

Enhancing Learning Objects with an Ontology-based Memory

Amal Zouaq and Roger Nkambou, *IEEE Member*

Abstract— The reusability in learning objects has always been a hot issue. However, we believe that current approaches to e-Learning failed to find a satisfying answer to this concern. This paper presents an approach that enables to capitalize existing learning resources by first, creating “content metadata” through text mining and natural language processing and second, by creating dynamically learning knowledge objects, i.e. active, adaptable, reusable and independent learning objects. The proposed model also suggests integrating explicitly instructional theories in an on-the-fly composition process of learning objects. Semantic web technologies are used to satisfy such an objective by creating an ontology-based organizational memory able to act as a knowledge base for multiple training environments.

Index Terms— Applications and Expert Knowledge-Intensive Systems, Computer-managed instruction, Intelligent Web Services and Semantic Web, Knowledge Management.



1. INTRODUCTION

Given the large amount of learning objects (LOs) and their continuous growth, the issue of storing, searching and indexing learning objects is not a trivial one. Most of the approaches [10], [21] suggested the use of standard metadata to index learning objects. Even if metadata allows the description of LOs characteristics (language, version, educational purpose, etc.), we believe that this kind of annotation is not sufficient to effectively guide a learner or a teacher towards the right content.

In fact, the vision of learning object repositories (LORs) that consider learning objects as static content aggregations seems outdated. This is a particularly acute issue since the corporate world has also adopted learning objects as a mean to insure the training of its members at a reduced cost. The just-in time just-enough learning concept requires a very fine-grained model of competences and learning materials. In this context, reusability must be an important parameter in LOs. According to Polsani [34]: “A Learning Object is an independent and self-standing unit of learning content that is predisposed to reuse in multiple instructional contexts”. Aside from the reusability dimension, this definition implies a number of important factors that should be considered when talking about learning objects:

- The independent dimension that implies the autonomy of the LO in term of data and behaviour, which contrasts with the current vision of LO in the e-Learning community;
- The pedagogical dimension that should be taken into account when creating a learning object but which should not be embedded in the LO itself to allow for multiple instructional adaptations.

The current packaging approach of LO fails to consider these two dimensions. Therefore, new processes must be set up in order to create active learning objects, able to manage their own conceptual structures and to take into account learning instructional theories.

How can these structures be easily set-up? In fact, learning objects themselves can provide a solid foundation for obtaining such structures. This can be done through a capitalization process of learning object content, which consists in disaggregating it into small instructional units and generating valuable structures such as concept maps and domain ontologies. The generation of such structures necessitates LO content mining.

This paper presents an approach for building Learning Knowledge Objects (LKO), i.e. active, independent and theory-aware LOs. LKOs are built through a reverse engineering process of existing textual learning resources. This vision is implemented in the Knowledge Puzzle Project through text mining, natural language processing and semantic annotation leading to an ontology-based organizational memory (OM).

The objectives of the project are:

- To produce a memory with new semantic structures to store fine-grained resources;
- To propose a new transformation process to feed the memory from LOs or other types of educational resources;
- To propose a new flexible composition process of LKOs thanks to the OM;
- To propose solutions for LKO deployment in training environments, including standard-based Learning Management Systems (LMS).

The paper is organized as follows: First, we discuss the state-of-the-art on LOs sharing and reusing issues. Second, we present the structure of the OM followed by the explanation of the capitalization process (the process that aims to feed the OM). Third, we explain how the OM is

• The authors are with the University of Quebec at Montreal, Pavillon Sherbrooke 200, rue Sherbrooke ouest, local SH-5720, Montreal, QC H2X 3P2, Canada. E-mail: {zouaq.amal, nkambou.roger}@uqam.ca.

Manuscript received (insert date of submission if desired). Please note that all acknowledgments should be placed at the end of the paper, before the bibliography.

exploited for LKOs composition and how resulting LKOs are deployed in standard and non-standard learning environments. Finally, we illustrate the value of the approach through a semantic evaluation and some results before concluding.

2. RELATED WORKS

Sharing learning objects has always been a great concern in the e-Learning community. Several solutions have been proposed [21], [23]. For example, the SeLeNe platform [23] aims at providing a distribution infrastructure to enable LO sharing in a peer network. SeLeNe also provides some basic functions that allow the creation of complex LOs from simple ones.

The importance of sharing LOs is closely linked to the notion of reuse and retrieval. Searching and composing LOs to fulfill a learning goal is not an easy task. Some research works proposed extensions to LO structure to better guide the user to a specific point of the LO. These extensions include manually created relationships between a LO and its parts [40] as well as semantic links between LOs and domain knowledge [25], [37]. These last years, the semantic web [6] has provided interesting perspectives to this idea of modeling domain knowledge through the establishment of domain ontologies. We believe that LO composition should be guided by a conceptual knowledge space such as domain ontologies that could emerge automatically from the LO content. Putting LOs in a knowledge space allows a better search and eases the composition process. It also permits the discoverability of the real learning object content while standard metadata based-search mainly focuses on LO's external characteristics (language, version, author...). Moreover, e-Learning standards (SCORM, IMS-LD...) currently do not include an ontological base that could offer a common knowledge space that can benefit to multiple learning environments including intelligent tutoring systems.

Since learning content is often in textual format, our project focuses especially on this type of content. The question is how to create a knowledge space associated with this content. Classical knowledge engineering tools constitute a solution but the cost associated with the involvement of domain experts is a significant limitation. To alleviate the knowledge acquisition bottleneck, producing this knowledge space can be supported by automatic techniques for learning a domain ontology from text. However, this is an important and difficult issue. Few research works tried to tackle this problem such as Text-to-Onto [26] and Onto-Learn [30]. Evenly, educational data mining is becoming a very important area in the Artificial Intelligence in Education community [19], [29], but very few works are concerned with domain ontology learning for educational purposes. Most of the works focus on learner model mining: several techniques are used to extract relevant data mainly from databases and learning sessions log files. Some works focus on learner or group classification, clustering or sorting [3], [16] when others focus on sequential pattern mining to depict some significant patterns indicating gaming, suc-

cess or failure [4], [22], [31]. Another important issue is to be able to set up a framework in which LO should result from an automatic composition process capable of fulfilling a specific competence need. Current approaches fail to provide such a capability at a fine-grained level. Furthermore, they do not embark learning theories' principles in the composition process whereas several works including [5], [28] have underlined the importance of these principles in instructional design. In fact, we believe that instructional theories can also provide the LO designer with formal principles for the aggregation of assets in the LO. However, these theories are not currently really considered during LO creation. Even if several projects have integrated instructional and learning theory in learning scenario design or high level learning objects composition [17], [38], they do not rely on relevant and small instructional units but on whole learning objects. These resources sometimes already carry an implicit theory that is not accessible to learning objects crawlers or indexers. Therefore, it is important to enable the explicit integration of learning theory in the LO design right at the time of the competence need and not before.

To sum up, while a variety of approaches to LO design are proposed in the literature, they lack a global vision about learning object that corresponds to Polsani's definition [34]. This vision necessitates the incorporation of various techniques including text mining and learning object composition and decomposition. The Knowledge Puzzle aims at providing this integrated approach.

3. ORGANIZATIONAL MEMORY STRUCTURE

The Knowledge Puzzle is an integrated framework that intends to capitalize learning objects content through the constitution of an organizational memory (OM). Fig. 1 depicts the detailed technical architecture of the components that create the OM content.

The central element of this architecture is the organizational memory. We propose a reverse engineering approach to LOs and other content (textual documents) that leads to a four-layer memory as shown in fig. 1: the document pool, the ontology layer, the resource layer and the rule layer.

The OM serves as the basis of a composition process, whose aim is to propose active, adaptable and theory-aware learning knowledge objects (LKO). Contrary to classical learning object (LO), an LKO is an active open resource that gives access to knowledge structures (domain ontology and concept map) related to the targeted competence and (semi) automatically obtained through text mining.

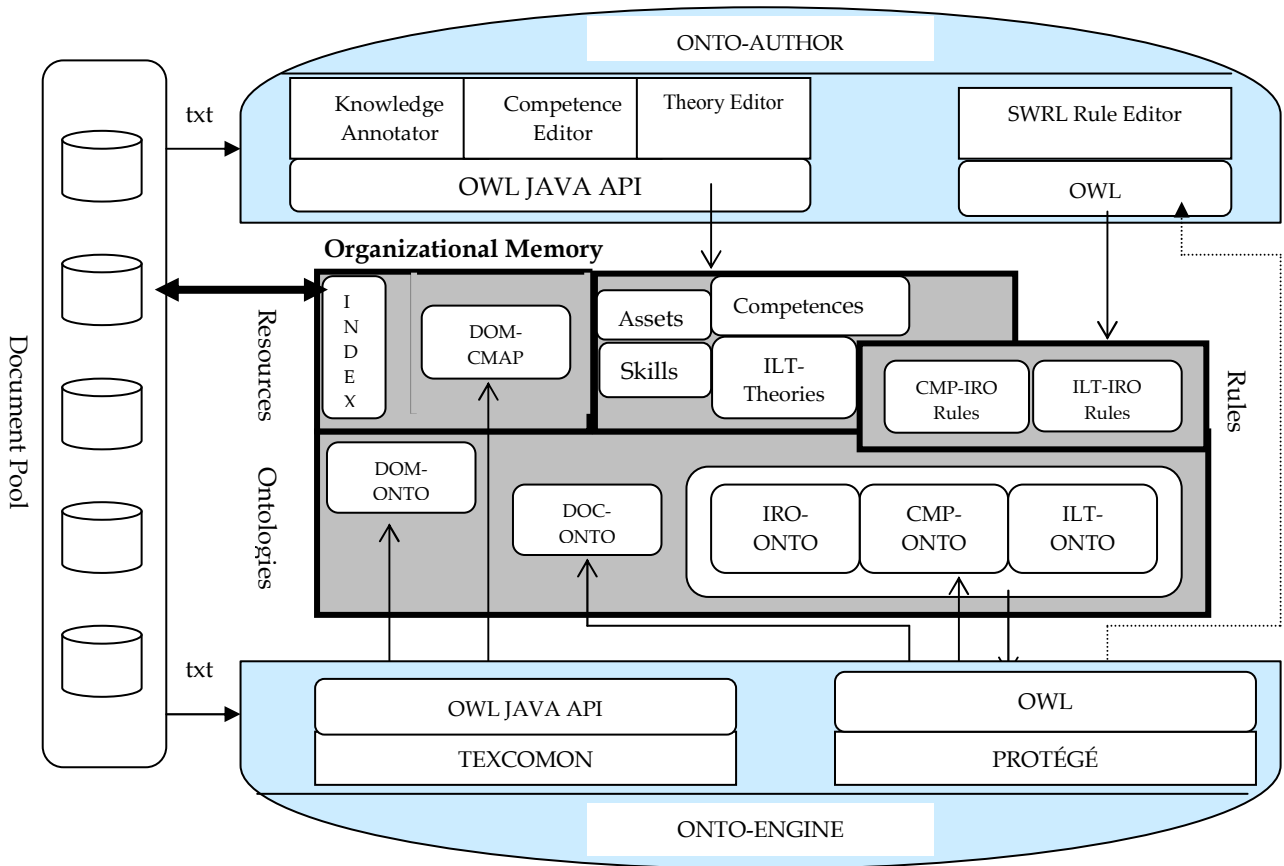


Fig. 1. The Knowledge Production Component Architecture

In general, OMs serve to store, maintain and reuse various chunks of knowledge in corporate and organizational contexts. As such, they constitute an interesting mean for what has become a necessity in today's economy: the just in time just-enough learning aim.

We believe that learning object repositories (LORs) are not well suited for this concept. In fact, the just in time just-enough learning suggests that learning content should be set up right in time (meaning at the moment when there is a need of bridging a competence gap) and with the right content (meaning that this content should be computed on the fly to fulfill the training need). In LORs, whole packaged LOs are stored regardless of the context in which they will be used. Composing a learning content at this level of granularity is not effective. Furthermore, LOs in LORs do not deal with adaptation according to a given learner model or according to specific competence needs. OMs represent a good alternative to solve the issue of granularity and adaptation. They are normally used in the industrial world to model the organizational knowledge [14], [44]; however, a recent work has also proposed the use of an OM in eLearning [1], [3]. Contrary to the work presented in [1], the Knowledge Puzzle framework uses text mining techniques for learning the organizational memory domain knowledge. Moreover, it enlarges the vision of an OM to enclose pedagogical knowledge and instructional theories.

Compared with existing approaches to learning object composition, the interest of this layered architecture relies on three main points:

First, the whole framework is guided by the ontological layer which follows the criteria stated by [41] for implementing eLearning on the semantic web: structure, content and pedagogy. The ontologies enable interoperability and understanding among various training systems.

Second, one of the main goals of our project is to help integrate the representations used by intelligent tutoring systems in learning objects. These representations describe the domain knowledge, the tutor model (pedagogical expertise) and the learner model. This allows us to consider a learning object not as a static entity, but as a mini intelligent tutoring system. In this way, a learning object, at the time of its generation, can be adapted for a given student, according to a given educational theory, and possesses a model of the domain knowledge it covers. To our knowledge, none of the existing approaches integrates these various models to produce eLearning resources.

Finally, even if some works advocated the use of one or more of the models (domain, instructional roles, competence) to describe or index the learning objects [15], [41], none of the approaches implemented the whole process of stating learning objectives, generating a domain ontology, and stating instructional theories to guide the learning object composition, nor did they propose a clear definition of what should be a learning object in this semantic web era.

The OM layers and components are described below.

3.1. The Document Pool

Existing learning resources constitute the document pool in order to be re-factored, indexed and capitalized. At this stage of the project, only plain text documents can be automatically processed and they should be related to the same domain knowledge in order to produce a consistent ontology. They should also contain declarative knowledge about the domain.

3.2. The Ontology Layer

The Ontology Layer is the pillar of the OM. It structures the different kinds of resources that are needed to compose the Learning Knowledge Objects (LKO). In fact, besides the domain ontology (that will be generated from learning objects' content), a number of ontologies are necessary to establish the Knowledge Puzzle integrated model. The ontology layer includes the following ontologies:

- *The document Ontology (DOC-ONTO)* creates a document index for each learning object or document. This index serves to store the LO's structure in term of paragraphs, sentences, figures, tables, etc.

- *The domain ontology (DOM-ONTO)* models the domain knowledge and is exported from the domain concept maps (DOM CMAP). The interest of using concept maps as intermediate structures lies in their power to effectively describe and index domain content and the necessity to develop bridges from less formal representations to more formal ones.

- *The Instructional Role Ontology (IRO-ONTO)* defines a number of instructional roles (Definition, Introduction ...) that guide the detection of relevant assets within learning materials. Fig. 2 shows an excerpt of the different instructional roles. These instructional roles serve as the basic knowledge fragments for the composition of the LKOs.

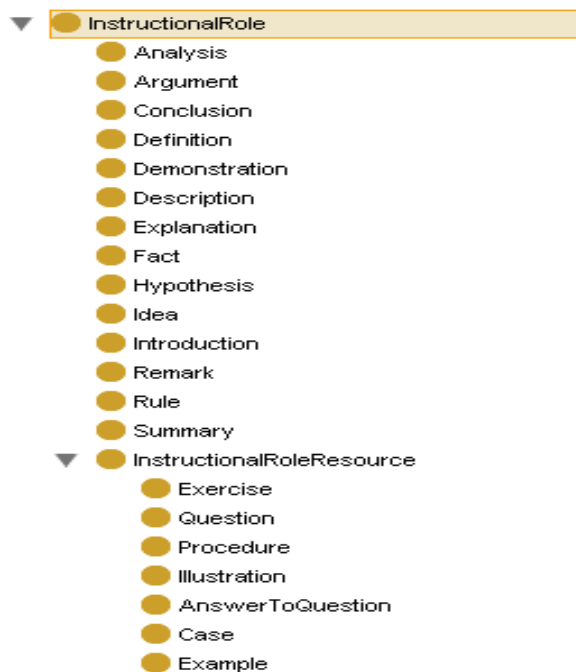


Fig. 2. An Excerpt of the Instructional Role Ontology

- *The competence Ontology (CMP-ONTO)* is expressed in term of skills on domain concepts. It is used to express the learning objective. We used the Bloom Skill's Taxonomy [7] as generic competence taxonomy. The following is an example of a rule that links a skill "define" to an instructional role "Definition":

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

This rule means that a definition is required to fulfill the skill "define".

- *The Instructional learning Theory Ontology (ILT-ONTO)* provides a pedagogical theoretical knowledge that guides the dynamic composition of the LKO. For the moment, ILT-ONTO is quite simple as it represents a pedagogical strategy as a number of instructional events that are linked to the resource layer through explicit SWRL (Semantic Web Rule Language) rules. However, it is possible to use a real instructional theory ontology (such as OMNIBUS [17]) to actually guide the composition process. The current instructional events in ILT-ONTO are derived from theories of education such as Gagné [13], Merrill [27], etc.

The Protégé Ontology Editor [36] serves for the definition of these specific ontologies. Since the OM is aimed at producing learning resources, each of the presented ontologies fulfills a necessary part of the framework. Inspired from the traditional intelligent tutoring system architecture, the OM represents the domain model (DOM-ONTO, IRO-ONTO, DOC-ONTO), the learner model through the definition of competences (CMP-ONTO) and the storage of acquired competences and abilities and the tutor model through the instructional theories (ILT-ONTO).

3.3. The Resource Layer

The Resource Layer stores the various assets obtained through semantic annotation as well as the domain concept maps (DOM-CMAP), the skills, the competences and the instructional theories. In general, it is constituted from the different ontology instances (A-Box).

3.4. The Rule Layer

The Rule Layer, expressed in SWRL format, represents procedural knowledge that acts as a glue between the different ontologies. Competences and instructional roles are linked through CMP-IRO Rules, whereas instructional learning theories and instructional roles are connected through ILT-IRO Rules. CMP-IRO rules serve to find the instructional roles able to fulfill a specific skill need (define, analyze, explain ...) and ILT-IRO rules are used to identify the instructional roles needed for each instructional event of the theory.

Now that the OM structure is presented, the next section provides details about how its content is obtained.

4. THE KNOWLEDGE PUZZLE PRODUCTION SUBSYSTEM

The Knowledge Puzzle architecture includes two subsystems: the first one, the production subsystem, enables the constitution of the OM's elements and the second one

their exploitation. The production subsystem is made up of two major tool suites: ONTO-AUTHOR and ONTO-ENGINE. The latter simply includes TEXCOMON and the Protégé Ontology Environment [36]. The Protégé OWL Java API is employed as the communication language between the tools and the ontologies.

The first component, TEXCOMON, exploits pattern matching techniques to learn the domain ontology from the document pool.

4.1. Capitalizing Learning Objects through an Innovative Ontology Engineering Approach (TEXCOMON)

The most important tool within the ONTO-ENGINE system is TEXCOMON. ONTO-ENGINE is simply a framework that includes both TEXCOMON and Protégé and uses the Protégé OWL Java API for communication purposes.

TEXCOMON stands for (TEXT-CONcept Maps-ONtology) to indicate the process followed in order to convert texts into domain concept maps, which are in turn transformed into an OWL ontology. This ontology represents the implicit domain knowledge contained in the learning objects, which is currently not accessible to training environments.

Briefly speaking, the ontology engineering process is as follows: Textual learning objects are taken as inputs, and an index structure is created. This index structure decomposes each document into paragraphs and sentences using UIMA-based annotators (Unstructured Information Management Architecture) [42]. Other structural annotations can be performed to manually identify figures, tables, etc.

A Keyword extraction algorithm [12] is then executed in order to retrieve document key words and key sentences. These sentences are parsed through the Stanford Statistical Parser [24] that outputs a typed dependency network [11] for each sentence. In TEXCOMON, the typed dependency network is called a grammatical concept map.

Once these grammatical structures are available, they are mined to find instances of lexicon-syntactic patterns that we defined in a linguistic knowledge base (made up of around 20 patterns for the moment). The lexicon-syntactic patterns allow identifying grammatical subgraphs that can be transformed to obtain semantic representations.

The following table (table 1) shows examples of lexicon-syntactic patterns and their semantic transformation methods. The table depicts the transformation process from grammatical links to semantic links.

Let t, u and v be domain terms linked by grammatical links (input and output links).

Table 1. Lexicon-syntactic patterns and their semantic transformation methods

Input Links (t)	Output Links(t)	Method
-	NOMI-NAL_SUBJECT (nsubj) with u as destination, DIRECT_OBJECT (dobj) with v as destination.	Create a new relationship between u and v labelled with t.
-	NOMI-NAL_SUBJECT (nsubj) with u as destination, COPULA (cop) with v as destination.	Create a new relationship between u and t labelled v.
Rcmmod (relative clause modifier)	Passive Nominal Subject (nsubjpass) with u as destination, Preposition (prep_) with v as destination	-Ignore u because it is a relative pronoun -s=Find subject (t) -newLabel = concatenate t and the preposition prep - Create a new relationship between s and v labelled newLabel.

A pattern is defined as a data structure composed of input and output grammatical links organized around a given term t. These links constitute the syntactic structure that should be fetched in each key sentence. Each detected configuration triggers a method to obtain a sentence concept map.

Formalizing such patterns in a linguistic knowledge base allows for progressively identifying grammatical structures and their semantic possible interpretations. However adding new patterns does not imply changing the underlying framework thus making it flexible and extensible. Domain-independent linguistic knowledge bases can then be re-used in other contexts. All the semantic maps are merged in order to create Domain Concept Maps (DOM-CMAP) around concepts. TEXCOMON integrates the different semantic representations (relationships and concepts) around a given domain term. Hence one concept map can be built from different documents. Fig. 3 depicts the domain knowledge acquisition process in TEXCOMON.

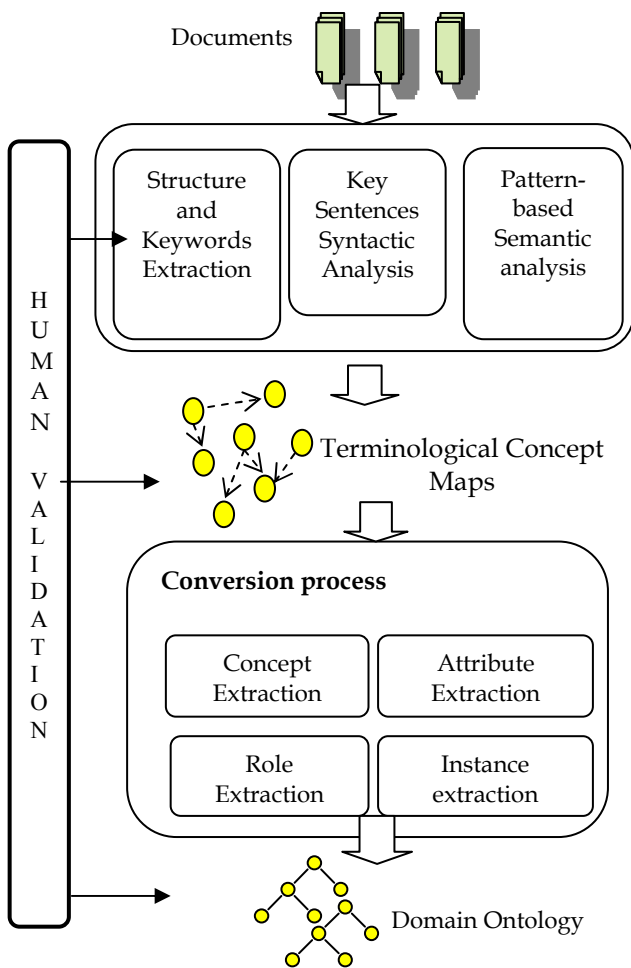


Fig. 3. The Domain Knowledge Acquisition Process in TEXCOMON [47]

A Domain Ontology (DOM-ONTO) is created from the available concept maps by detecting ontological concepts and relationships. Since the process of learning a domain ontology follows a set of learning tasks defined in [9], TEXCOMON implements these various tasks by determining significant concepts, attributes, relationships (hierarchical and conceptual) and instances. The significance of a concept is manually defined by stating a threshold representing the out-degree of a given concept. This states that a concept is considered as ontological when it has a significant number of relationships with others. The relationships linking significant concepts are also considered as ontological.

Other patterns are used to determine hierarchical links, instances and attributes. TEXCOMON encodes Hearst's patterns [18] for discovering hyponyms and instance relationships. In fact, differentiating sub-classes from instances is a difficult problem. In TEXCOMON, when a hyponym link is detected between two domain concepts, then a subclass link is created. If the hyponym is detected between a concept and one or more domain terms (that were not sufficiently connected to be considered as concepts) then the domain terms are considered as concept's instances. Similarly, attributes are detected by some patterns indicating them [35]:

- « X of Y » =>...the ID of the learner ...: The learner

has ID

- « X' Y » => ...learners' ID...: Learners have ID
- « X's Y » => ...Student's last name...: Student has last name

Finally an OWL file is generated that represents the domain ontology DOM-ONTO. More details about this process can be found in [45].

4.2. Capitalizing Learning Objects through Semantic Annotation and Edition (ONTO_AUTHOR)

Contrary to Polsani's definition of learning object, we believe that it is not the learning object itself that should be reused in multiple instructional contexts, but the instructional resources that compose the learning object. This necessitates that these resources are clearly identified. ONTO-AUTHOR is an authoring environment that uses the ontologies to define and extract the resources layer of the OM. ONTO_AUTHOR includes a variety of tools.

According to Polsani [34] "The formal composition of a LO is the arrangement of elements. An element could be text, image, video, animation, glossary, assessment, multimedia, etc. Preferably a LO should be a combination of multiple elements. The multiplicity not only reinforces the concept communicated, but it also opens up multiple avenues to foster a richer understanding of the idea(s) represented, facilitating learning based on learners' choices and learning characteristics. ". The Instructional Role Annotator is a semantic annotation tool dedicated to the annotation of instructional role instances in a document (or a learning object). This annotation aims at producing assets that will be used for LKO composition. Assets are very fine-grained knowledge blocks that confer a very high flexibility to the LKO composition process. They are elements (in the meaning stated by Polsani) that have a pedagogical function. For example, a text fragment can be a definition of a concept X. This definition can be reused in multiple training contexts that are linked to this concept X. Fig. 4 shows the Instructional Role Annotator Tool.

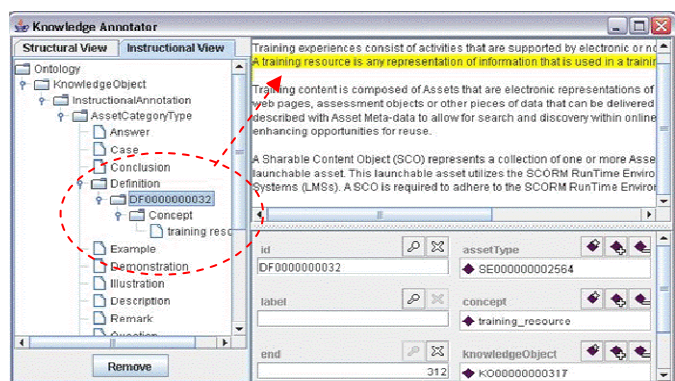


Fig. 4. The Instructional Role Annotator

The competence editor defines competences as skills that are linked to domain ontology concepts through an OWLObjectProperty "Concept". There is a debate on whether competence ontologies should also model the domain knowledge or if they should only take into ac-

count competence levels terminology leaving the domain knowledge in a separate ontology. We believe that keeping both ontologies separate from each other's fosters the reusability of the competence ontology and enables the graceful migration or evolution of the domain knowledge in an independent manner.

The SWRL Rule Editor serves to define the memory rule layer. Protégé already provides one editor of this kind [33].

Finally, the theory editor enables to create a learning theory and to link it to a set of instructional events represented as instances of the class "InstructionalEvent". For example, Gagné's theory states that the first instructional event should be to gain learner's attention. The instructional events of a given theory are then associated to SWRL rules in order to refer to OM's assets. For instance, a flash animation about the targeted concept can be provided to gain the learner attention.

A new theory is created by stating its instructional

events and the rules associated to each one of them. For the moment, the tool is quite simple but further improvements will enable the integration of more complex concepts such as learning conditions and other types of constraints that can occur in the learning process.

5. THE KNOWLEDGE PUZZLE EXPLOITATION SUBSYSTEM

As previously mentioned, the OM is used as a knowledge base that sustains the dynamic aggregation of learning knowledge objects. The OM feeds the Knowledge Puzzle Exploitation process that is composed basically of three layers: the composition layer, the standardization layer and the deployment layer. Fig. 5 shows the Knowledge Exploitation Process.

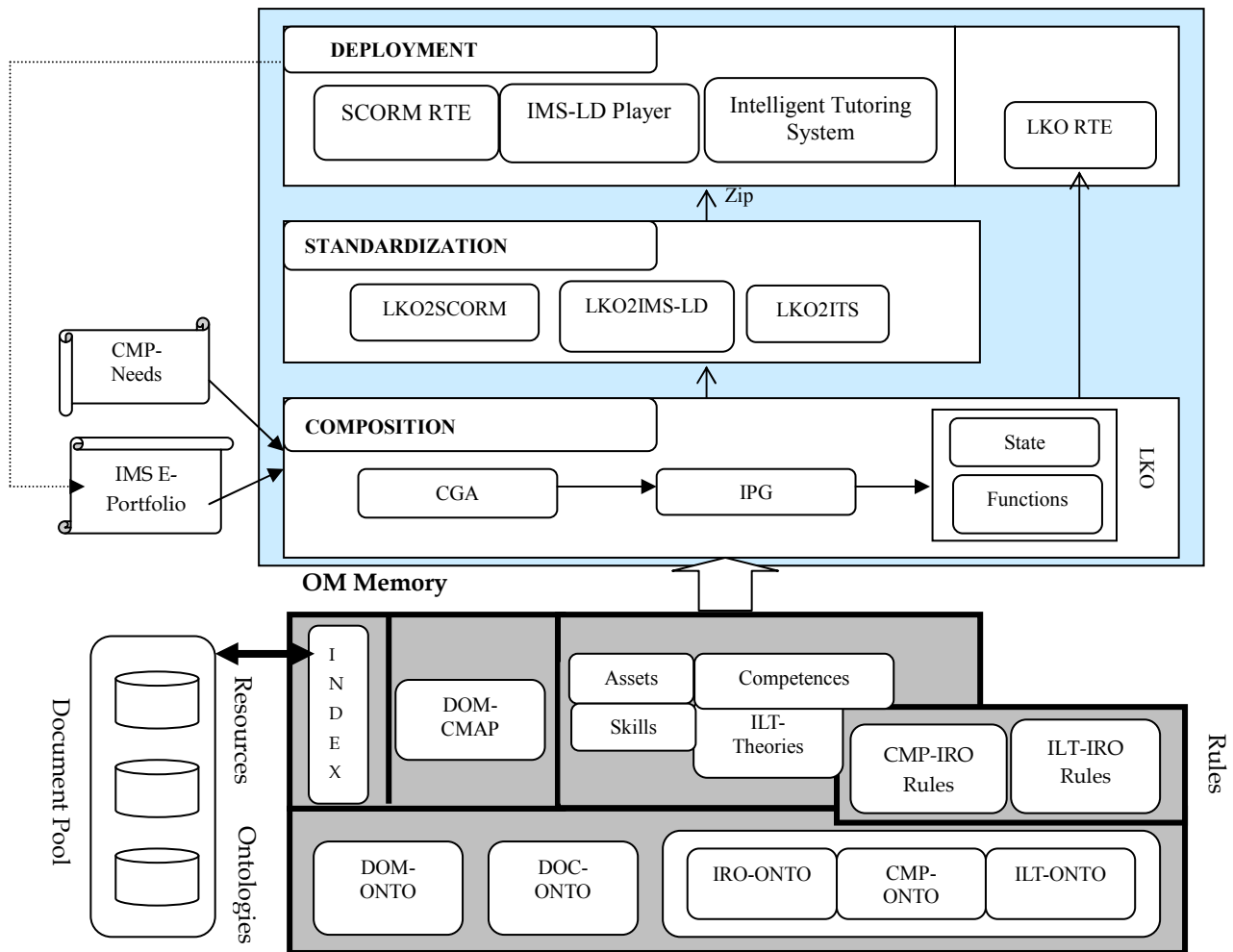


Fig. 5. The Knowledge Exploitation Component Architecture

5.1. Composition Layer

This layer enables to initiate the LKO's aggregation process starting from competence needs for a specific learner profile. Competence needs are expressed as an OWL file that describes the skills to acquire and the do-

main concepts that are concerned. The learner profile is expressed as an IMS ePortfolio [20] and converted in OWL format before being used by the composition subsystem. The Learner profile serves to store mastered competencies as well as learner's characteristics. It is an

overlay model meaning that the learner knowledge is progressively built and compared to an expert knowledge until it reaches the expert level.

When learning objectives are specified, a Competence Gap Analyzer (CGA) computes an adjusted competence definition tailored to the learner's profile. In fact, for each skill in the targeted competence, the learner profile is fetched in order to find out if this skill is already mastered. If this is the case then the skill is removed from the competence definition. The CGA then outputs a set of skills that are new to the learner in the form of an Adjusted Competence.

The Adjusted Competence is passed to an Instructional Plan Generator (IPG) that is in charge of composing the actual Learning Knowledge Object. The IPG exploits all the OM's layers:

- The resource layer offers the assets that represent the basic knowledge units in an LKO.

- The ontology layer, and more particularly, the Instructional Learning Theory Ontology (ILT-ONTO) describes the instructional events and conditions that will effectively guide the composition. A specific instructional theory must then be chosen (Merrill, [27], Gagné [13], Ad-Hoc...). This enables a flexible independent way for theory incorporation in Learning Knowledge Object. It also offers an explicit pedagogical framework understandable by humans and software.

- The Rule Layer is used to connect a particular competence and skill with the more appropriate assets able to fulfill it. For example, the skill "define" would require a "Definition". It also connects the Instructional learning events with the appropriate assets. The following rule is an example which links the instructional event "Insure_retention" to the instructional role "Summary":

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

The execution of the IPG produces a Learning Knowledge Object that can be seen as an independent object composed of a Data State and of a set of functions to manipulate it (an interface).

The data state is an OWL data structure formed from the various resources necessary for the LKO: the competence, the skills, the domain ontology around the concepts targeted by the skills as well as the domain concept maps and the learner profile. The LKO functions are a sort of a standard interface that enables any LKO to act as a small "Intelligent Tutoring System".

This standard interface offers the following functions:

- Scenario Control to guide the learner's progression through the LKO content;
- Evaluation of learner's actions and exercises;
- Learner's e-Portfolio update;
- Domain Ontology and Concept Maps Exploration;
- Explanation of different concepts by their context;
- Automatic Generation of LKOs related to the concept's context.

The added value of the LKO when compared to existing learning objects relies on its inner characteristics: an LKO acts as an independent small tutoring system, it has a domain model, it is guided by an instructional theory,

and it possesses an interface to act on its data and to provide an individualized training. An LKO does not need to be stored as a whole in a learning object repository. Since the aim of the organizational memory is to conserve only sound reusable pedagogical fragments, and not whole learning objects, an LKO can be generated by using these resources and a particular theory. In our point of view, learning objects must not exist as fixed static content packages. Semantic services coupled with instructional roles should be the new paradigms that sustain learning object generation. To summarize:

- An LKO is an independent object: it is implemented as a software package (an applet) that receives, at runtime, an on-the-fly generated OWL file able to fulfill a specific competence for a specific learner.

- An LKO possesses pedagogical knowledge in the form of a scenario plan generated according to an instructional theory. This theory (Gagné, Merrill, etc.) is also chosen right at the time of the aggregation process. The same content can be reused to produce another LKO compliant with another instructional theory. So the reusability dimension is not really related to the LKO itself but to the independent knowledge fragments (the instructional roles) that are stored in the OM. The composition service can then generate the same LKO as often as needed.

An example of a competence could be: "define Shareable Content Object". In the following figure (Fig. 6), we can see the generated LKO for this competence according to the Gagné Theory. We can also notice that two prerequisites are added to the LKO related to the skills "define asset" and "define media". Hence the resulting LKO is composed of three skills, each skill being taught according to the Gagné Theory. These prerequisites are added to the definition of the competence during the competence gap analysis. Once the adjusted competence is available, and once an instructional theory is chosen, the IPG loads the instructional events related to the theory (here the Gagné Theory). Since each instructional event is linked to a SWRL rule, the IPG executes these rules in order to gather the different assets that are required to fulfill each instructional event. The rule engine Jess is used to run the SWRL rules. As a result, a course structure compliant with the chosen theory is generated and linked to adequate instructional roles.

5.2. Standardization Layer

The standardization layer serves as an interface to the different standards and usable environments for an LKO. It comprises:

- A SCORM LKO Generator (LKO2SCORM) that generates a SCORM compliant content package. It encapsulates the LKO applet into a standard SCORM template.
- An IMS-LD LKO Generator (LKO2IMS-LD) that generates an IMS-LD conformant content package.
- An Intelligent Tutoring System LKO Generator (LKO2ITS).

The standardization layer exports, when needed, a zip file to any type of training environment.

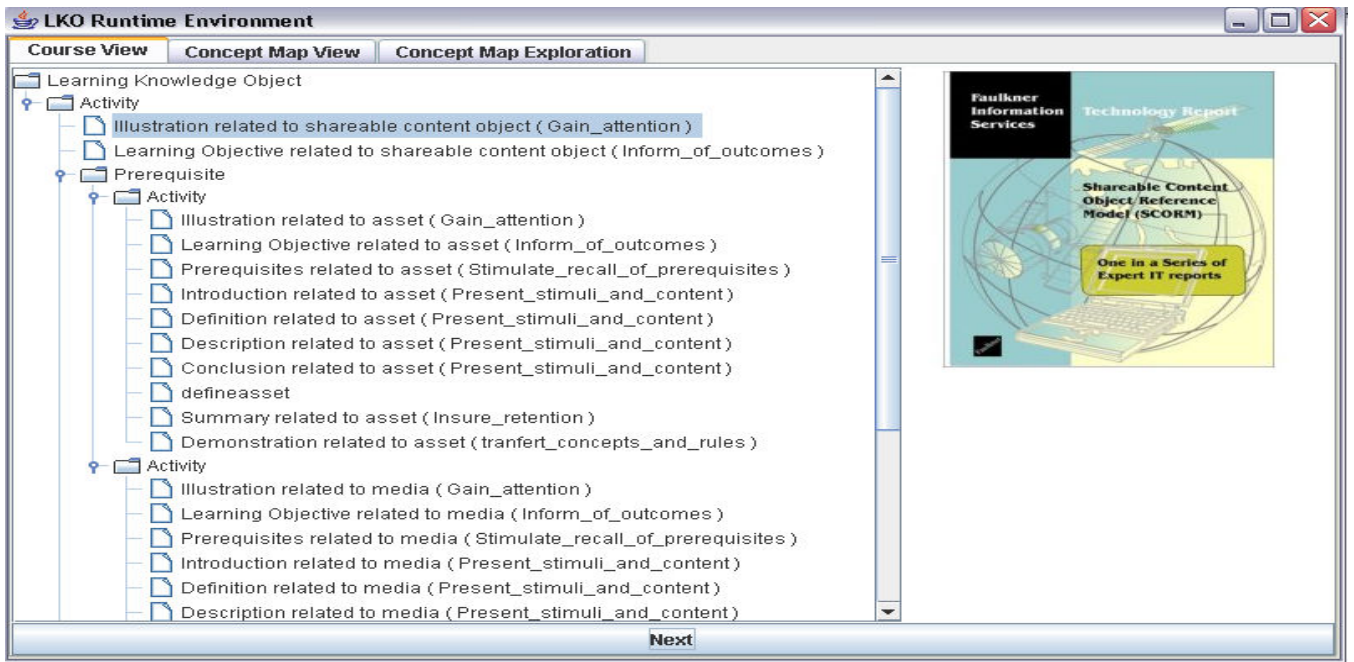


Fig. 6. A Generated Learning Knowledge Object according to Gagné’s theory

5.3. Deployment Layer

The LKOs are targeted towards any kind of training environment. The Knowledge Puzzle produces LKOs deployable in a SCORM Runtime Environment, in an IMDL player or in any intelligent tutoring system.

We also provide an LKO Runtime Environment (LKO-RTE) for users that do not have access to an ITS or that do not want to comply with a particular e-Learning stan-

dard. The LKO-RTE makes it possible to run the LKO as a standalone resource. The user interface gives access to relevant functions that are implemented within the LKO to support a variety of learning services including possible access and exploration of the concept map around the LKO’s concepts (Fig. 7), thus fostering meaningful learning [32].

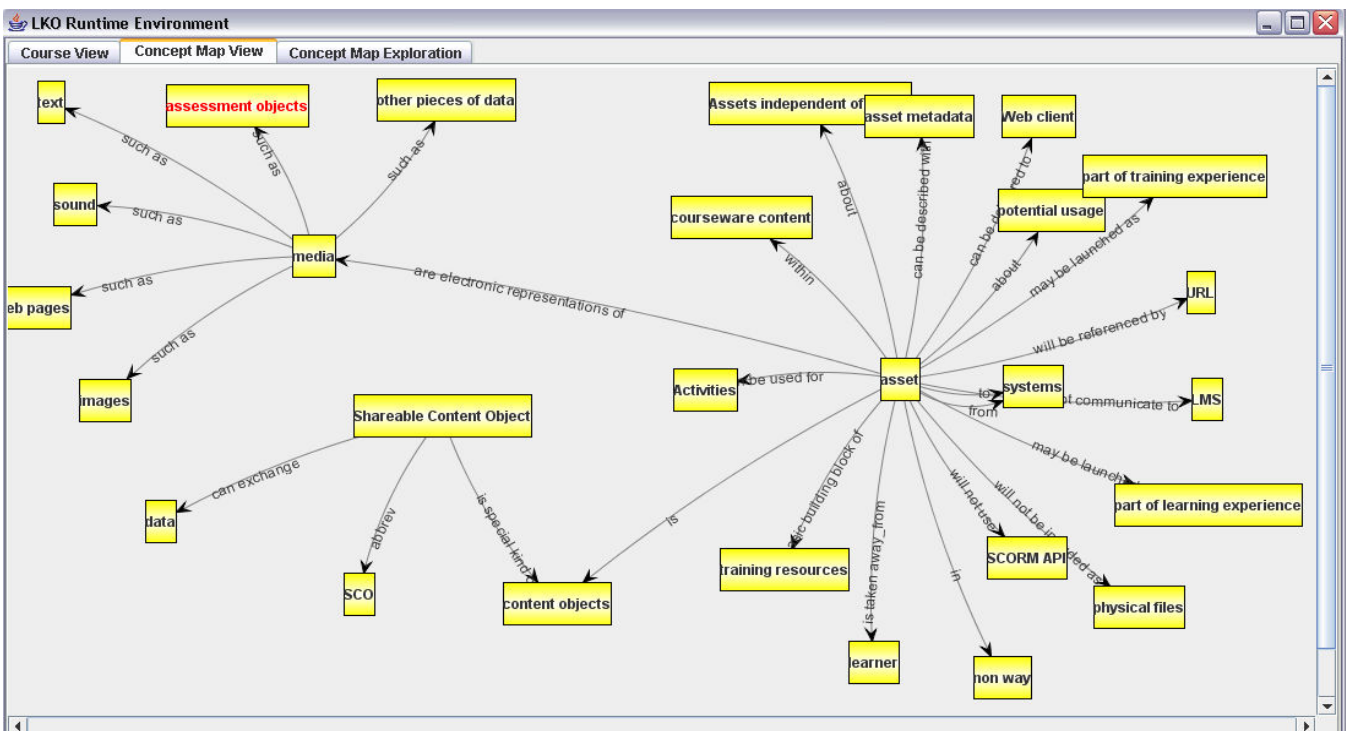


Fig. 7. The concept Map View in the current LKO

6. EVALUATION AND RESULTS

Multiple dimensions must be evaluated in a project such as the Knowledge Puzzle: The first and most important one involves the quality of the generated domain ontology. We performed a qualitative expert evaluation (a Semantic evaluation) to assess the plausibility and comprehensiveness of the generated concepts and relationships.

6.1. Experiment Description

6.1.1. Corpus

We applied the Knowledge Puzzle reverse engineering approach on a corpus composed from a set of 36 documents about the Sharable Content Object Reference Model (SCORM). These documents were created from the official SCORM Manuals available on [2]. In total, the documents had 188 paragraphs and 1578 sentences comprising 29879 words.

Two human experts performed the semantic evaluation. These experts had an extensive experience in eLearning standards. One expert has been involved in standardization groups and teaching about eLearning standards while the other one has been working on the development of a SCORM platform.

Given a set of triples generated by the Knowledge Puzzle from the ontological relationships, the experts were asked to decide:

- If the concepts were relevant according to the domain knowledge;
- If the relationships were correct in term of label and arguments;
- If the relationships were relevant according to the domain knowledge;

Each expert was asked to look at the domain ontology in the Protégé Ontology Editor [36] and to delete the inappropriate concepts and relationships. A percentage of pertinent concepts and relationships were then calculated by comparing the generated domain ontology and the expert validated ontology. The mean of the two percentages was considered as the final result.

6.1.2. Description of the Experiment with *TEXCOMON*

TEXCOMON extracted a set of 1,139 domain terms during the concept maps creation as well as 1,973 semantic relationships. From these domain terms and relationships, 4 domain ontologies corresponding to different thresholds were generated. As previously explained, a threshold I is specified to consider a domain term X as a domain concept. I is the number of semantic relationships (the out-degree of X) where X is the source concept.

In this experiment, we considered 4 thresholds I :

- $I=2$ which outputs the ontology KP-2
- $I=4$ which outputs the ontology KP-4
- $I=6$ which outputs the ontology KP-6
- $I=8$ which outputs the ontology KP-8

6.1.3. Description of the Experiment with *Text-To-Onto*

For comparison reasons, we performed another semantic evaluation over the same corpus with another major work in the field of ontology learning from text: *Text-To-Onto* [26]. Different steps are used in *Text-To-Onto* to learn the domain ontology:

First, term extraction is performed: we kept all the extracted domain terms regardless of their frequency, c -value, etc. In fact, *Text-To-Onto* provides a pruner that suggests concepts which could be removed from the ontology on the basis of their occurrence in the corpus. However, the pruner actually suggested pruning concepts that should not be removed from the resulting ontology and that would get the results worse. We believe that this is because *Text-To-Onto* relies only on statistical features (cumulative frequency) to prune some concepts and tends to keep only statistically significant concepts (while statistically non-significant concepts could also be important) [47]. Apart from that, there is no notion of filtering related to the out-degree of a node in *Text-To-Onto*, which prevented us from using the exact procedure used for *TEXCOMON* ontologies.

Second, hierarchical links are extracted through the *TaxoBuilder* tool (we used the n most frequent words option + the combination-based approach built on Hearst patterns and heuristics).

Third, relationships between domain terms are fetched through relation learning and association rule learning. Relation learning outputs a set of labelled links between the concepts much like *TEXCOMON* does. Association rule learning aims to discover frequently co-occurring items within a data set and to extract rules that relate these items. Two ontologies (TTO1 and TTO2) are generated from the corpus with *Text-To-Onto*. The main difference between these two ontologies resides in the use of a different support in association rule learning (respectively 0 and 0.1). The support of association rules equals the percentage of groups that contain all of the items listed in such association rules [47]. Therefore, a support of 0.1 for a given item means that the item occurs in 10% of all transactions. Here the items are represented by co-occurring words.

6.2. Semantic Evaluation Result

6.2.1. *TEXCOMON* Semantic Evaluation

Table 2 shows the number of generated concepts and relationships from the initial set of domain terms and relationships. Here primitive classes mean classes with necessary conditions while defined classes mean classes with necessary and sufficient conditions. In fact, the only defined classes that *TEXCOMON* was able to extract were equivalent class axioms. An equivalence relationship is stated mainly between a concept and the abbreviation used to designate this concept (for example SCORM = Shareable Content Object Reference Model).

Table 2. Number of concepts and relationships in each generated domain ontology in TEXCOMON

Ontology	Number of primitive classes	Number of defined classes	Number of hierarchical relationships	Number of conceptual relationships
KP2	413	9	372	288
KP4	139	10	125	153
KP6	82	10	84	103
KP8	57	10	66	74

The two experts were asked to eliminate the concepts and relationships considered as too vague, such as "section", "example" and to conserve only the concepts and relationships relevant to the domain. The following two tables (Table 3 and Table 4) show pertinence rates according to both experts.

Table 3. TEXCOMON concepts and relationships pertinence according to Expert 1

Ontology	Pertinent primitive Classes (%)	Pertinent defined Classes (%)	Pertinent hierarchical relationships (%)	Pertinent conceptual Relationships (%)
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

Table 4. TEXCOMON concepts and relationships pertinence according to Expert 2

Ontology	Pertinent primitive Classes (%)	Pertinent defined Classes (%)	Pertinent hierarchical relationships (%)	Pertinent conceptual Relationships (%)
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

Finally, Table 5 gives the mean score of the ontologies according to both experts.

Table 5. TEXCOMON Overall Evaluation

Ontology	Pertinent primitive Classes (%)	Pertinent defined Classes (%)	Pertinent hierarchical relationships (%)	Pertinent conceptual relationships (%)
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

Given the criteria described above, the primary results of the semantic analysis were quite satisfying. According

to human evaluators, we reached, in the worst case, a pertinence rate of 86.65% for primitive classes and a rate of 80.08% for pertinent conceptual relationships.

However, we must underline that many relationships were repetitive due to a lack of synonym detection in TEXCOMON (for instance: "a SCO is launched in RTE" and "a SCO is deployed in RTE" are considered as two relationships) and the lack of a correspondence between the active and the passive voice (For example: "a SCO is launched in RTE" and "RTE launches SCO" should normally be detected as an inverse property).

6.2.2. Text-to-Onto Semantic Evaluation

The experts repeated the same evaluation over Text-To-Onto ontologies. The first thing to remember is that Text-To-Onto has two ways to learn conceptual relationships: association rule learning which outputs non-labelled relationships and relation learning which relies on linguistic patterns and outputs labelled relationships.

Table 6 indicates the number of generated relationships in both cases. Table 7 summarizes the number of classes and relationships generated in Text-To-Onto. The only difference between TTO1 and TTO2 lies in the number of conceptual relationships which changes according to the support used in association rule learning. Please note that no unlabeled conceptual relationships were extracted in TTO-2 due to the support of 0.1.

Table 6. Number of generated relationships in Text-To-Onto

Ontology	Conceptual relationships with labels « Relation Learning »	Conceptual relationships without labels « Association Rule Learning »
TTO1	33	5650
TTO2	33	0

Table 7. Number of concepts and relationships in Text-To-Onto ontologies

Ontology	Pertinent primitive Classes (%)	Pertinent defined Classes (%)	Pertinent hierarchical relationships (%)	Pertinent conceptual Relationships (%)
TTO1	336	0	223	5683
TTO2	336	0	223	33

In fact, we realized with our experiment that even a support of 0.1 (which is low) was discarding all the association rules generated by Text-To-Onto. In fact, TTO-2 shows an important disparity of results in comparison with TTO-1. TTO-1 contains a lot of properties that mean nothing and that are not pertinent to the domain. The following tables (Table 8 and 9) summarizes the results with Text-To-Onto.

Table 8. Concepts and relationships pertinence according to Expert 1

Ontology	Pertinent primitive Classes (%)	Pertinent hierarchical relationships (%)	Pertinent conceptual relationships (%)
TTO1	72.02	58.74	0.3
TTO2	72.02	58.74	51.51%

Table 9. Concepts and relationships pertinence according to Expert 2

Ontology	Pertinent primitive Classes (%)	Pertinent hierarchical relationships (%)	Pertinent conceptual relationships (%)
TTO1	74.10	36.32	0.32
TTO2	74.10	36.32	54.55

Table 10 reflects the average relevance rate arising from the dual evaluation described above.

Table 10. Average relevance rate for concepts and relationships in Text-To-Onto

Ontology	Pertinent primitive Classes (%)	Pertinent hierarchical relationships (%)	Pertinent conceptual relationships (%)
TTO1	73.06	47.53	0.31
TTO2	73.06	47.53	53.03

Among the 336 concepts, only 73.06 % were considered as valid concepts. Only 33 labelled relationships with specified domain and range were learned with the relation learning tool among which only 17 were considered as valid. Finally, association rule learning output 5650 associations which were all discarded.

6.3. Result Analysis

With Text-To-Onto, we can notice the following facts: association rule learning is not satisfactory: the extracted relationships have no label, and no real meaning, which is problematic. According to the previous remarks, the major weakness of Text-To-Onto remains the very small number of conceptual relationships. The second weakness lies, in the opposite, on the very important number of association rules, which is too noisy. Moreover, when generating owl classes, Text-To-Onto creates two classes for a concept (e.g. *aggregation*) and its stem (e.g. *aggreg*), whereas both of them should refer to the same concept.

In general, TEXCOMON gives better results than Text-To-Onto in both concept and relationship learning as shown in the previous tables. Its major strength is in the conceptual relationship learning. However, TEXCOMON takes a longer time to process the corpus than Text-To-Onto (which is very quick).

Finally, we must underline that text mining tools must not only be judged on their extracted knowledge but they must also be evaluated according to the missing knowledge (Knowledge available in the corpus but not in the

generated ontology). Both TEXCOMON and Text-To-Onto must be improved to find more knowledge from text. We are also in the process of assessing more thoroughly the domain ontology by comparing the resulting ontologies in term of structural and comparative characteristics, leading to an evaluation methodology [46].

The second evaluation dimension involves the quality of the learning knowledge object themselves and the added value of incorporating instructional roles and instructional theories in their composition. In fact, research works have already proven the interest of the pedagogical dimension in learning objects [8], [43]. The debate in the e-Learning community is rather on how and when we should include pedagogy in learning objects rather than on whether or not we should include it. The principal preoccupation of the community is to conserve the reusability of learning objects. Thus constraining them to one pedagogical theory seems unfruitful. The interest of our approach is that a Learning Knowledge Object is constrained to a theory only at generation time. The theory is only a parameter in the LKO composition. Since content and pedagogy are clearly separated, it is possible to generate another LKO based on the same content but with another theory, thus solving the issue about reusability versus pedagogy.

7. CONCLUSION

We presented a platform, the Knowledge Puzzle Project, which enables to capitalize existing learning objects with the creation of an ontology-based organizational memory. The organizational memory concept, as presented in this paper, represents a new perspective for the e-learning field as it is founded on a rather different idea: learning objects as content packages must not exist. Instead, assets, i.e. small fine-grained instructional units, can be exploited by composition mechanisms and learning services in order to aggregate Learning Knowledge Objects to fulfill specific training needs. This vision must be sustained by an ontological structure that represents the different necessary knowledge types: the domain knowledge, the instructional knowledge, the instructional learning theories and the competence model.

We showed how the Knowledge Puzzle production subsystem enables the automatic domain ontology generation from learning object content through TEXCOMON, and how annotation and edition of assets, rules, competences and instructional roles are performed through ONTO-AUTHOR.

This gives an OM's content that the Knowledge Puzzle Exploitation Subsystem can exploit for the on-the-fly generation of active, independent, reusable and theory-aware learning knowledge objects. The LKOs can then be used in any training environment including standards platforms and intelligent tutoring systems. They can also be used as learning services or resources that provide access to rich knowledge structures.

Our solutions contrast with current practices in the eLearning area and represent a way to bridge the gap

between classical eLearning systems and intelligent tutoring systems. Furthermore, the LKO concept offers a great opportunity to go beyond the actual limited view of LOs by allowing the exploration of their content and making the integration of pedagogical knowledge more flexible in a given learning context.

We also performed a semantic evaluation based on human experts and we compared our tool TEXCOMON with Text-To-Onto, one of the state of the art tools in domain ontology learning from text. This evaluation showed improved extraction results with our tool.

Further work is needed to enhance both subsystems. In the production part, text mining and pattern extraction can be enhanced in order to decrease the generated noise. Furthermore, integration of more complex pedagogical scenarios and instructional theories [8] must be undertaken to enrich the composition process.

REFERENCES

- [1] M.-H. Abel, A. Benayache, D. Lenne, C. Moulin, C. Barry, and B. Chaput, "Ontology-based Organizational Memory for e-learning," *Journal of Educational Technology & Society*, 7 (4), 98-111.
- [2] ADL. Retrieved September 5, 2007, from <http://www.adlnet.org>
- [3] S. Amershi and C. Conati, "Automatic Recognition of Learner Groups in Exploratory Learning Environments," *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, LNCS No 4053, pp.463-472, Springer-Verlag, Berlin, 2006.
- [4] R. Baker, A. Corbett, K. Koedinger and I. Roll, "Generalizing detection of gaming the system across a tutoring curriculum," *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, LNCS No 4053, pp. 402-411, Springer-Verlag, Berlin, 2006.
- [5] A. Benayache, "Construction d'une mémoire organisationnelle de formation et évaluation dans un contexte e-learning : le projet MEMORae," Compiègne: Université de technologie de Compiègne, 2005.
- [6] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web," *Scientific American Magazine*, 5 (1): 34-43, 2001.
- [7] B. S. Bloom, "Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain", Longman, New York, 1956.
- [8] J. Bourdeau, R. Mizoguchi, V. Psyché and R. Nkambou, "Selecting theories in an ontology-based IIS authoring environment," *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, LNCS N 3220, pp.150-161, Springer-Verlag, Berlin, 2004.
- [9] P. Buitelaar, P. Cimiano, M. Grobelnik and M. Sintek, "Ontology Learning from Text," *Tutorial at ECML/PKDD 2005*, Porto, Portugal, 2005.
- [10] K. Cardinaels, M. Meire and E. Duval, "Automating metadata generation: The simple indexing interface," *Proceedings of the 14th International Conference on World Wide Web*, pp. 548 - 556, Chiba, Japan, 2005.
- [11] M.-C. De Marnette, B. MacCartney and C.D. Manning, "Generating Typed Dependency Parses from Phrase Structure Parses," *Proceedings of the 5th Conference on Language Resources and Evaluation*, pp. 449-454, Genoa, Italy, 2006.
- [12] E. Frank, G.W. Paynter, I. H. Witten, C. Gutwin, and C.G. Nevill-Manning, "Domain-specific key phrase extraction," *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pp. 668-673, San Francisco, USA, 1999.
- [13] R. M. Gagné, L. J. Briggs, and W. W. Wagner, "Principles of instructional design (4th ed.)," Fort Worth: HBJ College Publishers, 1992.
- [14] F. Gandon, "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 (EMCSR 2002)*, (pp. 623-628). Vienna, 2002.
- [15] D. Gasevic, J. Jovanovic and V. Devedzic, "Ontologies for Creating Learning Object Content," *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information & Engineering Systems*, pp. 284-291, Wellington: Springer, 2004.
- [16] S. Graf and R. Bekele, "Forming Heterogeneous Groups for Intelligent Collaborative Learning Systems with Ant Colony Optimization," *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, LNCS No 4053, pp. 217-226, Springer-Verlag, 2006.
- [17] Y. Hayashi, J. Bourdeau and R. Mizoguchi, "Ontological Support for a Theory-Eclectic Approach to Instructional and Learning Design," *Proceedings of the First European Conference on Technology Enhanced Learning (EC-TEL)*, pp. 155-169, 2006.
- [18] M. Hearst, "Automatic Acquisition of Hyponyms from Large Text Corpora," *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pp. 539-545, Nantes, 1992.
- [19] C. Heiner, R. Baker and K. Yacef (Editors), *Proceedings of Educational Data Mining Workshop Held in conjunction with the 8th International Conference on Intelligent Tutoring Systems*, Jhongli, Taiwan, 2006.
- [20] IMS ePortfolio Specification. Retrieved June 10, 2007 from www.imsglobal.org/ep/index.html
- [21] J. Jovanović, D. Gasevic and V. Devedzic, "Ontology-based automatic annotation of learning content," *International Journal on Semantic Web and Information Systems* 2(2): 91-119, 2006.
- [22] J. Kay, N. Maisonneuve, K. Yacef and O. Zaïane, "Mining Patterns of Events in Students' Teamwork Data," *Proceedings of the Workshop on Educational Data Mining Workshop, ITS'2006*, pp. 45-52, 2006.
- [23] K. Keenoy, A. Poulouvassilis, V. Christophides, P. Rigaux, G. Papamarkos, A. Magkanaraki, M. Stratakis, N. Spyrtatos and P.T. Wood, "Personalisation Services for Self E-learning Networks," *Proceedings of International Conference on Web Engineering*, LNCS No 3140, pp. 215-219, Springer-Verlag, Berlin, 2004.
- [24] D. Klein and C.D. Manning, "Accurate unlexicalized parsing," *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 423 - 430, Sapporo, Japan, 2003.
- [25] Y. Li and R. Huang, "Dynamic Composition of Curriculum for Personalized E-Learning," *Proceedings of the 14th International Conference on Computers in Education (ICCE2006)*, IOS Press, 2006.
- [26] A. Maedche and S. Staab, "Ontology Learning for the Semantic Web," *IEEE Intelligent Systems* 16 (2): 72-79, 2001.
- [27] M. D. Merrill, Instructional Transaction Theory (ITT): Instructional Design Based on Knowledge Objects, in Charles M. Reigeluth (Ed.). *Instructional Design Theories and Models: A New Paradigm of Instructional Technology*, Lawrence Erlbaum Associates, 1999.
- [28] R. Mizoguchi and J. Bourdeau, "Using Ontological Engineering to Overcome Common AIED Problems," *Journal of Artificial Intelligence and Education, Special Issue on AIED*, vol.11, pp. 107-121, 2000.
- [29] J. Mostow, J. Beck, H. Cen, A. Cuneo, E. Gouvea and C. Heiner, "An educational data mining tool to browse tutor-student interactions: Time will tell!," *Proceedings of the workshop on educational data mining*, pp. 15-22, 2005.
- [30] R. Navigli and P. Velardi, "Learning Domain Ontologies from Document Warehouses and Dedicated Websites," *Computational Linguistics*, 30(2):151-179, MIT Press, 2004.
- [31] R. Nkambou, E. Mephu Nguifo, O. Couturier and P. Fournier-Viger, "Problem-Solving Knowledge Mining from Users' Actions in an Intelligent Tutoring System," *Proceedings of the 19th Canadian Conference on Artificial Intelligence (AI'2007)*, Lecture Notes in Artificial Intelligence 3501, pp. 393-404, Springer-Verlag, Berlin, 2007.

- [32] J. D. Novak and A. J. Cañas, "The Theory Underlying Concept Maps and How to Construct Them," Florida Institute for Human and Machine Cognition. Pensacola, FL, USA: Florida Institute for Human and Machine Cognition, 2006.
- [33] M. J. O'Connor, H. Knublauch, S. W. Tu, B. Grosz, M. Dean, W. E. Grosso and M. A. Musen, "Supporting Rule System Interoperability on the Semantic Web with SWRL," *Proceedings of the 4th International Semantic Web Conference (ISWC2005)*, Galway, Ireland, 2005.
- [34] P. R. Polsani, "Use and Abuse of reusable Learning objects," *Journal of Digital Information*, 3(4): Article No. 164, 2004.
- [35] M. Poesio and A. Almuhaieb, "Identifying Concept Attributes Using A Classifier," *Proceedings of the ACL Workshop on Deep Lexical Acquisition*, pp. 18-27, Ann Arbor: Association for Computational Linguistics, 2005.
- [36] Protégé Ontology Editor. Retrieved June 10, 2007 from <http://protege.stanford.edu/>
- [37] S. Sánchez and M-A Sicilia, "On the Semantics of Aggregation and Generalization in Learning Object Contracts," *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT 2004)*, pp. 425-429. Joensuu, Finland, 2004.
- [38] I. Satoh and R. Okada, "Learning Object Combination Support System based on ARCS Model of Motivation," *Proceedings of Computers and Advanced Technology in Education (CATE 2005)*, Oranjestad, Aruba, 2005.
- [39] SCORM. Retrieved September 5, 2007, from Sharable Content Object Reference Model: <http://www.adlnet.gov/scorm/index.cfm>
- [40] W. W. Song, "A Metadata Framework for Description of Learning Objects," *Proceedings of the International Conference on Web-Based Learning (ICWL 2002)*, pp. 31-43, LNCS 2436, Springer-Verlag, Berlin, 2002.
- [41] L. Stojanovic, S. Staab and R. Studer, "eLearning based on the Semantic Web," *Proceedings of WebNet2001 - World Conference on the WWW and Internet*, Orlando, 2001.
- [42] UIMA. Retrieved June 10, 2007 from <http://uima-framework.sourceforge.net/>
- [43] C. Ullrich, "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, Japan, 2004.
- [44] L. Van Elst and A. Abecker, "Domain Ontology Agents for Distributed Organizational Memories," *Knowledge Management and Organizational Memories*, pp. 147-158, Kluwer Academic Publishers, 2002.
- [45] A. Zouaq, R. Nkambou and C. Frasson, "Building Domain Ontologies From Text For Educational Purposes," *Proceedings of the Second European Conference on Technology Enhanced Learning - "Creating new learning experiences on a global scale,"* Greece, 2007.
- [46] A. Zouaq and R. Nkambou, "Evaluating the Generation of Domain Ontologies in the Knowledge Puzzle Project," *IEEE Transactions on Knowledge and Data Engineering*, 08 Jan. 2009, IEEE computer Society Digital Library, <http://doi.ieeecomputersociety.org/10.1109/TKDE.2009.25>.
- [47] A. Zouaq and R. Nkambou, "Building Domain Ontologies from Text for Educational Purposes," *IEEE Transactions on Learning Technologies*, 1(1): 49-62, IEEE Computer Society, 2008.



Roger Nkambou is currently a 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 also serves as member of the program committee of the most important international conferences in Artificial Intelligence and Education.



Amal Zouaq received her Ph.D. (2008) in Computer Science from the University of Montreal. She is now a postdoctoral researcher at the Ecole Polytechnique de Montréal. She is also a lecturer in the Computer Science Department at the University of Quebec at Montreal. Her research interests include knowledge representation and extraction, Semantic Web, data mining, ontology engineering and computational linguistics. 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.