



Project no. 027391

Project acronym: LT4eL

Project title: Language Technology for eLearning

Instrument Specific Targeted Research Project

Thematic Priority Information Society Technology

D3.1 Ontology and integration report (1st cycle)

Due date of deliverable: 30-11-2006

Actual submission date: 21-12-2006

Start date of project: 1-12-2005

Duration: 30 Months

Organisation name of lead contractor for this deliverable: Institute for Parallel Processing, Bulgarian Academy of Sciences (IPP-BAS)

Revision [1]

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

D3.1 - Ontology Management in LT4eL - First Year Report

Contents

- 1 Introduction
- 2 LT4eL Ontology
- 3 Methodology for the Construction of a Domain Ontology
 - 3.1 Short Overview of Ontology Construction Methodologies
 - 3.1.1 Uschold and King Methodology
 - 3.1.2 Grüninger and Fox Methodology
 - 3.1.3 Methontology Methodology
 - 3.1.4 SENSUS Methodology
 - 3.1.5 Conclusions
 - 3.2 LT4eL Methodology
 - 3.2.1 Processing of the Keywords
 - 3.2.2 Formalisation of the meanings
 - 3.2.3 Link to an upper ontology
 - 3.2.4 Addition of new Concepts
 - 3.2.5 Addition of Relations
 - 3.2.6 Documentation
 - 3.2.7 Lexicons and Pilot Experiment
 - 3.3 Summary of the Ontology Creation Methodology
- 4 References
- 5 Appendix: Ontology Use Cases in LT4eL
 - 5.1 Integration into the Learning Management System
- 6 Appendix: Semantic Annotation

Introduction

In this report we present the work done on ontology within LT4eL during the first year of the project. The aim of the workpackage is to enhance Learning Management Systems (LMS) with semantic knowledge in order to improve the retrieval of the learning objects. Ontologies, which are a key element in the architecture of the Semantic Web, will be adopted to structure, query and navigate through the learning objects which are part of the LMS.

We will adopt Tim Gruber's definition of ontology: "An ontology is a specification of a conceptualization." In our work, the ontology is closer to a taxonomy, that is set of concepts arranged in Is-a hierarchy (i.e. an oak is a kind of tree). The ontology will play two roles:

- *Classification of learning objects.* Each learning object will be connected to a set of concepts in the ontology. This classification will allow ontological search, i.e. search based on concepts and their interrelations within the ontology.
- *Multilingual search for learning objects.* In this case the ontology plays the role of Interlingua between the different languages. Thus the user might specify the query in one language and get learning objects in other language(s).

The main effort during the first year was to create the ontology for the project. The construction of a domain ontology is not a trivial task as it requires a good coverage of the

chosen domain, linkage to an upper ontology and consistent underlying principles. In the case of LT4eL Project the targeted domain is *Computer Science for non-computer scientists*.

The creation of our domain ontology is data-driven. Hence, it depends on the keywords which have been selected and annotated by the partners.

In this document, we will present:

- a short overview of the Ontology Construction Methodologies
- parameterizing the methodology to the aims of LT4eL Project
- summary of the Ontology Creation Methodology
- conclusions and future work
- in several appendices we discuss the use cases of ontology within LT4eL architecture, the semantic annotation, semantic annotation tools and the pilot on lexicons.

LT4eL Ontology

We consider an ontology as a set of concepts which are connected via relations. The set of relations includes sub-concept (is-a, kind-of), part-of, caused-by, used-for, etc. The role of the ontology in Information Systems (see [30]) include:

- Information modelling
 - making better conceptual models (accurate, natural, explicit, reusable)
 - improving the methodology of conceptual analysis
- Information integration
 - agreeing on the intended meaning
 - ensure consistency
- Information access
 - semantic matching

With respect to the goals of LT4eL project and the application of ontologies in eLearning in general, we see the role of ontology mainly as a mechanism for agreeing on the intended meaning and in this way ensuring the multilinguality and semantic matching as a mechanism for search in a repository of learning objects. Our goal is an ontology of about 1000 concepts. The quality of the ontology will be evaluated in two respects: the consistency of the definitions and the coverage of the domain.

The consistency of the definitions will be ensured by using the OntoClean methodology (developed by Nicola Guarino and his co-workers) and the upper ontology in the process of the development of the domain ontology.

The coverage will be evaluated on the basis of ontological annotation of the learning objects in each language. The missing concepts and relations will be reported to the ontology developers and they will incorporate them in the ontology. Shortcomings due to multilingual mapping between concepts and terminology/text will be reported both to the ontology and to the vocabulary developers.

Methodology for the Construction of a Domain Ontology

Short Overview of Ontology Construction Methodologies

Several surveys on ontology construction methodology already exist. Thus, we will report on their findings here. We will present our choices for the different stages of the construction of the ontology in the next section (2.2 LT4eL Methodology). The literature focuses on two aspects with respect to the methodologies for creation of ontologies - organizational and

technological. The organizational aspect considers the steps involved in the creation of an ontology, whilst the technical aspect focuses on the tools necessary to realize each step. In this report our main focus will be the organizational side of the methodology. The overview in this section is based on: [1], [2], [3] (overviews of methodologies) and [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] (methodologies or different aspects of methodologies). First we will outline some of the most popular methodologies for ontology development, and then we will conclude this section with general points about the steps involved in ontology creation.

Uschold and King Methodology

This methodology was published first in [4]. They identified the following stages in the process of ontology creation: (1) Purpose Identification; (2) Building the Ontology; (3) Evaluation; and (4) Documentation. The building of the ontology (step 2) is further divided into three steps: (2.1) Ontology capture; (2.2) Ontology coding; (2.3) Integrating Existing Ontologies.

The *purpose identification* is an important step used in order to clarify why the ontology is being built and which its intended uses are. Some purposes stated in the literature are that the ontology is to act as a shared vocabulary for a domain, as a meta-level specification of a logical theory, and as a way of structuring a knowledge base. Purpose Identification of an ontology is directly relation to the scope of the ontology (see also [11]) - which concepts and relations will be chosen in the domain to be formalized, and on which level of granularity this formalization to be carried out. One important requirement mentioned in the paper is the reuse of the ontology not only within the group developing it, but also in a broader context. In our view the considered purposes are not mutually incompatible and thus when the ontology is developed, the group developing it could try to achieve as many as possible of them.

As mentioned, the actual building of the ontology comprises three steps: (2.1) Ontology capture; (2.2) Ontology coding; (2.3) Integrating Existing Ontologies. *Ontology capture* means: (1) identification of the key concepts and relationships in the domain of interest; (2) production of precise unambiguous text definitions for such concepts and relationships; (3) identification of terms to refer to such concepts and relationships; and (4) agreeing on all of the results from previous steps. One important question which arises with respect to ontology capturing is how the key concepts and relations will be identified. According to [4] there are three approaches: *top-down* - the most general concepts and relations are selected first and then they are specialised to the necessary level of domain dependence; *bottom-up* - first, the most specific concepts and relations in the domain are represented and then generalizations over them are defined; *middle-out* - the most fundamental concepts and relations in the domain are selected, and later their specializations and generalizations are added.

Ontology coding requires formalization of selected concepts and relations in a formal language. The process also requires encoding of axioms which the concepts and relations have to satisfy. In our opinion, at this stage the following question needs to be addressed: Is there a trade-off between the expressivity of the formal language for the ontology and the inference on this formal language. Sophisticated ontologies require encoding in very expressive languages like KIF, FOL, Modal Logic, however, these languages are not fully supported by ontological tools. Implemented languages like Description Logics (OWL-DL, for instance) allow only a part of the necessary axioms to be represented. Tackling this question is important, as its solution could lead to a representation language-dependent ontology which is not desirable. One option here is to have two versions of the ontology - one version is represented in an expressive language and one next version, derived from the first version, in an implemented language. Such an approach was adopted by the developers of the foundational ontology DOLCE - see [15] and [16].

Integrating existing ontologies is the basis of reusing existing ontologies. This task requires special attention with respect to the effort necessary to do such an integration. It depends on the quality of the existing ontology, the documentation, and whether it corresponds to the purpose of the new ontology. On the other hand it is very appealing with respect to the potential benefits.

The evaluation and the documentation stages are not described in great detail. The first is based on a definition given in [17]: "to make a technical judgement of the ontologies their associated software environment and documentation with respect to a frame of reference ... The frame of reference may be requirements specifications competency questions and/or the real world." The documentation stage needs established guidelines for documenting ontologies.

Grüninger and Fox Methodology

This methodology was developed under the TOVE (Toronto Virtual Enterprise) project - see [6] and [5]. It comprises the following stages of ontology development: (1) motivating scenarios; (2) informal competency questions; (3) terminology specification; (4) formal competency questions; (5) axiom specification; and (6) completeness theorems.

The *motivating scenarios* are stories about the different usages of the ontology in a given application. They highlight problems encountered in an application and their possible solutions. The solutions provide an informal intended semantics for the concepts and the relations to be included in the ontology. The *informal competency questions* are based on the motivating scenarios and will serve as the requirements of the ontology. The ontology must be able to provide answers to all these informal questions. These questions act as an evaluation process on the ontological commitments made in the previous stage. The questions have to reflect the specificity of the concepts and relations to be encoded in the ontology. Thus, they can also be used when one incorporates an already existing ontology into the new one. The transferred knowledge will also have to answer these questions. The *terminology specification* comprises of two steps - the determination of terms used in the informal competency questions; and the encoding of these terms in a formal language. The *formal competency questions* are the formalised requirements of the ontology. They are based on the informal competency questions. The *axiom specification* encodes the axioms that specify the definition of terms and the constraints on their interpretations. They are given in first-order logic, guided by the formal competency questions, as the axioms must be necessary and sufficient to express the competency questions and their solutions. The *completeness theorems* define the conditions under which the solutions to the competency questions are complete.

Methontology Methodology

The Methontology methodology is described in [8], [7] and [9]. It presents the construction of ontology based on knowledge level by identifying the following activities: (1) specification; (2) knowledge acquisition; (3) conceptualisation; (4) integration; (5) implementation; (6) evaluation; (7) documentation.

The *specification* defines the purpose of the ontology, including the intended users, scenarios of use, the degree of formality required, etc., and the scope of the ontology including the set of terms to be represented, their characteristics and the required granularity. The *knowledge acquisition* acquires knowledge about the domain of the ontology. Many different knowledge sources are analysed in order to achieve the task. The two main sources are expert interviews and analyses of domain texts. The *conceptualisation* structures the domain terms as concepts, relations, properties and instances. The *integration* of ontologies is required when ontologies or definitions from other ontologies are incorporated into the new one. During the *implementation* the ontology is formally represented in an ontological language (KIF, for example). The *evaluation* stage comprises of checks for incompleteness, inconsistency and redundancy cases in the ontology (see also [18]). The *documentation* reflects the results from the previous activities in natural language.

The methodology accepts that the life cycle of an ontology is based on the refinement of a prototype. The ontology goes through the following states: specification, conceptualisation, formalisation, integration, and implementation. Knowledge acquisition, evaluation and documentation are carried during the entire life cycle.

SENSUS Methodology

This methodology is presented in [10]. The main approach advocated here is the development of a domain ontology by linking it to a general ontology for natural language processing - SENSUS. SENSUS has more than 50000 concepts organised in a concept-sub-concept hierarchy. The concepts are mainly incorporated from WordNet - see [25] and [26]. The size of SENSUS and the way in which it is constructed ensures a broad coverage of the world knowledge; however, it does not contain concepts at the domain level. The stages for development of new domain ontology are as follows: (1) A set of "seed" terms are selected as representative concepts of a domain. (2) The seed terms are linked to the concepts in SENSUS manually. (3) All the concepts from the seed terms to the root of the hierarchy are added to the domain ontology. (4) New terms are included: new terms relevant to the domain terms, or terms from SENSUS that are siblings of the seed terms. Sibling terms are defined as terms that are under a more general term of a seed term in the hierarchy. In this way the domain ontology grows by two processes - adding more general concepts from SENSUS and adding sub-concepts of the added general concepts. The second operation has to be applied with care in order not to add specific concepts on the basis of overgeneralisation in step 3.

More on these methodologies and some others can be found in the papers cited at the beginning of the section.

Conclusions

Here are some conclusions:

1. The life cycle of an ontology is similar to that of other software products [2]: design phase, prototyping, implementation, exploitation, support, documentation.
2. There is a difference between a domain and an application in the domain. The ontology has to reflect the domain which it represents and at the same time be application-independent, yet satisfying the needs of the application.
3. "Ontology development is necessarily an iterative process" [11]. Usually the iteration is from a less expressive to a more expressive version and from an informal to a formal representation.
4. An evaluation of the sources of information is essential to the development of an ontology. The evaluation has to be done with respect to the availability of the source, the effort to use the source and how reliable these sources are.

All these conclusions are taken into account during the definition of our methodology for the creation of the domain ontology.

LT4eL Methodology

In this section we define the methodology for the creation of a domain ontology in the domain of computer science for non-computer scientists within the LT4eL project. During the creation process of the ontology we have ensured that the ontology covers (most of) the keywords selected by the project partners in their learning objects as well as the domain in great detail. Granularity is required in order to ensure better text annotation. In addition to this, the ontology was aligned with an upper ontology in order to ensure consistency with respect to the general ontology development methodology. In order to ensure coverage and granularity of the ontology, we envisaged that its creation required the following steps:

Processing of the Keywords

One of the main tasks for which the ontology will be used is the semantic annotation of learning objects that the partners processed during the first phase of the project. In order to

ensure a relatively wide coverage with respect to the learning objects we began creating the ontology by processing the keywords annotated by the partners in their learning objects. Thus, we started the creation of the ontology not only from documents in one language, but we considered the extracted keywords in all 9 languages, with their english translation. This ensures a better initial coverage of the conceptual space and reduces the complexity of mapping multilingual lexical entries on the ontology.

The processing itself was done in the following way:

1. *Initial classification of keywords*

This step involved selecting keywords which are connected to the usage of computers by non-computer scientists. Defined in this way, the domain is extremely large or vague and contains some concepts that are not, strictly speaking, part of computer science. Such concepts are related to the representation of objects that are frequently processed by computers such as documents, papers, etc and domains of usage of computers like distance learning, desktop publishing, etc. A large proportion of the keywords are related also to the creation of web pages and related technologies. Each selected keyword was transformed to its normalized form (without inflective forms).

2. *Definition collection*

The Internet was searched for definitions of the selected keywords. The idea of this step was to define in 'human friendly language' the concepts connected to the keywords. We have collected a bunch of definitions for most of the keywords because different definitions highlight different features of the related concept. Also, the collected definitions can be considered as a corpus with definitions for the concepts, and later could be used as a testing basis for automatic acquisition of conceptual information.

3. *Definition selection and Sense differentiation*

On the basis of the collected definitions we formulated (or just selected) one definition for each concept represented by a keyword. These definitions are the human explanations of the meanings of the concepts that have to be encoded in the ontology. When necessary, two or more meanings were isolated for one keyword. For instance, the keywords *header* and *word* have more than one meaning. Other keywords show regular polysemy. For example, MPEG might be the organization as well as the standard. Our rule of thumb was to prefer the more general meaning(s) to the more specific ones. The reason for such a rule lies in the aims of the project, i.e. to cover the CS domain for non-CS users. Hence, we skipped the meanings indicating pure programming terms.

Formalisation of the meanings

The next step was to define formal definitions of the extracted concepts and relations in OWL-DL - see [19]. OWL-DL was chosen because one can find implemented reasoners for it. We decided to define the concepts in two separate steps. First, for each meaning, an appropriate class in the domain ontology was created. Later, we will also add the properties of the concepts and relate them to other classes in the ontology. The result of this step is an initial formal version of the ontology. In order to ensure appropriate taxonomic relations between the concepts in the ontology and to facilitate the mapping to an upper ontology, we mapped each concept to synsets in WordNet ([25], [26]) (more precisely to OntoWordNet [31], which is a version of WordNet 1.6 mapped to DOLCE ontology). The mapping was performed via two main relations *equality* and *hypernymy*. The first is between a class in the ontology and a synset in WordNet which (lexically) represents the same concept, while the second is a relation between a class in our ontology and a synset denoting a more general concept. Thus, we first create the taxonomy of our ontology. Later we will interconnect the concepts in it on the basis of other relations. The connection of OntoWordNet to DOLCE will allow an evaluation of the defined concepts with respect to meta-ontological properties as they are defined in the OntoClean approach - see [20], [21] and [22]. OntoClean methodology for conceptual analysis is based on the idea of classification of the concepts, defined in an ontology, with respect to meta-properties like *identity*, *essence*, *unity*, *dependence*, etc. Different concepts have different combinations of these meta-properties. On the basis of these combinations, the concepts are classified as *types*, *material roles*, *formal roles*, *categories*, etc. The taxonomic relation can be established only between concepts of some

kinds. Thus, assigning meta-properties to concepts in an ontology helps for improving of its quality.

In fact we did two interconnected mappings from the domain ontology to OntoWordNet and to WordNet 2.0. The first mapping is used as it is described in the previous paragraph. It facilitates the mapping to the Upper Ontology and defines the taxonomy of the domain ontology to a certain extent. The mapping to WordNet 2.0 provides us with several benefits: first, WordNet 2.0 is bigger than OntoWordNet and gives us a better mapping; second, in OntoWordNet only the hyperonymy relation is presented and all other relations from WordNet are missing. Therefore, the mapping to WordNet 2.0 will help us to encode some of the relations between the concepts in our ontology; third, WordNet 2.0 is aligned to SUMO and thus we will have an indirect mapping from the domain ontology to another upper level ontology. The two mappings were done by different people independently. Thus, they play validation check for each other.

Link to an upper ontology

Establishment of the connection between the upper and the domain ontology will help us check the consistency of the domain ontology with respect to the ontology construction methodology behind the upper ontology and to inherit the knowledge encoded in the upper ontology to the domain one.

There are several choices we can select in order to determine which upper ontology to be used as a basis of the development of the domain ontology. The initial list of ontologies included: DOLCE Ontology (<http://www.loa-cnr.it/DOLCE.html>), SUMO Ontology (<http://www.ontologyportal.org/>), OpenCyc Ontology (<http://www.cyc.com/cyc/opencyc/overview>), Omega Ontology (<http://omega.isi.edu/>), Basic Formal Ontology (<http://ontology.buffalo.edu/bfo/BFO.htm>), PROTON Ontology (<http://proton.semanticweb.org/>), SmartWeb Integrated Ontology (http://smartweb.dfki.de/ontology_en.html). We considered the following criteria for the selection of the upper ontology: (1) The ontology has to be constructed on a rigorous basis which reflects the OntoClean (or similar methodology) and suits our domain; (2) Ease of representation in some of the ontological languages (OWL-DL preferably); (3) There are domain ontologies constructed with respect to it (in order to facilitate the links with our domain ontology); and (4) Support is provided to us by the authors of the upper ontology. After some initial evaluation of the candidate ontologies and consultations with other evaluations of upper ontologies - see [24] and [23] - we selected DOLCE Ontology as our upper ontology. Note that this decision does not aim at eliminating mappings to other ontologies. It just ensures the consistency of our work. Hence, the mapping to SUMO is also gained via the mapping to WordNet 2.0. As a consequence, later on a comparison might be done between these mappings.

Here we present a summary of the ontological choices behind DOLCE - for more details and explanations see [15]:

Descriptive vs. Revisionary ontology

"A *descriptive ontology* aims at capturing the ontological stands that shape natural language and human cognition. It is based on the assumption that the surface structure of natural language and the so-called commonsense have ontological relevance. As a consequence, the categories refer to cognitive artifacts more or less depending on human perception, cultural imprints and social conventions. Under this approach, there are no major restrictions on the postulation of ontological categories because overall philosophical or scientific paradigms are neglected. This attitude stands in contrast to the *revisionary approach*. The revisionist considers linguistic and cognitive issues at the level of secondary sources (if considered at all), and does not hesitate to paraphrase linguistic expressions (or to re-interpret cognitive phenomena) when their ontological assumptions are not defensible on scientific grounds." ([15, page 7])

Multiplicative vs. Reductionist ontology

"In designing ontologies, one has to model a considerable amount of concepts. These concepts form a wide taxonomy and are often intertwined in several ways. Since the complexity of the resulting system is quite high, there are considerable advantages in limiting the actual primitives to a small subset of the concepts. If this is possible, then many notions can be reconstructed in terms of the chosen primitives. A reductionist ontologist takes this view as a major guideline; he aims at describing a great number of ontological concepts with the smallest number of primitives. On the other hand, a multiplicative ontologist points at reaching a very expressive system without bothering about the complexity of the ontology. Indeed, the aim is to provide a reliable account of reality despite of the large number of basic concepts needed." ([15, page 8])

Possibilism vs. Actualism

"Actualism claims that only what is real exists, while possibilism admits possibilia (situations or worlds) as well." ([15, page 8])

Endurants and Perdurants distinction

"Classically, endurants (also called continuants) are characterized as entities that are 'in time', they are 'wholly' present (all their proper parts are present) at any time of their existence. On the other hand, perdurants (also called occurrents) are entities that 'happen in time', they extend in time by accumulating different 'temporal parts', so that, at any time t at which they exist, only their temporal parts at t are present." ([15, page 11]) Based on this distinction one can define two different approaches to ontology modelling: perdurantism and endurantism - see [23, page 7]. Perdurantism assumes that entities extend in time and space. That means entities have both spatial and temporal parts (and, therefore, four dimensions). Endurantism treats entities as 3D objects (sometimes called endurants or continuants) that pass through time and are wholly present at each point in time.

DOLCE ontology is a descriptive, multiplicative ontology which adopts possibilism as an approach to existence and perdurantism as an approach to change. By selecting DOLCE as an upper ontology for our domain ontology, we also adopt the formal background on which this ontology is constructed - the OntoClean approach. These properties of DOLCE indicated that it was the most appropriate upper ontology for our purposes. Descriptivity reflects the primary usage of the our domain ontology - representing the end user view on the domain. The same reason is true for multiplicative nature of the ontology: the end user views over the domain are rare, non-redundant and uniform. Possibilism is important with respect to supporting interior designs with materials which do not actually exist at the moment, but could exist in principle. Also possibilism will be important when the trends are defined within the trend analyser - the trends in principle will not describe existing situations, but possible ones. The perdurantism ensures the support of changes of the represented entities in time. This will be important for support of the evolution of an interior design and/or its reuse. Thus, DOLCE is appropriate as an upper ontology for our domain. With respect to the other criteria for the selection of an ontology, it can be mentioned that DOLCE is coming in a form which is encoded in OWL-DL; it reflects the requirements of the OntoClean approach; there are extensions of DOLCE with more specific concepts and relations which can be used in the process of creation of the domain ontology. We have contacted the authors of DOLCE for support, which they kindly promised.

The linking from the domain ontology to DOLCE ontology is done via the mapping to OntoWordNet. Each concept will be connected to one or more concepts in DOLCE. Similarly, each domain relation will later be attached to a relation in DOLCE. Where there is no appropriate relations in DOLCE, such relations will have only local definitions in the domain ontology. The meaning of the links between the two ontologies will be is-a (concept and relation specialization). Thus, the concepts and the relations will inherit the definitions of the corresponding concepts (relations) in DOLCE. Also the OntoClean meta-properties will be inherited. The next step will be to check the consistency of the inherited information. If there are conflicts, then we will examine the involved concepts and relations and the domain ontology will be redefined locally. When the concept/relation definitions are inherited from DOLCE, it could be the case that there is a need to create new domain concepts/relations in

order to have better inheritance. This is the difference between our approach and the approach of SENSUS methodology where whole branches from SENSUS hierarchy are added to the domain ontology.

We consider this step as top-down approach to the creation of the ontology. Comparing to the definitions of top-down, bottom-up and middle-out approaches of [4] (see above), our approach is strictly top-down with respect to the inheritance from DOLCE to the domain ontology. With respect to the links from the domain ontology to DOLCE, we can assume them to be bottom-up, because at some places it could be necessary to introduce new concepts between the domain concepts and the concepts in DOLCE. We will call these new concepts middle-level concepts. Among the concepts in the domain ontology it is hard to predict the level of specificity of the extracted concepts; it will depend on the information represented in the standards in the domain.

Addition of new Concepts

Processing only some of the keywords extracted from the learning objects does not suffice for the creation of an ontology with a wide coverage over the domain, because keywords reflect only the topics of the learning objects. In order to ensure better coverage we have added concepts that are denoted by synsets in OntoWordNet and which are related to our domain. The result of all these steps up is the first version of the ontology to be incorporated within LT4eL system.

Addition of Relations

Relations between the concepts in the domain ontology will be done in two ways. First, relations that are defined in DOLCE will be inherited to the domain ontology. These relations will determine some general relationships between the concepts in the domain ontology. Second, domain specific relations will be introduced on the basis of analysis of the definitions that we have collected for the keywords. These relations will be created manually.

Documentation

In the process of construction of the ontology we keep track on the sources of each concept and relation. The track records point to the context from where the concept originated. Also, we will keep information about all changes in the definition of the concept or the relation. An important part of the documentation will be the natural language definitions created during the initial stage of concept definitions.

Lexicons and Pilot Experiment

In parallel to the ontology construction, we will also create lexicons for all languages of the project: English, German, Dutch, Portuguese, Maltese, Polish, Czech, Romanian, and Bulgarian. These lexicons will provide the vocabulary for the ontology concepts and relations in the corresponding language. In this way, we will facilitate the usage of the ontology for interaction with the human users of the ontology and the annotation of the learning objects. These lexicons could be considered as ordinary terminological lexicons in the domain, except that each term in them will have a formal definition represented in the ontology.

▪ Pilot Experiment with a Multilingual Lexicon

As it was said above, during processing the keywords we have also created a terminological lexicon in the domain. The main aim of the lexicon is to fix some set of concepts to be encoded within the ontology. Although the original set of keywords was compiled on the basis of nine languages. The English translation was the main source of terms and definitions. Thus, we expected some of the encoded meanings to be English specific. In order to investigate earlier the problems related to multilingual access to the ontology, we decided to create

terminological lexicons in German, Dutch, Romanian and Bulgarian related to the ontology. The experience from this pilot lexicon creation will be useful for the creation of the full lexicons for all nine languages during the second year of the project.

The structure of the lexicon is encoded in a DTD. It includes the following elements: the **<entry>** elements contain all the information about one meaning in the lexicon; the **head word group** (**<hwg>**) element of an entry contains all the lexical items (encoded in separate **head word** elements) that represent the meaning of encoded in the entry; the **definition** (**<def>**) element of an entry contains the definition of the meaning encoded in the entry; the **translation group element** (**<trg>**) elements contains for each language the corresponding lexical items for the meaning encoded in the entry. There is one **<trg>** element for each of the languages in the pilot. The lexical items for a given language are represented separately in **<tr>** elements inside of the corresponding **<trg>** element. Additionally, we have added an element for the definitions that are collected on the Internet for the English lexical items (element **<defbag>**), the mappings to WordNet 2.0 (element **<wn20>**) and OntoWordNet (element **<own>**) and the formal definition of the concept in OWL (element **<owldef>**).

After we have selected the appropriate meanings for about 200 entries all the partners involved in the pilot added the lexical items for their languages. The main problems identified during the creation of the multilingual terminological lexicon are related to the differences on the conceptual level in the different languages. Of course this is a well-known problem. The solutions to this problem that we envisage on the basis of the pilot are the following:

More detailed classes in the ontology. In cases where it is possible, we will create more specific concepts in the ontology. For example, the concept of "shortcut" as it is defined now is the general one, but the lexical items in English to some extent depend on the operating system, because often each operating system introduce its own terminology. When the notion is borrowed in other languages it could be borrowed in different granularity, thus, we could need more specific concepts in the ontology in order to ensure correct mapping between languages.

More complex mapping between the ontology and some lexicon. Our initial idea was that each meaning of a lexical item in any language is mapped to exactly one concepts in the ontology. If for some lexical item this one-to-one mapping is not appropriate or it will require a very complicated changes in the ontology we envisage a mapping based on OWL expressions. This mechanism will allow us to keep the ontology simpler and more understandable and to handle cases that do not allow appropriate mappings.

Using of non-lexicalized phrases. When a concept is missing lexicalization in some language we decided to use a free phrases for expressing of the concept in the given language. This solution require a special treatment of the "head words" in the lexicons, because such phrases will allow bigger freedom with respect to their occurrences in the text.

Variability is a problem even with respect to the lexicalized cases and our idea is to represent the most frequent (based on the learning objects we already processed) variants for each concept. We will not be able to solve this problem in general in the project, but we hope to demonstrate some approaches to it.

The ambiguity of the terms in the different languages impose additionally the problem of language dependant navigation over the ontology. If, for example, a user is using "key", but it is not clear whether this is the a part of the keyboard or this is a code, then the navigation system will not be able to select the right concept. In such cases we envisage to ask the user to be more specific about the concept of interest or to investigate different parts of the ontology.

In general, we consider the pilot of great help to the success of the project, because we know better what to expect as problems during the next phase of the project, also, now, more people know the ontology and this will help the next steps of development of full lexicons for all languages.

Summary of the Ontology Creation Methodology

We have selected a methodology which reflects the nature of the ontology for LT4eL. The general approach to the creation of the ontology is one of gradual formalization of the concepts in the ontology from natural language terms with informal definition to formal relational model of the domain. In this respect, we follow the hierarchy of precision of ontologies as presented by Nicola Guarino [30]:

- Lexicon
 - Vocabulary with NL definitions
- Simple Taxonomy
- Thesaurus
 - Taxonomy plus related-terms
- Relational Model
 - Unconstrained use of arbitrary relations
- Fully Axiomatized Theory

The keyword processing corresponds to the lexicon level. The mapping to OntoWordNet and WordNet ensures the levels of simple taxonomy and thesaurus. The mapping to DOLCE gives us a way to achieve the level of the relational model with some axioms inherited from DOLCE. Thus the resulting ontology will be a relational model and it will be conformant to the DOLCE methodological principles. In our view, an ontology on this level is most appropriate for the LT4eL project objectives.

References

- [1] Mariano Fernández-López. (ed.). 2002. Deliverable 1.4: A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies.
- [2] Mariano Fernández- López. 1999. Overview of Methodologies for Building Ontologies. Proceedings of IJCAI99's Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends. pp. 4.1-4.13.
- [3] Mike Uschold and Martin King. 1995. Towards a Methodology for Building Ontologies. Proceedings of IJCAI95's Workshop on Basic Ontological Issues in Knowledge Sharing.
- [4] Mike Uschold and Michael Gruninger. 1996. Ontologies: Principles, Methods and Applications. Knowledge Engineering Review. 11(2).
- [5] Michael Grüninger and Mark Fox. 1994. The Role of Competency Questions in Enterprise Engineering. IFIP WG 5.7 Workshop on Benchmarking. Theory and Practice. Trondheim, Norway.
- [6] Michael Grüninger and Mark Fox. 1995. Methodology for the Design and Evaluation of Ontologies. Proceedings of IJCAI95's Workshop on Basic Ontological Issues in Knowledge Sharing.
- [7] Asunción Gómez-Pérez. 1996. Towards a Framework to Verify Knowledge Sharing Technology. Expert Systems with Applications. 11(4), pp. 519-529.
- [8] Mariano Fernández, Asunción Gómez-Pérez and N. Juristo. 1997. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. Proceedings of AAAI97 Spring Symposium Series, Workshop on Ontological Engineering. pp. 33-40.
- [9] Mariano Fernández, Asunción Gómez-Pérez, Alexandro Pazos Sierra and Juan Pazos Sierra.

1999. Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment. *IEEE Expert (Intelligent Systems and Their Applications)*. 14(1), pp. 37-46.
- [10] William Swartout, Patil Ramesh, Kevin Knight, Thomas Russ. 1997. Toward Distributed Use of Large-Scale Ontologies. Symposium on Ontological Engineering of AAAI. Stanford (California).
- [11] Natalya F. Noy, Deborah L. McGuinness. 2001. *Ontology Development 101: A Guide to Creating Your First Ontology*. Knowledge Systems Laboratory.
- [12] Guus Schreiber, Bob Wielinga and Wouter Jasnweijer. 1995. "The KACTUS view on the 'O' word". In *Proceedings of the Nationala Dutch AI Conference. NAIC'95*.
- [13] Dean Jones, Trevor Bench-Capon, Pepijn Visser. 1998. Methodologies For Ontology Development. In *Proc. IT&KNOWS Conference, XV IFIP World Computer Congress, Budapest*.
- [14] Howard Beck and Helena Sofia Pinto. 2002. "Overview of Approach, Methodologies, Standards, and Tools for Ontologies." Agricultural Ontology Service, UN FAO. <http://www.fao.org/agris/aos/Documents/BackgroundAOS.html>
- [15] Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, Alessandro Oltramari and Luc Schneider. 2002. The WonderWeb Library of Foundational Ontologies. WonderWeb Deliverable D17, August 2002. <http://www.loa-cnr.it/Publications.html>.
- [16] Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino and Alessandro Oltramari. 2002. *Ontology Library (final)*. WonderWeb Deliverable D18, December 2003. <http://www.loa-cnr.it/Publications.html>.
- [17] Asunción Gómez-Pérez, Natalia Juristo and Juan Pazos. 1995. Evaluation and Assessment of the Knowledge Sharing Technology. *Towards Very Large Knowledge Bases*. N.J.I. Mars. Ed. IOS Press. pp. 289-296.
- [18] Asunción Gómez-Pérez. 1996. *Towards a Framework to Verify Knowledge Sharing Technology Expert System With Applications*. Vol 11. N. 4. Pages: 519-529.
- [19] OWL. Web Ontology Language (Overview). <http://www.w3.org/TR/owl-features/>
- [20] Aldo Gangemi, Nicola Guarino, Claudio Masolo and Alessandro Oltramari. 2001. Understanding top-level ontological distinctions. *Proc. of IJCAI 2001 workshop on Ontologies and Information Sharing*.
- [21] Christopher Welty and Nicola Guarino. 2001. Supporting Ontological Analysis of Taxonomic Relationships. *Data and Knowledge Engineering*.
- [22] Nicola Guarino and Christopher Welty. 2002. "Evaluating Ontological Decisions with OntoClean." *Communications of the ACM*, 45(2): 61-65.
- [23] Daniel Oberle, Anupriya Ankolekar, Pascal Hitzler, Philipp Cimiano, Michael Sintek, Malte Kiesel, B. Mougouie, S. Vembu, S. Baumann, Massimo Romanelli, Paul Buitelaar, R. Engel, D. Sonntag, N. Reithinger, Berenike Loos, R. Porzel, H.-P. Zorn, V. Micelli, C Schmidt, Moritz Weiten, F. Burkhardt, J. Zhou. 2006. "DOLCE ergo SUMO: On Foundational and Domain Models in SWIntO (SmartWeb Integrated Ontology)", Submission to *Journal of Web Semantics* (2006).
- [24] Salim Semy, Mary Pulvermacher, Leo Obrst. 2004. *Toward the Use of an Upper Ontology for U.S. Government and Military Domains: An Evaluation*. MITRE Technical Report 04B0000063, September, 2004.
- [25] George A. Miller. 1995. WORDNET: A Lexical Database for English. *Communications of ACM*, 11. p. 39-41.
- [26] Christiane Fellbaum. 1998. Editor. *WORDNET: an electronic lexical database*. MIT Press.

[27] Paul Buitelaar. 2006. Knowledge Markup and Ontology Learning for Semantic Metadata Extraction. An invited talk at the First Workshop on Natural Language Processing for Metadata Extraction - NLP4ME 2006. Varna. Bulgaria.

[28] Kiril Simov, Zdravko Peev, Milen Kouylekov, Alexander Simov, Marin Dimitrov, Atanas Kiryakov. 2001. CLaRK - an XML Based System for Corpora Development. UCREL Technical Paper number 13. Special issue. Proceedings of the Corpus Linguistics 2001 conference, edited by Paul Rayson, Andrew Wilson, Tony McEnery, Andrew Hardie and Shereen Khoja. ISBN 1 86220 107 2. Lancaster University (UK).

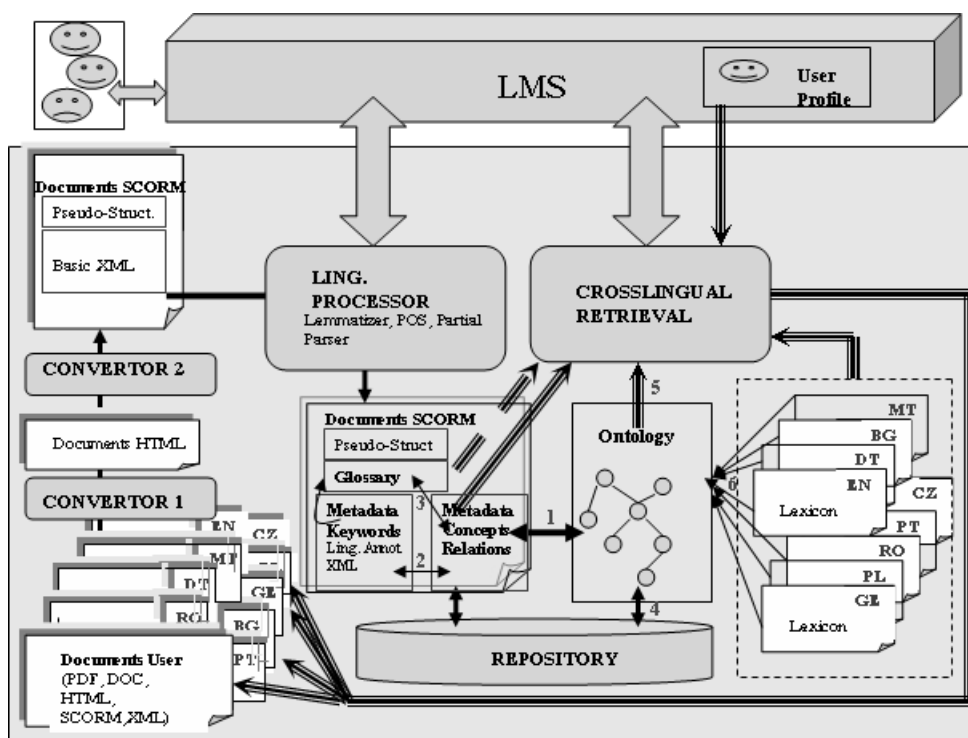
[29] Aldo Gangemi, Domenico M. Pisanelli, Geri Steve. 1999. An Overview of the ONIONS project: Applying Ontologies to the Integration of Medical Terminologies. In: Data and Knowledge Engineering, vol. 31.

[30] Nicola Guarino. 2000. Ontological Analysis and Ontology Design. An invited tutorial at the First Workshop on Ontologies and Lexical Knowledge Bases - OntoLex 2000. Sozopol, Bulgaria.

[31] Aldo Gangemi, Roberto Navigli, and Paola Velardi. 2003. The OntoWordNet Project: extension and axiomatisation of conceptual relations in WordNet. International Conference on Ontologies, Databases and Applications of SEMantics (ODBASE 2003), Catania, Italy.

Appendix: Ontology Use Cases in LT4eL

Here we describe the role of the Ontology Management System (OMS) within the LT4eL architecture and the use cases for the ontology component. In the following picture the LT4eL architecture is presented:



The Ontology Management System interaction with the rest of the LT4eL system is represented by numbered arrows. Link 1 represents the connection of OMS with the metadata section of a given learning object. The author (or the annotator) of the learning object can select any concepts and/or relations from the ontology that reflect the content of the learning object. Links 2 and 3 represent (possible) connection to the other elements of the metadata section of the learning object (keywords and glossary) and the content of the learning object (linguistic annotation). Link 4 depicts the relations between the ontology and the (potential) grouping of learning objects into courses. Link 5 represents the usage of the ontology for formulating of Information Retrieval queries to the LOs Repository. These queries are defined

as a set of concepts and relations from the ontology (the structure of the query is discussed below). The definition of the queries can be mediated by the lexicons aligned to the ontology. The alignment facilitates the multilingual usage of LT4eL system. The alignment is depicted by links 6. Generally speaking the alignment defines the meaning of the lexical items in the lexicons via concepts and relations in the ontology.

Based on these links between the ontology and the rest of the LT4eL system the following use cases are identified:

- **Ontology Design Use Case**

It requires services like: Editing Ontology, Storing Ontology, Lexicon Construction and Management. The main user involved in this use case is the LT4eL Ontology Engineer. The need of ontology and lexicon management arises in several cases: (1) A new ontology is under development; (2) An extension of existing ontology is necessary; (3) New lexical items are required for some concepts and relations. In such cases the tasks are: (1) A new domain is necessary to be formalized as ontology; (2) New concepts and relations are necessary, or it is necessary to modify existing ones; (3) New lexical items represent the concepts and relations defined in the ontology; (4) A lexicon for a new language has to be aligned to the ontology. In achieving these tasks the ontology engineer modifies the ontology or the lexicon of some language in order to incorporate some new ontological/lexical information. In this work the ontology engineer could need help from domain experts and computational linguists.

- **Ontology Annotation Use Case**

It requires services like: Visualization of the Learning Object, Identification of the Annotated Elements, Selecting Ontological Information (Classes and Relations) from Ontology. The main user involved in this use case is the author of the learning object or content annotator. The goal is to improve the retrieval of learning objects on the basis of ontology information added to the metadata of the learning objects. The need of ontology annotation arises when new learning objects are added to the repository. The result from the linguistic processing is available: annotated learning object, list of keywords, glosses. The result from the ontology annotation is that some concepts and relations from the ontology are added to the metadata of the learning object. As an intermediate step is the addition of the ontological information in the content part of the learning object. Such a step will be useful because the context in which the ontological information is added might help the disambiguation in cases of ambiguity. In achieving the task the author of the learning object or a specially trained content annotator has to add concepts and relations to the learning object. In this work they have to select the concepts and the relations from the ontology that are relevant to the content of the learning object. They could perform the following actions: navigation through the ontology; consultation of the lexicon on the bases of the keywords and the glosses (mapping from the keywords, phrases to the lexicon). The selected concepts and relations are stored in the metadata section of the learning object or temporarily in its content part. There are different levels of precision of the ontology annotation. For the moment, we envisage annotation with the more specific concepts and relations relevant to the content of the learning object.

- **Self Education Use Case**

It requires services like: Ontology Navigation, Ontology Query Formulation, Learning Object Selection. The main user involved in this use case is a student or a casual learner who wants to learn more in some area. The main goal of the learner is to identify the learning objects in the repository that are relevant to the area of interest. After the learning objects relevant to the learner's needs are identified they are extracted from the repository. This use-case is similar to the usage of keywords, glosses and other metadata attached to the learning objects. The difference is that the learner is formulating the query in terms of concepts and relations from the ontology. The query formulating in this way is evaluating with respect to the ontology annotation of the learning objects in the repository and the relevant learning objects are extracted. The formulation of the query can be done by inspection of the ontology directly

or via some of the lexicons. The form of the query depends on the ontology annotation of the learning objects. In some cases inference will be necessary in order to find the relevant learning objects. The learning objects can be in different languages depending on the learner.

- Course Compilation Use Case

This use case is very similar to the previous one because it will require the same services and actions. The difference is that in this case the main user is a person who knows the area and it will be much easy to formulate queries for searching of learning objects.

Having these use-cases in mind OMS of LT4eL will provide the following services:

- Editing of Ontologies - (Ontology Editor Component);
- Storage of Ontologies - (Ontology Repository Component);
- Inference Services (Consistency, Realization, Classification, Navigation) - (Ontology Processor Component);
- Ontology Annotation Services - (Ontology Navigation, Keywords to Lexicon mapping) - (Ontology Annotation Component);
- Lexicon Alignment Services - (Lexicon Component).

One open question here is the User interface for Cross Lingual Retrieval Component. It will be necessary this component to implement formulation of ontology queries via the ontology or via the lexicons. It is necessary to integrate it together with the other parts of the interface such as keywords navigation, glossary navigation, other metadata navigation.

The ontology language for the OMS of LT4eL will be OWL-DL [19]. The choice is motivated on the basis of the inference available for this sub-language of OWL. Where possible we will use even smaller sublanguages.

The tools that will implement the above services will be based on already existing tools for ontology management. The current list of such tools includes: Protégè (<http://protege.stanford.edu>), Jena (<http://jena.sourceforge.net/>), Pellet (<http://www.mindswap.org/2003/pellet/download.shtml>). Where it is necessary, new functionalities will be implemented on the top of the selected tools. For the ontology annotation the CLaRK System will be adapted (<http://www.bultreebank.org/clark/index.html>). The exchange format with other components of the LT4eL System will be XML based. The program language for OMS is Java (except, probably, for some third-party tools).

Integration into the Learning Management System

The full picture is given in the integration report. We therefore give a rough outline of the integration of the ontological system with the ILIAS Learning Management System (and other potential software clients). The system will provide several tools for ontology storage, management, access and editing. Which tools will be used depending on the use case.

- Each tool will be placed on the ontology management server of the project, provided with a webservice interface and servlet/JSP web interface.
- The Learning Management System (and possible other clients) will communicate with the tools using the webservice interface; test users will use the servlet/JSP interface.
- The tools will receive an ontology name, a query, a language code which specifies the language of the ontology access (if appropriate)
- The tools will return a list of classes and/or relations selected by the query, their names (lexical items) in the given language, the annotation grammars

Appendix: Semantic Annotation

From the perspective of Learning Management Systems the semantic annotation and more

specific ontological annotation in our case concerns only the metadata section of the description of each learning object. In this section some ontological information is stored and later it is used in order to index the learning objects for retrieval. This separation of the metadata from the content of the learning object ensures freedom in the process of annotation of learning objects. There is no requirement the annotation to be anchored to the content of the learning object. The annotator of the learning object can include in the annotation all the concepts and relations he or she decided to be important for the relevant classification of the learning object. But this freedom might become a problem during the annotation process if the annotation is required to be detailed. Then the question which concepts and which relations are really presented in the content becomes important. Therefore, an annotation inside of the content of learning objects is to great extent an obligatory intermediate step in the annotation of the learning objects with ontological information.

Within the project we envisage to do both kinds of annotation. The metadata annotation will be the one used during the retrieval of learning objects from the repository. The content annotation will be used in two ways: (1) as a step to metadata annotation of the learning objects; and (2) as a mechanism to validate the coverage of the ontology. The process of annotation on content level will be semi-automatic. It will comprise the following steps:

- Identification of the text chunk that will be annotated (automatic);
- Assigning of all possible senses (concepts or relations) for the chunk (automatic);
- Sense disambiguation - selection of the appropriate concepts or relations (manual).

The identification of text chunks to be annotated is usually done by some kind of grammar such as regular grammar, cascaded regular grammar, etc. These grammars depend on the ontology with which the learning objects will be annotated, the lexicons aligned to the ontology, the language of the learning object. Generally, the linguistic annotation will facilitate the ontological annotation. For example, 'help' as the name of a command will be annotated with the appropriate ontological information only if the occurrence of 'help' in the text is annotated as a noun, but not a verb. The assigning of all possible senses for a text chunk will be facilitated by the lexicons for the corresponding language. Additionally to the lexicons there will be rules for paraphrasing of text chunks that have the same meaning. The last component is important because we need to identify the cases when a concept is mentioned in the content of the learning object by an expression that is not stored in the lexicon or the grammar for a given language. The last point will be only partially covered, because the problem is huge. The manual selection of the appropriate concept or relation for a given text chunk will depend on the experience of the annotator, the context of the text chunk. In some cases the annotator will need to consult the ontology and some of the lexicons. One problem with the content annotation is the possibility of overlapping of text chunks. Such cases could be solved by several overlapping annotations or by defining a strategy for selecting just one of the competing chunks.

The annotation on metadata level could be done automatically by collecting all content annotations and storing them in the metadata section of the learning object, or manually by inspection of the content annotations and storing only some of them.