# Ontology extraction from existing educational content to improve personalized e-Learning experiences

Nicola Capuano, Luca Dell'Angelo, Francesco Orciuoli
Centro di Ricerca in Matematica Pura e Applicata
via Ponte don Melillo, 84084, Fisciano (SA), Italy
email: capuano@crmpa.unisa.it, dellangelo@crmpa.unisa.it, fraorc@crmpa.unisa.it

Sergio Miranda, Francesco Zurolo
MoMA s.r.l. - Modelli Matematici a Applicazioni
via Aldo Moro, 84081, Baronissi (SA), Italy
email: miranda@momanet.it, zurolo@momanet.it

## Abstract

*Nowadays, the use of domain ontologies in e-Learning applications is rapidly increasing due to the important role they play in knowledge representation, sharing of didactical material and content personalization. However, the ontology building processes is still extremely difficult to achieve. In this paper we present a semi-automatic process based on knowledge extraction from existing SCORM educational content aimed to speed up and facilitate the realization of domain ontologies and the breakdown of SCORM packages in fine-grained, rearrangeable learning objects appropriate for building personalized e-Learning experience.*

## 1. Introduction

In the modern society, learning is not yet confined to schools and universities but has become a lifelong need in a wide range of situations, either personal and professional. The growing request of continous training in past years had as effect a great investment in research and realization of virtual learning environments and digital content development. Also, the large amount of content developed and the appliance of different approaches for learning results in the necessity of content sharing and reusability. A large number of e-Learning systems use standardized structured content based on industry standards in order to satisfy such requests. SCORM is actually the widely adopted and supported standard in LMSs.

However, satisfying the wide range of learners' needs in e-Learning environments, requires providing learners with personalized educational experience based on their individual needs. Adaptive educational systems rely on the modeling of student's knowledge in the domain of learning, on a representation of the domain structure in terms of its elementary concepts and on evaluation of how well a student knows these concepts. Ontologies are well suited for the purpose of domain knowledge representation. An ontology is a formal, machine-understandable specification of a conceptualization of a domain of interest [2]. Ontologies are used to support semantic search, making possible to query multiple repositories and discover associations, among learning objects, that are not directly understandable.

Unfortunately, the importance of ontologies comes with the challenge of their building process. As a matter of fact, such process is time consuming and error prone. Also, from the adaptivity point of view, SCORM has some limitations. Its navigational model does not consider the necessity of personalization: it is fixed for a particular (or a set of) e-Learning experience(s).

In this paper we present a semi-automatic process based on knowledge extraction from existing SCORM educational content aimed at speeding up and facilitating the realization of domain ontologies expecially for users without knowledge engineering skills. The process, also, performs a breakdown of SCORM packages in fine-grained, rearrangeable learning objects appropriate for building personalized e-Learning experience.

The paper is organized as follows: Section 2 presents some related works; in Section 3 we describe our approach to build personalised e-Learning experiences through the use of ontologies; in Section 4 we describe the proposed approach to ontology extraction. Finally, Section 5 concludes the work.

978-0-7695-3800-6/09 $26.00 © 2009 IEEE
DOI 10.1109/ICSC.2009.69

## 2. Background and related works

### 2.1. SCORM

SCORM (Sharable Content Object Reference Model) is a set of standard and specifications developed by Advanced Distributed Learning (ADL) initiative. It is a reference model for the learning objects packaging and aggregation that aims at enabling learning object (re)use from any compatible LMS. Core SCORM comprises three documents: Content Aggregation Model (CAM), Run-Time Environment (RTE) e Sequencing and Navigation (SN).

The basic elements in SCORM are i) the Asset, which is content in its most basic form, electronic representation of text, images, sound and any piece of data that can be delivered on a web client, and ii) the SCO (Sharable Content Objects), a collection of one or more Assets that represents the lowest level of granularity of learning resource that can be tracked by a LMS using the SCORM Runtime Environment. These primary elements represent the learning objects that compose a hierarchycal course structure, a pedagogically neutral means to aggregate learning resources for the purpose of delivering a desired e-Learning experience

The sequencing and navigation rules in SCORM are based on IMS Simple Sequencing (IMS-SS) specification and are intended to provide the means to conditionally branch from one learning resource to other learning resources depending on whether the learner has completed certain material or reached an acceptable score.

Different authors ([4] [5] [9]) discuss limitations on the SCORM meta-data model.

### 2.2. e-Learning and personalization

Adaptive e-Learning systems, even if still quite few in number, aim to revolution online education by providing personalized e-Learning experience for each learner.

KGTutor, a knowledge grid based intelligent tutoring system [17], proposes a model for the construction of intelligent tutoring experiences in a more pleasant and effective way. It uses students characteristics, such as previous knowledg, to choose, organize, and deliver the learning materials to individual students. During the learning progress, the system can also provide objective evaluations and customized suggestions for each student according to their learning performance.

Another representative system that uses Semantic Web techniques in an e-Learning environment is the Courseware WatchDog [10]. WatchDog is completely ontology-based and uses clustering techniques to create personalised views of the learning objects. Moreover, it has some techniques for the management of the evolution of ontologies related to the educational content.

Knowledge modeling in these works is related to the creation and maintenance of ontologies.

### 2.3. Ontology learning

In literature Ontology Learning indicates the process of extracting ontological representations from large corpus of unstructured text-based documents. It is concerned with knowledge acquisition from text and builds on the large body of work in this direction within Natural Language Processing (NLP), Artificial Intelligence (AI), and Machine Learning [14],[16].

Systems based on these tecniques extracts terminology from a corpus of domain text, such as specialized Web sites and warehouses or documents exchanged among members of a virtual community. Then they filter the terminology using natural language processing and statistical techniques that perform comparative analysis across different domains, or contrastive corpora. Concepts are related according to taxonomic and other semantic relations. Relations are extracted using knowledge bases like WordNet[1].

Examples of ontology extraction system based on these tecniques are described in [16] and [8].

## 3. A personalization model for e-Learning

In the following subsections, we describe our approach to educational domains modeling using ontologies.

An e-Learning domain ontology can be modeled with a graph in which nodes are relevant concepts within the educational domain of interest and edges are binary relations between two concepts. Our approach mainly foresees three kinds of relations: *HasPart* (HP) that is an inclusion relation, *IsRequiredBy* (IRB) that is an order relation and *SuggestedOrder* (SO) that is a weak order relation.

If we have to model the educational domain *D* we have to conceptualize the knowledge underlying *D* and find a set of terms representing relevant concepts in *D*. The result of the previous step is the list of terms *T =(C, C1, C2, C3)* where T is one of the plausible conceptualizations of D. In order to explain the semantics of *HasPart* relation we can refer to the ontology illustrated in Fig. 1 where the three *HasPart* relations *HasPart*(C,C1), *HasPart*(C,C2), *HasPart*(C,C3) means, in terms of e-Learning, that in order to learn concept C learners have to learn concepts C1, C2 and C3 without considering a specic order.

In Fig. 1, we note the existence of elements that are not concepts nor relations. The new elements to introduce are the learning objects LO1 , LO2 and LO3 . The connection between a concept and a learning object, for instance C1 and LO1, is a *HasResource* (HR) relation. *HasRe-*

[1] http://wordnet.princeton.edu/

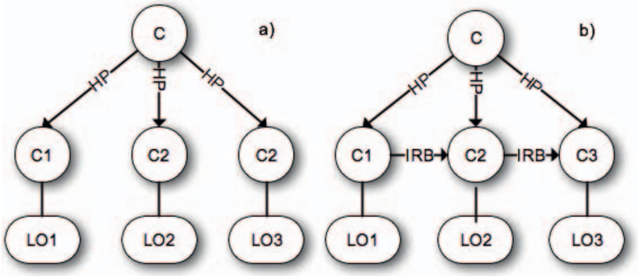


**Figure 1.**

*source(C1;LO1)* means that the educational content packaged in learning object LO1 explains concept C1 . In general, the relation HasResource can be represented by the function *HasResource*(C1; C2;...; Cn; LO1) meaning that LO1 explains all concepts C1 , C2 ,..., Cn; also, we can have at the same time the relations *HasResource*(C1; LO1), *HasResource*(C1; LO2), *HasResource*(C1; LO3), meaning that is possible to have more than one learning object (LO1, LO2, LO3) explaining the same concept (C1).

If we assume that our learning objective is C1 then the correspondent assembled e-Learning experience is composed only of LO1, otherwise if the learning objective is C then the assembled e-Learning experience will be composed of one of the plausible permutation of [LO1;LO2;LO3].

The ontology shown in Fig. 1 presents two *IsRequiredBy* relations, *IsRequiredBy*(C1;C2) and *IsRequiredBy*(C2; C3). The two relations mean that C1 has to be necessarily learned before C2 and C2 has to be necessarily learned before C3. In this case if C is the learning objective, learners have to learn the ordered sequence of concepts [C1; C2; C3] and correspondingly they can join the e-Learning experience assembled by the ordered sequence of Learning Objects [LO1; LO2; LO3]. Alternative permutations like [C2; C1; C3] will be invalid.

Lastly, suppose you have a *SuggestedOrder* relation between concept C1 and concept C2 that is *SuggestedOrder*(C1; C2), this relation states that the modeler thinks that is preferable to explain concept C1 before concept C2 , but this is not mandatory.

We have developed a complete e-Learning system (IWT - Intelligent Web Teacher, [1][6][12]) that on the basis of the extracted ontologies (representing the disciplinary domain of interest), can be used to define the sequence of concepts needed (by a learner) to acquire a satisfactory knowledge of learning objectives identied (by teacher) as target concepts of the given ontologies. A complete description of the overall e-Learning system is beyond the scope of this article.

# 4. Proposed approach to ontology extraction

## 4.1. Motivation

Despite the great benefit of personalization, in a pragmatic scenario, users still tend to build e-Learning experiences in more static ways. Tipically, users are inclined to assemble static e-Learning experiences collecting existing documents in sequences (or using pre-packaged courses) rather then work to build the knowledge infrastructure required by adaptive e-Learning systems.

Mainly, the choice can be attribute to the difficulties about the ontology construction process. In adaptive learning systems, ontologies are exploited not only to organize learning objects and to state their inter-relationships but also to build personalised e-Learning experiences and to maintain up to date students cognitive states. The ontology construction task requires concept identification and definition; also requires the identification and definition of relations among concepts. Despite the definition of language standards[2] and the availability of editing and management tools[3][13], populating domain ontologies with a sufficiently large number of concepts in a complex application domain is a tedious, time-consuming and error-prone process.

Moreover, users could have no competencies about domain conceptualization, modeling and they could miss the basic skills for the ontology building process.

The e-Learning environment built on the model described in section 3 provides advanced personalization functionalities, but still suffer the hostility to ontology building from most part of the users.

For simplicity, in our scenario we assume that educational documents are SCORM packages or self produced content like slides or word processing document. The proposed process is conceived to support users with the construction of all the elements required to build a personalized e-Learning experience starting from a SCORM package. The process is designed to provide support expecially to users with low (or totally missing) knowledge engineering competencies.

The elements involved are i) an e-Learning domain ontology, such as described in section 3 , ii) a set of learning objects, obtained from a breakdown of the initial set of documents, splitting packages and/or, eventually, single documents, iii) a set of metadata linked to the aforesaid learning objects, in order to make them available for the personalization algorithms.

The process is semi-automatic and some steps require user interaction. Moreover, the resulting ontology could contain some anomalies because of the automatic proce-

[2] http://www.w3.org/TR/2002/WD-owl-ref-20021112/

[3] http://protege.stanford.edu/

dures of extraction. Anomalies are shown to the user which can remove it by manual editing.

Our approach does not rely on classic ontology learning tecniques based on NLP and machine learning, or, al least, not primarly. There are two motivation for this. First, we work with a small set of documents; NLP tecniques rely on a large corpus of unstructured text, so, a lot of algorithms do not apply. Second, we suppose that the content has a structure, an explicit or implicit one, such as formatting information, chapter splitting, modules and lessons: we aim to take advantage from all this information.

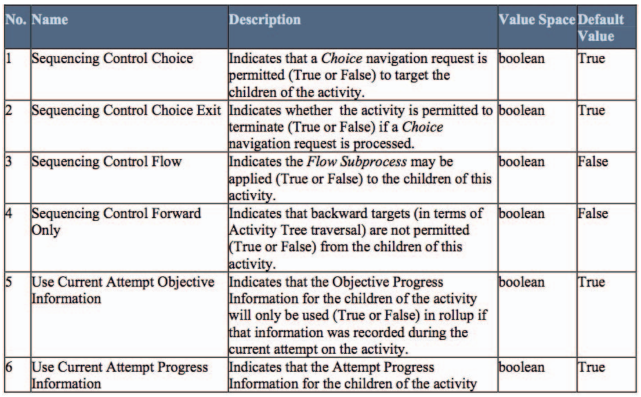
## 4.2. Process

The idea behind the proposed process of ontology extraction is to take advantage of both implicit and explicit knowledge within standard packaged structured content.

Aggregation and navigational structures are planned with a precise purpose in the mind of the SCORM content developer: to delivery a specific e-Learning experience in a particular way. So at a first level, the proposed process relies on the ability of the SCORM content developer to organize content in a well structured package. In a classic scenario, a SCORM course is logically divided in modules and each module is divided in lessons; modules and lessons are mapped in SCOs and Assets. We can assume that a lesson is designed to explain a fine grained concept whereas a module - that is, a set of lessons - aims at explaining a coarse grained concept.

At the same time, the SCORM content developer designs navigational rules to enable LMS to delivery a meaningful sequence of modules and lessons, that is, a sequence of resources, SCOs and Assets. We will show further how our process extracts and maps this information (about structure and navigation) on an ontological structure. This is the primary difference between our approach and the classic ontology extraction tecniques: we take advantage of all information about structure and navigation

We consider a SCORM package as input. The whole process can be summarized as as follows: i) draft ontology creation (detailed in section 4.2.1) - where a starting version is created from SCORM structure information; ii) resources analysis (detailed in section 4.2.2) - where a chunk of ontology (sub-ontology) is extracted for each resource of the SCORM; each chunk will be merged with the draft ontolgy; iii) resource splitting and metadatation - where the package is splitted according to the extracted conceptualization; iv) anomalies resolution - where the user edits the resulting ontology to remove possible anomalies due to the automatic extraction algorithms.



| No. | Name | Description | Value Space | Default Value |
|---|---|---|---|---|
| 1 | Sequencing Control Choice | Indicates that a *Choice* navigation request is permitted (True or False) to target the children of the activity. | boolean | True |
| 2 | Sequencing Control Choice Exit | Indicates whether the activity is permitted to terminate (True or False) if a *Choice* navigation request is processed. | boolean | True |
| 3 | Sequencing Control Flow | Indicates the *Flow Subprocess* may be applied (True or False) to the children of this activity. | boolean | False |
| 4 | Sequencing Control Forward Only | Indicates that backward targets (in terms of Activity Tree traversal) are not permitted (True or False) from the children of this activity. | boolean | False |
| 5 | Use Current Attempt Objective Information | Indicates that the Objective Progress Information for the children of the activity will only be used (True or False) in rollup if that information was recorded during the current attempt on the activity. | boolean | True |
| 6 | Use Current Attempt Progress Information | Indicates that the Attempt Progress Information for the children of the activity | boolean | True |

**Figure 2. SCORM Control modes**

### 4.2.1 Draft ontology creation

In this phase, a simple, draft version of the ontology is created working on the SCORM CAM information.

Content in SCORM packages is organized hierarchically (as shown in Fig.4 a)), with a different hierarchy for each organizaton element. An organization identifies the sequencing rules to use depending on the target audience the learner belongs to. If the package contains multiple organizations the user must select the one on which the process will work for the subsequent steps because for different organizations - and different sequencing rules - we will have different resulting ontologies.

From now, we can consider a single organization element as root of a hierarchy of items, one for each SCO/Asset, and we can map such hierarchy on an ontology.

The mapping (in Fig. 4 a),b)) is done with the following rules: i) we create a root concept for the draft ontology with the title of the root organization; ii) for each item in the hierarchy, we create a concept (described by the title of item element) in the draft ontology; iii) for each couple of items in a parent-child relation in the SCORM hierarchy we create a *HasPart* relation among the corresponding concepts in the draft ontology.

Then, we assign semantics to the navigation rules (refer to Fig. 2 for a subset of SCORM rules) and we map each rule with an ordering relation or a combination of ordering relations. For example (Fig. 4), we add a *SuggestedOrder* relation between two concepts if the item corresponding to the source concept has a *Sequencing Control Choice = true*; we add a *IsRequiredBy* relation if the value is *false*. The decision about the semantic of navigation rules can be different, but is just a matter of choice.

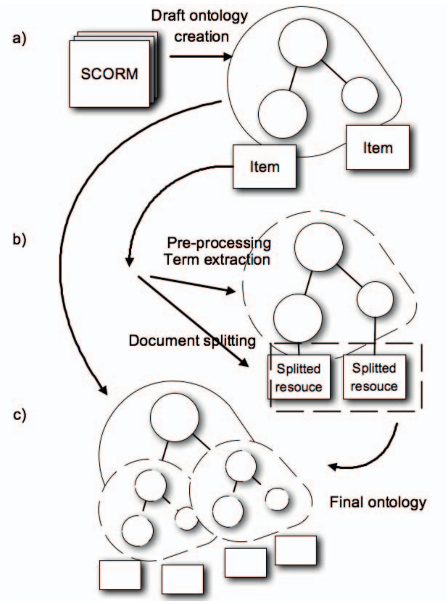


**Figure 3. A process overview**

### 4.2.2 Resources processing

At a deeper level, the process investigates the basic elements (Items/Assets, Fig. 3 b)) that compose the package. Our assumptions about analyzable documents can be summarized as follows: i) each document is assumed to have a partially defined structure. This allows us to make assumptions about how to initially model the ontology; ii) it is possible to consider also unstructured documents, but they must be pre-processed to make them semistructured at least; iii) our experimentation has been made considering PPT and DOC formats, but the approach is extendible to other document formats.

We can distinguish the following phases ([3] for a detailed description).

**Pre-processing** In this phase the documents are prepared for the extraction. We can distinguish several sub-phases, described in the following.

*Format conversion*: The documents are converted in an XML version (with additional annotations). *Stemming*: terms are reduced to their stem or root form (e.g. writing *in* write). *Part-of-speech tagging (POST)*: terms in the document are annotated as corresponding to a particular part- of-speech (i.e. names, verbs, adjectives, adverbs, etc.). *Stopword list*: terms that do not carry useful information (such as articles, conjunctions and verbs) are deleted. *Synonymous identication*: we use the WordNet lexical database to acquire the synonyms of each term: the acquired terms are associated to the first term and are taken into account during the text processing. *Terminology Extraction*: All the aforementioned sub-phases are performed to extract the relevant terminology related to a particular domain. These words are useful to conceptualize a knowledge domain. Also, we store the terms' position in the document for the splitting algorithm. *User intervention*: Even if the approach can be performed automatically by the implemented system, the human intervention can be useful to improve the pre-processing step. This can be particularly important to manage term deletion through the stopword lists.

**Concepts and relationships creation** We have implemented several statistical and data mining algorithms in order to identify the concepts and their relationship in the resulting ontology. We consider for example an algorithm that retrieves term frequencies from text (described in detail in [15]). The output of this algorithm can be used for the creation of concepts in the ontology.

To derive the concept hierarchy, we adopted a hierarchical clustering algorithm (see [15] for the complete explanation) that accesses background knowledge from existing ontological entities to label the extracted hierarchy. Moreover, we also implemented an algorithm that is based on frequent couplings of concepts by identifying linguistically related pairs of words, in order to acquire conceptual relations (see [14] for the complete description of the algorithm).

**Document splitting** We have implemented some algorithms in order to split documents according to the position of relevant terms (or cluster of terms). In the simplest case, if a term is relevant for a set of slides, these will be grouped in a new document corresponding to the concept of the relevant term.

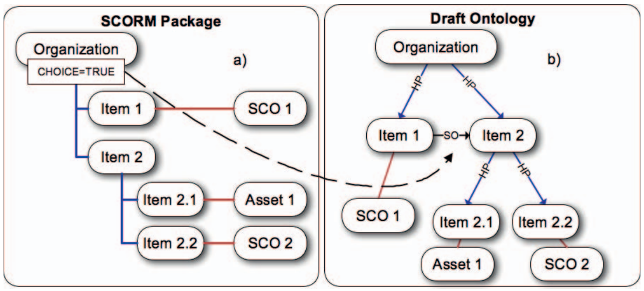


**Figure 4. SCORM structure mapping**

## 5. Conclusion

Most part of e-Learning users tend to not take advantage of personalization in e-Learning systems due to the required preliminary work of modeling domain by ontologies, because of difficulty of the process and expecially because of missing knowledge engineering competencies.

We developed an integrated approach (an e-Learning model, a process and a software that implements the process that aims at supporting users in the task of ontology building.

The system is in a prototype release. It is is developed mainly in Java technology and is based on open source libraries.

The document management subsystem is based on the Jakarta POI[4] libraries, which provide a set of functionality to access office documents. The SCORM analyzer is based on the ADL Runtime Environment libraries. The visualization and layout subsystem is based on jGraph[5] and Graphviz[6]. Also, the system is based on a set of libraries for text pre-processing[7],[8].

The developed software extract a series of elements useful for building a personalized e-Learning experience, simply starting from a SCORM package. The system requires small user interactions and does not require knowledge engineering competencies.

As result, the tool facilitates the access to the personalization functionalities for all typology of users. In particular, the tool outputs a draft domain ontology that can be refined with other advanced editing tools provided by the IWT system[1].

## References

[1] G Albano, M Gaeta, P Ritrovato *IWT: an innovative solution for AGS e-learning model* - International Journal of Knowledge and Learning, 2007 - Inderscience

[2] T. Gruber. Towards principles for the design of ontologies used for knowledge sharing. *Int. J. of Human and Computer Studies*, 43:907928, 1994.

[3] M. Gaeta, F. Orciuoli, P. Ritrovato - *Advanced ontology management system for personalised e-Learning* Knowledge-Based Systems 22 (2009) 292301

[4] N. A. Abdullah, C. Bailey, and H. Davis, ”Synthetic hypertext and hyperfiction: Augmenting SCORM manifests with adaptive links,” in Proceedings of the 15th ACM Conference on Hypertext and Hypermedia, Santa Cruz, EUA, 2004, pp. 183-184.

[5] D. Simoes, R. Luis, and N. Horta, ”Enhancing the SCORM Metadata Model,” in Proceedings of the 13th international World WiConference. New York, EUA, 2004, pp. 238-239.

[6] N. Capuano, M. Gaeta, S. Miranda, F. Orciuoli, P. Ritrovato *LIA: An Intelligent Advisor for e-Learning* - Emerging Technologies and Information Systems for the Knowledge Society, 187-196, September 20, 2008

[7] A. Maedche and S. Staab, *Ontology Learning for the Semantic Web*, IEEE Intelligent Systems,vol. 16,no. 2,Mar./Apr. 2001,pp. 7279.

[8] Roberto Navigli, Paola Velardi, and Aldo Gangemi. *Ontology learning and its application to automated terminology translation.* IEEE Intelli- gent Systems, 18(1):2231, 2003.

[9] Silva, L. et al (2006): Using Conceptual Lattices to Represent Fine Granular Learning Objects through SCORM Meta-Objects, In: Williams, S. (Ed): The Electronic Journal of e-Learning, Volume 4 Issue 2, pp. 141 148, available online at http://www.ejel.org

[10] J. Tane, C. Schmitz, G. Stumme, S. Staab, R. Studer, *The courseware watchdog: an ontology-based tool for nding and organizing learning material*, Beitrge der Fachtagung an der Universitt, Kassel University Press, 2003, pp. 93104.

[11] Tim Berners-Lee. Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor. Harper San Francisco, 1999.

[12] Giovannina Albano, Matteo Gaeta, Saverio Salerno - *E-learning: a model and process proposal* - International Journal of Knowledge and Learning Volume 2, Number 1-2 / 2006

[13] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke OntoEdit: Collaborative Ontology Development for the Semantic Web Proceedings of the rst International Semantic Web Conference 2002 (ISWC 2002), June 9-12 2002, Italy.

[14] Er Maedche and Steffen Staab. Mining non-taxonomic conceptual relations from text. In In: R.Dieng and O Corby. EKAW00 European Knowledge Acquisition Workshop. October 2-6, 2000, Juan-les-Pins. LNAI, Springer.

[15] Christopher D. Manning and Hinrich Schtze. Foundations of Statistical Natural Language Processing. The MIT Press, June 1999.

[16] Paola Velardi, Roberto Navigli, Alessandro Cucchiarelli, and Francesca Neri. *Evaluation of OntoLearn, a methodology for automatic population of domain ontologies*. Ontology Learning from Text: Methods, Applications and Evaluation. IOS Press, 2006.

[17] H. Zhuge, Y. Li, *KGTutor: a knowledge grid based intelligent tutoring system*, APWeb (2004) 473478.

[4] http://poi.apache.org/

[5] http://www.jgraph.com/

[6] http://www.graphviz.org/

[7] General Architecture for Text Engineering - http://gate.ac.uk/

[8] http://snowball.tartarus.org/