

The research projects presented above focused more on training material. They focused on the generation of learning objects metadata and proposed ontologies to improve metadata, but neglected in general the use of ontologies to describe learning objects content except in (Gasevic et al., 2004) where domain ontologies are used to index learning object content.

The SeLeNe project (Keenoy et al., 2004) does not exploit a domain ontology, whereas we do. The Trial Solution do use a thesaurus as its domain knowledge and tries to classify document content by searching the document for a list of sentences and keywords provided by the thesaurus. This lightweight domain ontology does not reflect automatically new domain knowledge nor does it exploit document content. All the projects have the objective of creating a sort of learning object repository whereas we have the objective of constituting a memory of knowledge objects (organizational memory) that can be used to dynamically assemble learning knowledge objects. Moreover, as far as we know, none of the projects used, as we do, natural language processing tools to represent document content into concept maps and to generate a domain ontology. We are concerned with the indexing of source documents as well as with the automatic composition of learning knowledge objects based on the organizational memory.

To enable reusability of learning content as a whole or as portions, an explicit and formal definition of the learning object structure must be provided. This can be done through the introduction of instructional design.

Instructional design and learning knowledge objects

According to Bourdeau, Mizoguchi, Psyché, and Nkambou (2004), few authoring environments offer knowledge representation of instructional theories and principles, and none of them possesses declarative knowledge about how to structure a learning environment or what instructional

methods should be used. In fact, instructional design is necessary to guide the authoring process of learning materials that effectively help the learner in achieving his learning goals (Ullrich, 2005).

The use of domain independent principles of instruction enables to compose a teaching material based on sound pedagogical strategies and allows memorizing why an instructional decision was made. A clear composition structure enables a software to access the pedagogical design of learning resources and hence help reuse of these resources. Moreover, a human designer must be able to search for training material based on pedagogical principles. Thus instructional design must be supported in the authoring process and also in the automatic composition of learning objects in an ontology-based environment.

Bourdeau et al. (2004) suggest that common conceptual structures can be used to explain existing instructional theories. The Knowledge Puzzle takes an instructional theory as input and offers a mapping of the theory's principles with the asset categories (instructional role ontology). An Instructional Theory is formalized as a set of Instructional Steps that are expressed in the form of rules combining asset categories and predefined methods.

SWRL (Semantic Web Rule Language) is used as the rule formalism. SWRL enables Horn-like rules to be used with an OWL knowledge base. SWRL rules can then exploits OWL classes, instances, and properties in their antecedent (body) and consequent (head).

- Predefined methods exploit the knowledge already available in the organizational memory. For instance, if a theory requires the presentation of prerequisites, then there is a method able to use the domain ontology to retrieve the appropriate prerequisites. In fact, four predefined methods are offered for the moment:
- The Learning Objective Method is used to find the learning objectives of the current learning knowledge object. These objectives can be found in the definition of the competence's skills and in the competence description;
- The Prerequisite method searches the appropriate prerequisites according to the competence ontology and to the learner model;
- The Learner Score method is used to determine the actual score of a learner in a question or an exercise;

The Content Method is used to retrieve the actual content of the learning knowledge object. It serves to trigger the rules associated with the Bloom's level that must be mastered. As previously stated, competencies are represented as a set of skills on concepts, these skills being determined according to Bloom's taxonomy of educational objectives. Each level in the taxonomy is matched with the most probable asset categories, and SWRL is used another time to declaratively encode the instructional strategy related to each level or each verb of the taxonomy. For example, in order to define a concept (which is the Knowledge level of Bloom Taxonomy), then an Introduction and a Definition about the concept must be provided. The following SWRL syntax formalizes this rule:

AbilityAcquisition(define) And Concept(?y) And AssetCategoryType(?z) And assetCategory(?y, ?z) → Introduction(?z) And Definition(?z) And concept(?z, ?y)

Figure 11 summarizes the most probable asset categories to use in order to master each Bloom's level.

As a proof of concept, we chose to implement the famous Gagné's theory for the design of instruction based on nine Events of Instruction (Gagné, Briggs, & Wagner, 1992). It is a theory that describes a hierarchy of intellectual skills organized according to complexity that can be used to arrange the learner's external conditions of learning. According to Gagné et al. (1992), instruction

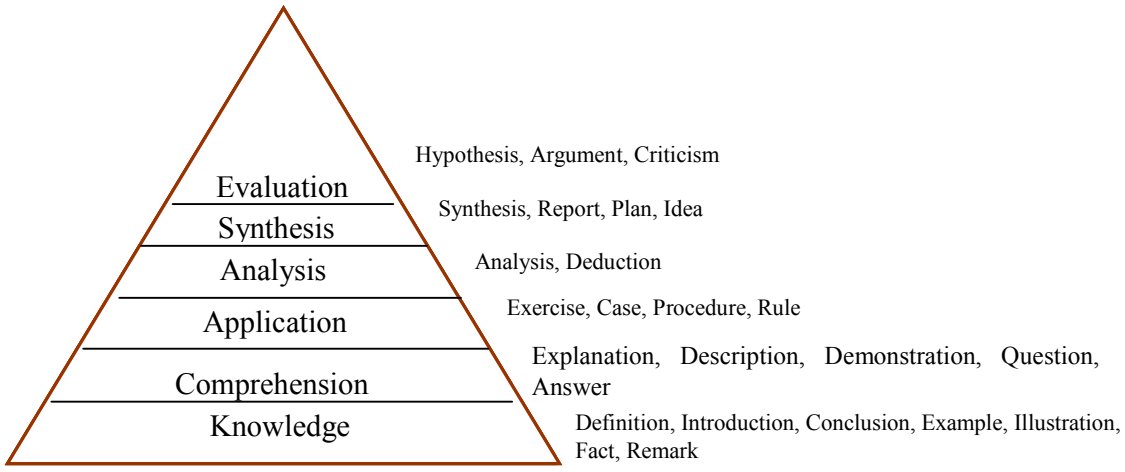


Figure 11: Correspondence between Bloom Taxonomy Levels and Asset Categories

can be seen as a set of instructional events that have distinct effects upon the learner. Gagné's theory identifies nine steps. The first one is to gain the learner's attention (reception) and control it, then to inform the learner of the expected outcomes of the learning activity. Once it is done, the recall of relevant prerequisites should be stimulated. The presentation of the stimulus material is then performed with an appropriate learning guidance and feedback about performance. The learner can verify his performance in one or more situations and additional examples must be used to insure transfer (generalization) and retention.

We modeled the 9 events of instructions as steps of Gagné's Theory (a formalization of such theory in term of operational ontology is already available (Bourdeau et al., 2004). For each of the steps, we elaborated a set of SWRL rules used to fulfill its learning objective based on a combination of asset categories or on the described predefined methods.

For example, to gain the learner attention, which is the first step of the Gagné theory, an illustration of the concept can be presented. This is schematized in the following rule:

InstructionalStep(Gain_attention) And Concept(?y) And AssetCategoryType(?z) And assetCategory(?y, ?z) → Illustration(?z) And concept(?z, ?y)

The available rules are changeable - an interesting point that must be underlined. Indeed, an instructional designer can use the Knowledge Puzzle platform to encode his own rules or to apply another instructional theory. In order to do so, he must use the **Theory Editor** in order to enter the theory's instructional steps. Then he can employ the **Pedagogical Scenario Editor** to define, for each instructional step, a set of SWRL rules that takes as consequent either data (asset categories) or the predefined methods presented above. The instructional designer has to select in the rule panel the action type (either data or method) then he simply selects the desired asset categories or methods. A text box shows the resulting rule. When satisfied, the instructional designer can save his rules.

We define a Learning Knowledge Object as a learning object that is able to explain its own structure and the pedagogical intension behind it. Figure 12 summarizes how a Learning Knowledge Object is generated starting from documents and annotations and using theory-aware SWRL rules.

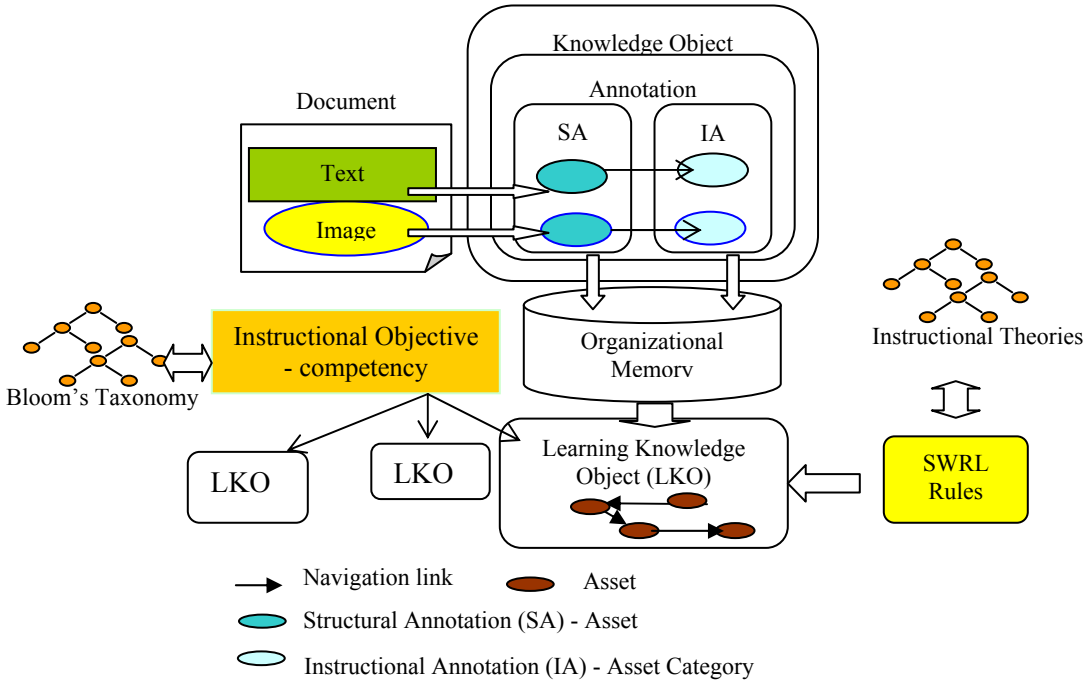


Figure 12: A Learning Knowledge Object creation Process

The available theories are used to generate a theory-aware learning knowledge object. Being theory-aware does not mean that the Learning Knowledge Object has a fixed structure or is restricted to one theory. In fact, any instructional theory could be applied to the OM content to constitute a Learning Knowledge Object. This brings the following advantage: the composition structure is known hence a human or a program is able to understand it and to understand the instructional objective of the expert that gave the instructional rules. This is not provided in current content models such as SCORM (SCORM, 2007) or ALOCOM (Verbert & Duval, 2004). The ability to access the composition logic based on expert's rules can enable a tutoring agent to find similar rules in a problem remediation process (for example if the learner got bad results in the current Learning Knowledge Object).

In fact, the Gagné's theory is the default one to generate learning knowledge objects, but a human expert who wants to design a learning object is able to select another theory. He can see the results of the generated Learning Knowledge Object in the **Instructional Plan Generator** (Figure 13). This tool enables to see the resulting structure and instantiates all the SWRL rules of the chosen theory. A Learning Knowledge Object is generated for each skill in a competence. Each skill of the competence is represented by an activity that uses all the theory's steps in order to master the learning objective linked to the skill.

The Instructional Plan Generator shows the available learning resources (asset categories) for each step. If a learning resource is not available, it informs the human expert about it. The expert can then use an **Asset Category Editor** in order to provide the missing knowledge. He can also decide to change a particular rule for the current learning knowledge object. This changes the generated structure and the system is able to save the current used rule as an alternative rule for the realization of the step. Thus the system learns by observing the expert's actions.

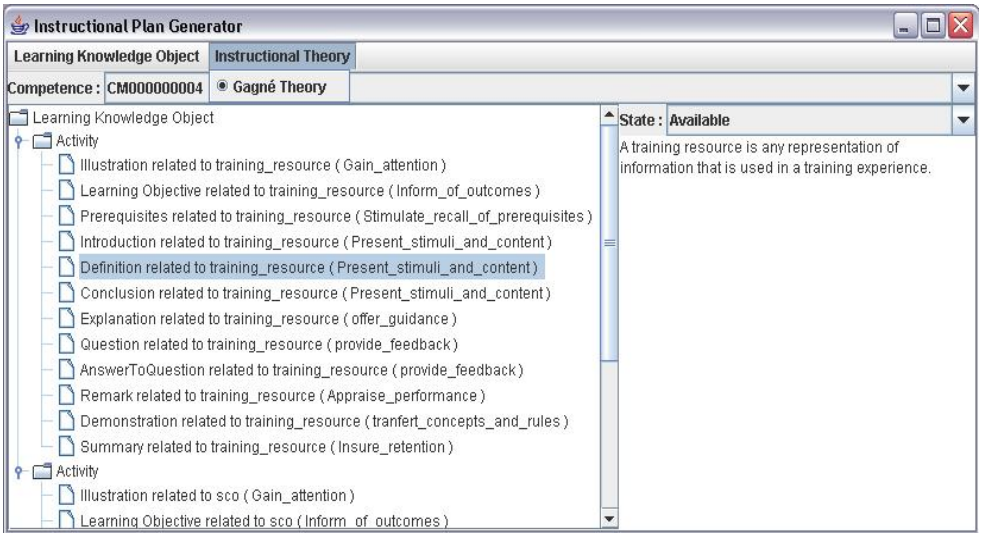


Figure 13: The Instructional Plan Generator

Finally the expert can decide to create a Learning Knowledge Object from scratch through a **Learning Knowledge Object Editor**. For each step of the selected theory, he can search for asset categories or methods through a Knowledge Object Retrieval Tool. He can then compose a learning resource for each step by selecting the appropriate asset categories. Again the system observes the expert's actions and save them as rules.

The paper depicted above the generation process according to the instructional theories. But of course, the Learning Knowledge Object is also tailored to learner's needs. We adopt a ***lazy aggregation*** approach in the sense that we push back the aggregation process until a competence must be mastered by a given learner. This gives the aggregation enough flexibility and enables individual adaptation. A **Competency Gap Analyzer** compares the learner profile (stored in the Organization Ontology) with the competence definition to detect training needs. User's learning objectives are then indicated to the **Instructional Plan Generator (IPG)**. Then according to a pedagogical scenario, the planner searches the OM to gather relevant assets and generates **Learning Knowledge Objects**. Finally, a training environment deploys the learning session in conformance with the generated plan.

Now that we depicted the learning knowledge object generation process, it is important for us to underline that we don't want to build these learning resources in isolation. In fact, we want to use the Knowledge Puzzle Content Model to produce learning materials that can be conformant with current e-Learning standards and more specifically with the SCORM standard.

SCORM Standardization of Learning Knowledge Objects

The goal in this paper is to produce learning knowledge objects that can be compatible with the SCORM (Sharable Content Object Reference Model) standard, which is one of the most used e-Learning standards. SCORM aims to produce reusable learning content as "instructional objects" with high-level requirements such as content reusability, accessibility, durability, and interoperability. It is composed of three main parts: the Content Aggregation Model, the Run-Time Environment, and the Sequencing and Navigation model.

The content Aggregation Model provides a model for creating content packages, applying metadata to the package components and defining a set of navigation rules for sequencing learning activities.

The Run-Time Environment describes a content launch process, a standard communication mechanism (Application Programming Interface) between learning content and the training environment (LMS) and a standard data model to communicate information between the learning content and the LMS.

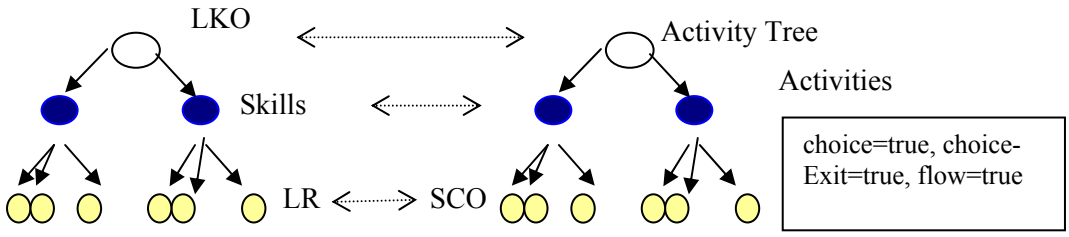
The Sequencing and Navigation Model defines the sequencing guidelines to present content to learners adapted to performance and learners choice at runtime. It uses Activity Trees to describe how learning activities are structured. Activity Trees contains clusters that consist of a single parent activity and its immediate children. The children of a cluster are either leaf activities or other clusters. Leaf activities are linked to content objects and the sequencing strategy defines how these content objects are delivered.

Sequencing and navigation

In the Knowledge Puzzle Content Model, a Learning Knowledge Object corresponds to an Activity Tree composed of a number of activities corresponding to the Learning Knowledge Object skills. A skill is formalized as a SCORM Activity, i.e. a meaningful unit of instruction. Because an activity implements all the steps of an instructional theory, it is in mapped on multiple Sharable Content Objects (SCOs). Each generated learning knowledge object component based on asset categories or on methods execution is considered as a SCORM asset. A SCO is then composed of multiple assets.

For the moment, the system does not apply a very complicated sequencing scheme to the Learning Knowledge Object. In fact, it uses a linear sequencing strategy (choice=true, choiceExit=true, flow=true) to teach each skill, and does not really take advantage of the new SCORM Sequencing and navigation capabilities. This will come in the following version of the system.

Figure 14 depicts the mapping between SCORM Sequencing and Navigation (SN) Components and the Knowledge Puzzle Content Model (KPCM):



LKO : Learning Knowledge Object SCO : Sharable Content Object

LR : Learning Resource obtained through asset categories or methods

Figure 14: A mapping between the KPCM and the SCORM SN

Content packaging

In SCORM, there is no well-defined place to link a learning resource to its educational objectives. The use of LOM classification element enables to circumvent this drawback.

As far as metadata is concerned, a Learning Knowledge Object is associated to a Content aggregation package as it is intended to be delivered to a learner. A complete metadata description file is provided even if it is not required by SCORM (it is considered a best practice). Specifically, the LOM classification element is used to define the competence associated with the learning knowledge object. The SCORM Content aggregation Model is also used to specify the skill associated to each generated sharable content object. A skill is defined to meet a learning objective in the

Bloom's taxonomy. The learning objective metadata for the activity can be created under the LOM classification element with an "educational objective" purpose. The classification element with a "discipline" purpose defines the domain concept that is concerned by the described skill (Figure 15).

```
<lom> <classification>
<purpose><source> LOMv1.0</source>
<value> educational objective</value>
</purpose> <taxonPath>
<source>
<string language="en-US"> Bloom Taxonomy </string> </sc
<taxon><id> AL0001 </id>
<entry><string language="en-US"> Acquisition Level </strin
</entry></taxon>
<taxon>
<id> ALD0001 </id>
<entry> <string language="en-US"> define </string> </entry
</taxon> </taxonPath>
<description> </description>
<keyword>define</keyword>
<purpose>
<source>LOMv1.0</source>
<value>Discipline</value>
 </purpose>
 <taxonPath>
 <source language="en-US">Domain Ontology</source>
<taxon><id>CO000001
</id><entry>SCORM</entry></taxon>
</taxonPath>
</classification>
</lom>
```

Figure 15: Classification element in the Activity's metadata

A Learning Knowledge Object (LKO) is represented by a content aggregation with a manifest file. An example of a LKO structure is depicted in its metadata file (Figure 16).

```
 <?xml version="1.0" encoding="UTF-8" ?>
 <manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
xmlns:imsmd="http://ltsc.ieee.org/xsd/LOM"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
xmlns:imsss="http://www.imsglobal.org/xsd/imsss"
xmlns:adlseq="http://www.adlnet.org/xsd/adlseq_v1p3" …..>
<metadata> <schema>ADL SCORM</schema>
<schemaversion>CAM 1.3</schemaversion></metadata>
<organizations default="ORG-7A2E ">
<organization identifier="ORG-7A2E " structure="hierarchical">
 <title>Define SCORM Content Model</title>
<item identifier="ITEM-8319 " isvisible="true">
<title>Learning Knowledge Object</title>
<item identifier="ITEM-7B88" identifierref="RES-FAE4"
isvisible="true">
<title>Gain_Attention</title> </item>
<item identifier="ITEM-F307" identifierref="RES-AD51"
isvisible="true">
<title>Inform_of_Outcomes</title></item>
……
<imsss:sequencing> <imsss:controlMode choice="true"
choiceExit="true"
 flow="true" forwardOnly="false" /> -
<imsss:sequencingRules>- <imsss:preConditionRule>
<imsss:ruleConditions
 conditionCombination="any">
<imsss:ruleCondition operator="noOp"
condition="always" />
</imsss:ruleConditions>
<imsss:ruleAction action="skip" />
</imsss:preConditionRule>
</imsss:sequencingRules>
<imsss:rollupRules objectiveMeasureWeight="1.0000"
 /> </imsss:sequencing>
</item>  </organization>
</organizations>
<resources>
<resource identifier="RES-FAE4" adlcp:scormType
="sco" type="webcontent" href="asset1_GA.htm">
<file href="asset1_GA.htm" /> </resource>
<resource identifier="RES-AD51 " adlcp:scormType
="sco" type="webcontent" href="asset2_IO.htm">
<file href="asset2_IO.htm" />
</resource>…</resources> </manifest>
```

Figure 16: LKO Metadata Fragment (imsmanifest.xml)

In order to test the learning knowledge objects compliance with the SCORM standard, we used the SCORM 2004 3rd Edition Conformance Test Suite. It contains the conformance testing software to perform self-testing on LMSs, SCOs and Content Packages. We ran two kinds of tests: a first one to test the LKO structure against the SCORM Content Aggregation Content Package Application Profile, and a second one to test the LKO execution in a SCORM environment by importing the learning knowledge Object structure into the SCORM Runtime Environment (Sample RTE 1.3.3). The first and second test succeeded and we were able to launch the LKO in the Sample RTE 1.3.3.

Conclusion and Further Work

This paper presented an ontology-based approach to semi-automatically annotate document content. Annotations are performed at the content level and at the metadata level. At the content level, document concept maps are automatically extracted using machine learning and natural language processing. These concept maps constitute a domain ontology. At the metadata level, documents are indexed at multiple levels: domain, structure, pedagogy, and competence. This multiple indexation is based on a formal ontological model.

This paper also introduced a new content model, the Knowledge Puzzle Content Model, which defines two components: assets (structural view) and asset categories (instructional view). This decomposition makes possible to retrieve and use assets and assets categories and to automatically create pertinent learning knowledge objects (LKOs). The generation of LKOs uses instructional theories in the form of SWRL rules as an aggregation pattern. Hence the same knowledge structures can be adapted to fit a particular instructional theory and a particular learner model.

The Knowledge Puzzle Content Model maps to a number of existing content models. Thus the annotations can be used to generate standard learning objects (SCORM). To accomplish this generation, we defined a mapping between SCORM components and the Knowledge Puzzle's generated structures.

The next goal in this research will be to explore the use of natural language processing to automatically extract other kinds of metadata such as asset categories, which are manually annotated for the moment. As far as domain ontology is concerned, we would like to implement a statistical layer over document concept maps that will synthesize the domain knowledge and help the expert decide about the importance of domain concepts and relations. This will help formalize the domain ontology through statistical measures. Finally, regarding learning knowledge objects, the objective is to deploy more difficult sequencing strategies that take advantage of SCORM 2004's sequencing capabilities hence constituting more complex learning knowledge objects.

References

Abecker, A., Bernardi, A., & Sintek, M. (1999). Proactive knowledge delivery for enterprise knowledge management. *Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering, Learning Software Organizations, Methodology and Applications*, pp.103–117, Kaiserslautern, Germany.

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

AICC (2007). Retrieved March 19, 2007 from http://www.aicc.org/

Aroyo, L., & Dicheva, D. (2004). The new challenges for e-learning: The educational semantic web. *Educational Technology & Society*, *4*(7), 59-69.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, 34–43.

Bloom, B. S. (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 Intelligent Tutoring Systems,* pp.150-161, Maceio, Brazil.

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 International Workshop on Applications of Semantic Web Technologies for E-Learning*, AIED, pp. 73-76, Amsterdam, Holland.

Buitelaar, P., Cimiano, P. & Magnini, B. (2005). Ontology learning from text: An overview. In P. Buitelaar, P. Cimiano, & B. Magnini (Eds.), *Ontology learning from text: Methods, evaluation and applications*, pp. 3-12. Volume 123 of *Frontiers in artificial intelligence and applications.* IOS Press.

Buitelaar, P., Olejnik, D., & Sintek, M. (2004). A protégé plug-in for ontology extraction from text based on linguistic analysis. *Proceedings of the 1st European Semantic Web Symposium*, pp.31-44, Heraklion, Greece.

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, Chiba, Japan.

Charniak, E. (2000). A maximum-entropy-inspired parser. *Proceedings of the First Conference on North American Chapter of the Association for Computational Linguistics*, pp. 132 - 139, Seattle, Washington.

Collins, M. (1999). Head-driven statistical models for natural language parsing. Ph.D. thesis, University of Pennsylvania.

Dehors, S., Faron-Zucker, C., Giboin, A., & Stromboni, J.-P. (2005). Semi-automated semantic annotation of learning resources by identifying layout features. *Proceedings of AIED'2005 International Workshop on Applications of Semantic Web Technologies for E-Learning*, Amsterdam, Holland.

De Marneffe, M-C., MacCartney, B. & Manning, C.D. (2006). Generating typed dependency parses from phrase structure parses. *Proceedings of the 5th Conference on Language Resources and Evaluation*, Genoa.

De Hoog, R., Kabel, S., Barnard, Y., Boy, G., DeLuca, P., Desmoulins, C., Riemersma, J., & Verstegen, D. (2002). Re-using technical manuals for instruction: creating instructional material with the tools of the IMAT project. *Proceedings of ITS'2002 Workshop on Integrating Technical And Training Documentation*, pp. 28-39, San Sebástian, Spain.

Devedzic, V. (2004). Education and the semantic web. *International Journal of Artificial Intelligence in Education, 14*, 39-65. IOS Press.

Faure, D. & Nedellec, C. (1999). Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system ASIUM. *Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling and Management*, pp. 329-334, Dagstuhl Castle, Germany.

Fournier-Viger P., Najjar, M., Mayers, A. & Nkambou, R. (2006). A cognitive and logic based model for building glass-box learning objects. *Interdisciplinary Journal of Knowledge and Learning Objects*, 2: 77-94. Available at http://ijklo.org/Volume2/v2p077-094Fournier-Viger.pdf

Frank, E., Paynter, G.W., Witten, I..H., Gutwin, C., & Nevill-Manning, C.G. (1999). Domain-specific key phrase extraction. *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pp. 668-673, San Francisco, USA.

Gagné, R. M., Briggs, L. J., & Wagner, W. W. (1992). *Principles of instructional design* (4th ed.). Fort Worth: HBJ College Publishers.

Gamallo, P., Gonzalez, M., Agustini, A., Lopes, G., & de Lima, V.S. (2002). Mapping syntactic dependencies onto semantic relations. *Proceedings of the ECAI Workshop on Machine Learning and Natural Language Processing for Ontology Engineering*, pp. 15-22, Lyon, France.

Gandon, F. (2002). A multi-agent architecture for distributed corporate memories. *Proceedings of the Third International Symposium, From Agent Theory to Agent Implementation*, at the 16th European Meeting on Cybernetics and Systems Research, , pp. 623-628, Vienna, Austria.

Gasevic, D., Jovanović, 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, New Zealand.

Haase, P., & Volker, J. (2005). Ontology learning and reasoning - Dealing with uncertainty and inconsistency. *Proceedings of the ISWC Workshop on Uncertainty Reasoning for the Semantic Web (URSW)*, pp. 45-55, Galway, Ireland.

Handschuh, S., & Staab, S. (2003). CREAM - Creating metadata for the semantic web. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, *42*(5), 579-598.

Hwang, C.H. (1999). Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information. *Proceedings of the 6th International Workshop on Knowledge Representation meets Databases*, pp. 14-20, Linkoeping, Sweden.

IMS (2007). Retrieved March 19, 2007 from http://www.imsglobal.org/competencies/index.html

Jovanović, J., Gasevic, D., & Devedzic, V. (2006). Ontology-based automatic annotation of learning content. *International Journal on Semantic Web and Information Systems*, *2*(2), 91-119.

Kahan, J., Koivunen, M., Prud'Hommeaux, E. & Swick, R. (2001). Annotea: An open RDF infrastructure for shared web annotations. *Proceedings of the WWW10 International Conference*, pp. 623-632, Hong Kong.

Keenoy, K., Poulovassilis, A., Christophides, V., Rigaux, P., Papamarkos, G., Magkanaraki, A., Stratakis, M., Spyratos, N., & Wood, P.T. (2004). Personalisation services for self e-learning networks. *Proceedings of ICWE*, pp. 215-219, Munich.

Klein, D. & Manning, C.D. (2003). Accurate unlexicalized parsing. *Proceedings of the 41st Meeting of the Association for Computational Linguistics,* pp. 423 – 430, Sapporo, Japan.

Knublauch, H., Fergerson, R. W., Fridman Noy, N., & Musen, M. A. (2004). The Protégé OWL plugin: An open development environment for semantic web applications. *Proceedings of the International Semantic Web Conference*, pp.229-243, Hiroshima, Japan.

Koohang, A. (2004). Creating learning objects in collaborative e-learning settings. *Issues in Information Systems*, *4*(2), 584-590.

LOM (Learning Object Metadata) (2007). Retrieved March 19, 2007 from http://ieeeltsc.org/wg12LOM/

Lytras, M. D. & Sicilia, M-A. (2005). Modeling the organizational aspects of learning objects in semantic web approaches to information systems. *Interdisciplinary Journal of Knowledge and Learning Objects, 1*, 255-267. Available at http://ijklo.org/Volume1/v1p255-267Lytras_Sicilia.pdf

Maedche, A., & Staab, S. (2000). Semi-automatic engineering of ontologies from text. *Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering*, Chicago, USA.

Maedche, A. & Staab, S. (2004). Ontology learning. In S. Staab, & R. Studer (Eds), *Handbook on ontologies,* pp. 173-190. Springer.

Malaxa, V & Douglas, I (2005). A framework for metadata creation tools. *Interdisciplinary Journal of Knowledge and Learning Objects, 1*, 151-162. Available at http://ijklo.org/Volume1/v1p151-162Malaxa28.pdf

Marshall, C. (1998). Toward an ecology of hypertext annotation. *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia*, pp. 40-49, Pittsburgh, USA.

Nash, S. (2005). Learning objects, learning object repositories, and learning theory: Preliminary best practices for online courses. *Interdisciplinary Journal of Knowledge and Learning Objects*, *1*, 217-228. Available at http://ijklo.org/Volume1/v1p217-228Nash.pdf

Navigli, R., Velardi, P., & Gangemi, A. (2003). Ontology learning and its application to automated terminology translation. *IEEE Intelligent Systems*, *18*(1), 22-31.

Nkambou, R., Frasson, C., & Gauthier, G. (2003). CREAM-Tools: An authoring environment for knowledge engineering in intelligent tutoring systems. In T. Murray, S. Blessing, and S. Ainsworth, (Eds.), *Authoring tools for advanced technology learning environments: Toward cost-effective, adaptative, interactive, and intelligent educational software*, pp. 93-138. Kluwer Publishers.

Novak, J. D. & Cañas, A. J. (2006). The theory underlying concept maps and how to construct them. Technical Report IHMC CmapTools 2006-01, Florida Institute for Human and Machine Cognition, Retrieved March 19, 2007 from http://cmap.ihmc.us/Publications/ResearchPapers/TheoryUnderlyingConceptMaps.pdf.

Park, Y. (2004). GlossOnt: A concept-focused ontology building tool. *Proceedings of KR 2004*, pp. 498-506, Whistler, Canada.

Popov, B., Kirayakov, A., Ognyanoff, D., Manov, D., & Kirilov, A. (2004). KIM - A semantic platform for information extraction and retrieval. *Natural Language Engineering, 10* (3-4), 375-392.

Popov, B., Kirayakov, A., Ognyanoff, D., Manov, D., Kirilov, A., & Goranov, M. (2003). Towards semantic web information extraction. *Proceedings of the Human Language Technologies Workshop. At 2nd International Se-mantic Web Conference (ISWC2003)*, pp. pp. 1-22, Florida, USA.

Protégé Ontology Editor (2007). Retrieved March 19, 2007 from http://protege.stanford.edu/

SCORM (2007). Retrieved March 19, 2007 from http://www.adlnet.gov/scorm/index.cfm

Schmidt, A. (2005). Bridging the gap between e-learning and knowledge management with context-aware corporate learning (extended version). *Professional Knowledge Management (WM 2005) Post Proceedings*, pp. 203-213, Kaiserlautern, Germany.

Schutz, A. & Buitelaar, P. (2005). RelExt: A tool for relation extraction in ontology extension. *Proceedings of the 4th International Semantic Web Conference*, pp. 593-606, Galway, Ireland.

Sicilia, M. A. (2005). Ontology-based competency management: Infrastructures for the knowledge intensive learning organization. In M. D. Lytras & A. Naeve (Eds*.), Intelligent learning infrastructures in knowledge intensive organizations: A semantic web perspective*, pp. 302-324. Hershey, PA: Idea Group.

Tuso, G. & Longmire, W. (2000). Competency-based systems and the delivery of learning content. In D. Brightman (Ed.), *Learning without limits*, pp. 33-38, Informania Inc.

UIMA (2007). Retrieved March 19, 2007 from http://uima-framework.sourceforge.net/

Ullrich, C. (2004). Description of an instructional ontology and its application in web services for education. *Proceedings of Workshop on Applications of Semantic Web Technologies for E-learning*, pp. 17-23, Hiroshima, Japan.

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., & Ciravegna, F. (2006). Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Journal of Web Semantics, 4*(1), 14-28.

Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A., & Ciravegna, F. (2002). MnM: Ontology driven semi-automatic and automatic support for semantic markup.*Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW 2002)*, pp. 379-391, Spain.

Verbert, K. & Duval, E. (2004). Towards a global architecture for learning objects: A comparative analysis of learning object content models. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pp. 202-208, Chesapeake, VA.

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 the OTM 2004 Workshop on Ontologies, Semantics and E-learning*, , pp. 713-722, Agia Napa, Cyprus.

Zouaq, A., Frasson, C., & Nkambou, R. (2006). An ontology-based solution for knowledge management and elearning integration. *Intelligent Tutoring Systems, 2006*, 716-718

Biographies

Amal Zouaq is a Ph.D. Student at the University of Montreal. She is a member of the Heron Laboratory and of the GRITI Group. Her research interests focus on domain ontology generation, competence ontologies and learning object generation through the use of natural language processing. She also works in the area of Intelligent Tutoring Systems (ITS) with the aim of bridging the gap between the ITS community and the e-Learning community.

Roger Nkambou is currently an Associate Professor in Computer Science at the University of Quebec at Montreal, and Director of the GDAC (Knowledge Management Research) Laboratory (http://gdac.dinfo.uqam.ca). He received a Ph.D. (1996) in Computer Science from the University of Montreal. His research interests include knowledge representation, intelligent tutoring systems, intelligent software agents, ontology engineering, student modeling and affective computing. He is author of more than 80 publications in AIED (Artificial Intelligence in Education) area.

Claude Frasson is Professor in Computer Science at University of Montreal. He is also Director of Heron Laboratory specialized in Intelligent Tutoring Systems (ITS) and Director of the GRITI group, a multidisciplinary research group involving seven universities / 70 researchers in Quebec. He is the founder of the ITS international conference which regroups, every two years, about 350 researchers in the world specialized in Artificial Intelligence and Education, and more specifically in e-Learning. Since 1996, he was pioneering the new generations of e-Learning, involving the contribution of Artificial Intelligence. He also serves as member of the Executive Steering Committee of several internationally renowned Scientific Associations specializing in Artificial Intelligence and Education.