ONTOLOGY AND ONTOLOGY ENGINEERING: ANALYSIS OF CONCEPTS, CLASSIFICATIONS AND POTENTIAL USE IN E-LEARNING CONTEXT

Lina Tankelevičienė

February 2008

Technical Report MII-SED-08-01

Institute of Mathematics and Informatics, Akademijos 4, Vilnius LT 08663, Lithuania
http://www.mii.lt
Abstract

Our research is oriented towards possibilities of improving an e-Learning system while integrating modern achievements in the fields of semantic web, in general, and ontology engineering, in particular. In this report, we analyse different definitions of concept Ontology, provide a brief introduction on formalisms (classes, relations, formal axioms, and instances) for knowledge representation on the ontological level. Also we analyse different types of ontologies, singled out according different aspects. We present the analysis of ontology development/representing tools, including ontology languages. Then we draw the potential business value of integrating ontology into an information system, especially accentuate e-learning case.

Keywords: domain ontology, ontology engineering, ontology types, e-learning system.
Contents

1. Introduction .............................................................................................................4

2. The Concept Ontology: Revisited ..............................................................4
   2.1. Ontology Definitions .................................................................5
   2.2. The Structural Parts of an Ontology ..............................................6

3. Classifications of Ontologies .....................................................................7
   3.1. Lightweight and Heavyweight Ontology .........................................7
   3.2. Schema-Ontology and Topic-Ontology ...........................................8
   3.3. Other classifications ...........................................................................9

4. Ontology Engineering ................................................................................9
   4.1. The Scope of the Subject ...............................................................9
   4.2. Choosing the Language for Ontology Modeling/Representation ....10
   4.3. Methodologies and Techniques for Ontology Development .........11

5. Potential Use of Ontology in e-Learning Context ...................................12
   5.1. Types of Integrating of Ontology in IS............................................12
   5.2. Learning in a Rich Virtual Environment: the Demand of more Interactivity on Student-Study Material Level ..............................................................13
   5.3. Using Domain Ontology in Different Phases of e-Learning ........16
   5.4. Supporting Adaptivity and Information Retrieval .......................17
   5.5. Potential Business Value in e-Learning Context .........................19

6. Conclusions ......................................................................................................21

Reference .........................................................................................................21
1. Introduction

Domain ontologies gain its popularity due to ability 1) to implement reuse on knowledge level; and 2) to enhance information retrieval processes. Our research is oriented towards application of ontology in the e-learning domain. Our final goal is concerned with extending the architecture of a typical Learning Management System (further, LMS) by the means of domain ontology and intelligent components.

So, our picked problem domain is e-Learning, strictly speaking, participants and their performed activities. Our solution domain concerns ontology modelling/engineering. The goal, expressed abstractly, is better performance. Therefore, we begin our analysis from two sides:

1) Analysis of the problems in nowadays e-Learning environments, concerning data/information/knowledge resources used and processes, in which knowledge management appear. The purpose of this part is extraction of possibilities for qualitative or quantitative change.

2) Analysis of technologies, tools, and methodologies, concerning ontology modelling/engineering; design of framework for change; design of scenarios for innovative use in e-Learning setting.

In this technical report we analyse ontology definitions, clarify the structural elements of an ontology, summarise different classifications of ontologies and argue for our decision to choose some type. Then we define the scope of ontology engineering and highlight the problems of choosing a) language for modelling/representing domain ontology; and b) methodology for development of domain ontology. Further we analyse types of integration of ontology in information system (further, IS), spotlight the problems in e-Learning context and show the potential matching of ontology engineering and subject domain separating from resources for improvement of e-Learning system.

2. The Concept Ontology: Revisited

Ontology is an emerging instrument for knowledge representation, share, reuse and interoperability. Due to varying understanding of the ontology concept, it is often used by different meanings. Here we analyse Ontology conception in Software Engineering/Semantic Web domains.
2.1. Ontology Definitions

Concept “ontology” was primarily used in philosophy. It is often treated as synonym for “metaphysics“. Ontology is defined in Webster’s Dictionary as follows:
- a branch of metaphysics concerned with the nature and relations of being;
- a particular theory about the nature of being or the kinds of things that have existence.

Etymologically the word is derived from *ont* and *logia* – in Greek language it means *to be* and *science*. Therefore, ontology is often defined as *study of existence*.

Some of the definitions, used in computer science field, are presented in (Guizzardi, 2005). They are summarised and commented in Table 1.

<table>
<thead>
<tr>
<th>Definition</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ontology is “a representation of a conceptual system that is characterized by specific logical properties”.</td>
<td>Accentuates the collection of statements or other semantic definitions for a domain.</td>
</tr>
<tr>
<td>Ontology is “a synonym of conceptualization”.</td>
<td>Emphasizes, that we deal with an abstract, simplified view of the part of the world.</td>
</tr>
<tr>
<td>Ontology is “a special kind of knowledge bases”.</td>
<td>Emphasizes, that ontology is engineering artefact.</td>
</tr>
</tbody>
</table>

We assume the following definition of ontology: *“Ontology is a conceptual specification that describes knowledge about a domain in a manner that is independent of epistemic states and state of affairs“* (Guizzardi, 2007). This definition emphasizes that ontologies are universal models of domains or models of known knowledge in a domain. Therefore, in the context of e-learning, we can distinguish:

1. *Domain level*, which concerns the domain knowledge.
2. *Course level*, which concerns the practical implementation of e-learning. The course consists of a set of learning resources, including both teaching/learning materials and activities.
3. *Technological level*, which deals with learning objects (LO) and information objects. Here we assume that composite LOs also contain learning scenarios.
Such framework allows us also to distinguish between domain engineering and distance study course engineering. Therefore reuse can be employed on a higher level.

Ontologies can be defined formally, which allows for knowledge sharing and automatic reasoning. Ehrig (2004) formally defines ontology on the algebraic level as a structure

\[ O = (C, T, \leq_C, \leq_T, R, A, \sigma_R, \sigma_A, \leq_R, \leq_A, I, V, \iota_C, \iota_T, \iota_R, \iota_A), \]

where:

- \( C, T, R, A, I \) and \( V \) are disjoint sets for concepts, datatypes, relations, attributes, instances and data values;
- partial orders \( \leq_C \) on \( C \) are concept hierarchy, which defines taxonomic structure; similarly, \( \leq_T \) defines type hierarchy, \( \leq_R \) defines relation hierarchy, \( \leq_A \) defines attribute hierarchy;
- functions \( \sigma_R : R \to C^2 \) and \( \sigma_A : R \to C^2 \) are called relation signature and attribute signature;
- a function \( \iota_C \) is called concept instantiation, functions \( \iota_T, \iota_R, \iota_A \) are datatype, relation and attribute instantiations, respectively.

We believe that for the application purposes we can restrict ourselves with a less formal definition, derived from logic area. In our case, ontology must satisfy two main requirements:

1) It must be \textit{formalised} in order a computer could process it;
2) It must be \textit{shared} in order a community of experts of some field could agree with it and use it in applications.

Therefore, we adopt a formal definition of ontology from (Guizzardi, 2005). Ontology is a 4-tuple \(<C, R, I, A>\), where \( C \) is a set of \textit{classes (concepts)}, \( R \) is a set of \textit{relations}, \( I \) is a set of \textit{instances}, and \( A \) is a set of \textit{axioms}. The mentioned structural parts of ontology will be described more widely in next chapter.

\textbf{2.2. The Structural Parts of an Ontology}

The structural parts of an ontology are as follows:

\textbf{Classes} (other synonymous terms: \textit{concepts}, \textit{categories}, \textit{types}) represent important concepts of the domain. In the domain \textit{e-Learning Tools}, the important concepts, e.g., are Software Product, Animation Development Tool, Purpose,
Curriculum Level, etc. Classes in the ontology are usually organised in taxonomies, where generalisation-specification mechanisms are applied.

**Relations** (properties, slots, attributes, roles) represent associations between the concepts of a domain. Most often the *is-a* and *consist-of* relationships are used. However, the taxonomical structure is not the only one possible. Ontologies usually contain binary relations, where the first argument is called the domain, and the second one is called the range. The attributes are sometimes distinguished from relations. The difference between attributes and relations is that the range of an attribute is a datatype, not a class.

**Instances** (individuals) represent individuals in ontology. Instances can be defined in ontology or in database of factual data.

**Formal axioms** are used for expressing propositions that are always true, e.g., in the e-learning course the same person can not be in the lecturer and in the student role at the same time. Formal axioms are used to infer new knowledge. If axioms are not included into ontology itself, reasoning mechanisms must be implemented in program side of the system (in the code).

### 3. Classifications of Ontologies

Literature review shows that there are different classifications of ontologies. Some of them concern different aspects, e.g. generality level, richness; some of them are alternatives of some kind. Different classifications are comprehensively analysed in (Ruiz, 2006). Here we highlight more important in our viewpoint classifications, which enforce us to choose one alternative, and briefly summarise the other classifications.

#### 3.1. Lightweight and Heavyweight Ontology

There can be two types of ontologies, depending upon a language used for formalisation and the purpose of ontology: lightweight and heavyweight. **Lightweight ontologies** include concepts with properties and taxonomies, but do not include axioms. **Heavyweight ontologies** are richer in expressiveness, but they are harder to manage.

Since the lightweight ontologies are less restrictive, they are usually wider acceptable, which is very important for knowledge sharing and reuse. We are planning to find balance between expressiveness and decidability. The less
expressiveness the language provides, the better reasoning mechanisms are implemented. This is very important in the context of immediate feedback generation and increasing efficiency of system in common and simple tasks. Differences between lightweight and heavyweight ontologies are listed in Table 2.

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Lightweight ontology</th>
<th>Heavyweight ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural components</td>
<td>Classes, relations, instances (not mandatory)</td>
<td>Classes, relations, instances (not mandatory), formal axioms</td>
</tr>
<tr>
<td>Expressiveness</td>
<td>Lower</td>
<td>Higher</td>
</tr>
<tr>
<td>Manageability</td>
<td>Easier</td>
<td>Harder</td>
</tr>
<tr>
<td>Constraints</td>
<td>Less</td>
<td>More</td>
</tr>
<tr>
<td>Applicability</td>
<td>Wider</td>
<td>More narrow</td>
</tr>
</tbody>
</table>

**3.2. Schema-Ontology and Topic-Ontology**

Other classification of ontology as schema-ontology and topic-ontology is introduced in (Kiryakov, 2006). The author accentuates the possibility to formalise the domain while using set-theoretical model and set theoretical operations. The main differences of both types are given in Table 3.

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Schema-ontology</th>
<th>Topic-ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>The base for defining (explaining) semantics</td>
<td>Set theory</td>
<td>Empiricism</td>
</tr>
<tr>
<td>Semantics type</td>
<td>Extensional</td>
<td>Intensional</td>
</tr>
<tr>
<td>Instances as structural part</td>
<td>The set of instances represents the semantics of class.</td>
<td>Don’t influence on sub-classing relations of classes.</td>
</tr>
</tbody>
</table>

We choose schema-ontology for capturing subject domain knowledge, because:

- It better corresponds with our understanding of the concept of ontology;
- It deals with formal or semiformal representation, it represents a top-down systematic approach;
- It better fits in our instructor-led e-Learning context.
3.3. Other classifications

The rest classifications are summarised in Table 4.

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Proposed by</th>
<th>Types mentioned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generality level</td>
<td>Guarino (1998)</td>
<td>- High–level ontologies&lt;br&gt;- Domain ontologies&lt;br&gt;- Task ontologies&lt;br&gt;- Application ontologies</td>
</tr>
<tr>
<td></td>
<td>Fensel (2003)</td>
<td>- Generic or common-sense ontologies&lt;br&gt;- Representational ontologies&lt;br&gt;- Domain ontologies&lt;br&gt;- Method and task ontologies</td>
</tr>
<tr>
<td>Type of conceptualisation structure</td>
<td>Van Heist (1997)</td>
<td>- Terminological ontologies specify the terms that are used to represent knowledge in the domain.&lt;br&gt;- Information ontologies specify the record structure of databases.&lt;br&gt;- Knowledge modeling ontologies specify conceptualizations of the knowledge.</td>
</tr>
<tr>
<td>Richness of internal structure</td>
<td>Lassila (2001)</td>
<td>- Controlled vocabulary - a finite list of terms.&lt;br&gt;- Glossary - a list of terms and meanings.&lt;br&gt;- Thesauri - provides some additional semantics.&lt;br&gt;- Term hierarchies (or informal hierarchies) - provides generalisation and specialization.&lt;br&gt;- Strict subclass hierarchies (or formal hierarchies).&lt;br&gt;- Frames – includes property information.&lt;br&gt;- Ontology with value restrictions.&lt;br&gt;- Ontology with logical constraints.</td>
</tr>
<tr>
<td>Formality (depends on formality level of language used)</td>
<td>Uschold (1996)</td>
<td>- Formal, e.g., expressed in first-order logic.&lt;br&gt;- Informal, e.g., expressed in natural language.&lt;br&gt;- Semi-formal, e.g., expressed in UML.</td>
</tr>
</tbody>
</table>

4. Ontology Engineering

4.1. The Scope of the Subject

The topics of ontology engineering are comprehensively described in (Guarino, 1998; Devedzic, 2002; Corcho, 2007), including methods and methodologies for the development of ontologies, ontology development process and
lifecycle, ontology tools and languages. The importance of ontology engineering is recognized, but applications of ontology are still analysed more widely than ontology engineering itself. The attention is made to the following aspects: ontology learning, knowledge acquisition process, ontology merging, alignment, evaluation.

### 4.2. Choosing the Language for Ontology Modeling/Representation

The first ontology languages were introduced at about 1990. They were basically based on first-order logic (e.g., Knowledge Interchange Format – KIF) or frames (e.g., Ontolingua, Loom, CycL). The spread of Internet influenced other generation of ontology languages, so called Web-based ontology languages, including SHOE, XOL, DAML+OIL, RDF, RDF Schema, OWL. The last one, Ontology Web Language (OWL) was proposed as W3C recommendation (http://www.w3.org/TR/owl-features/) and now it is actively used.

From the other hand, in the software engineering field Unified Modeling Language (UML), raised from object-oriented methodologies, is very popular and widely adopted.

Arguments for choosing one from two variants are listed in Table 5.

<table>
<thead>
<tr>
<th>OWL (according to Mika, 2007)</th>
<th>UML (according to Kogut, 2001; Mika, 2007)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Less constraints, that means, language provides more expressiveness (e.g. disjointness, union, intersection and equivalence of classes).</td>
<td>- UML is an open standard maintained by the OMG. UML has standard mechanisms for defining extensions;</td>
</tr>
<tr>
<td>- Properties are global and can be used with multiple classes.</td>
<td>- UML is widely adopted in industry;</td>
</tr>
<tr>
<td>- Properties can be easily defined as subproperties.</td>
<td>- It is well known outside the AI and Semantic Web community;</td>
</tr>
</tbody>
</table>

The main impact on our choice is made by our final goal. UML is oriented and is better for ontology modelling, and OWL – for ontology engineering. We seek to final realisation of solution proposed, therefore, we choose OWL. The final goal also influences the requirements to expressiveness of language for ontology modelling.
According to mentioned expressiveness levels, OWL provides three sublanguages:

1) OWL Lite supports classification hierarchies and simple constraints. It is intended to be easy to support in different tools.

2) OWL DL supports more expressiveness, but still supports computational completeness and decidability. OWL DL allow us to use all possible OWL language constructs, but some uses cases are additionally restricted (for example, a class cannot be an instance of another class).

3) OWL Full provides maximum expressiveness with no computational guarantees.

4.3. Methodologies and Techniques for Ontology Development

There are different methodologies for ontology development (summarised and compared in Lopez, 1999). One of the most cited is methodology of Uschold and Gruninger (Uschold, 1996). It defined four main phases:

1) Identifying a purpose and scope;
2) Building the ontology: ontology capture, ontology coding, integrating existing ontologies;
3) Evaluation (verification and validation);
4) Documentation.

Authors also provide with initial guidelines for ontology development: clarity for effective communications in future, coherence, extensibility, minimal ontological commitment. General steps involved in constructing ontology are also listed in (Leuf, 2006):

1) Acquire the domain knowledge;
2) Design the conceptual structure;
3) Develop the appropriate detail;
4) Verify;
5) Commit.

These ontology development steps in their essence are closely related with common software development processes. Ontology development processes also must deal with common software engineering principles: incrementality, modularity and decomposition, generality. Employing these principles implies achievement of qualities of an ontology such as correctness, usability, and reusability.
Other methodologies are based on their authors practical works, and sometimes even tools used impact on them. E.g., Gavrilova (2005) proposes 5-steps recipe: 1) Glossary development; 2) Laddering; 3) Disintegration; 4) Categorization; 5) Refinement.

Boyce (2007) defines general stages as follows: 1) Determining the domain and source and also purpose and scope of the ontology; 2) Ascertaining if any ontology has been developed in the same subject area previously; 3) Enumerating important domain terms; 4) Defining class/concept hierarchy; 5) Defining concept properties (internal structure); 6) Attaching facets to the properties, that is: describing value type, cardinality, etc; 7) Creating instances of the classes.

Noy (2001) describes the process for developing ontology in following steps: 1) Determine the domain and scope of the ontology. 2) Consider reusing existing ontologies. 3) Enumerate important terms in the ontology. 3) Define the classes and the class hierarchy. 4) Define the properties of classes – slots. 5) Define the facets of the slots. 6) Create instances. The steps are oriented towards application of Protégé tool.

Summarising, we can state that the first steps in the ontology development are acquiring the domain knowledge (specification) and designing the conceptual structure (conceptualisation). Next, the important concepts in ontology are enumerated and categorized, and the relationships between concepts are defined as a class/concept hierarchy. Next, the properties (attributes) of the classes are defined. The last step is the creation of the instances of classes.

5. Potential Use of Ontology in e-Learning Context

5.1. Types of Integrating of Ontology in IS

Ontology can be used for IS and in IS differently. Guarino (1998) separated two dimensions for describing the impact of ontology on IS:

- A temporal dimension, concerning whether an ontology is used at development time or at run time;
- A structural dimension, concerning the particular way an ontology can affect the main IS components.
According to temporal dimension, when an ontology is used at development time, we deal with ontology-driven development of IS, and in the second case – with ontology-driven IS.

Additionally, when an ontology is used at run time, Guarino (1998) distinguish two more cases:

- Ontology-aware IS: it is known about the existence of a (possibly remote) ontology and IS component can use it. Ontology is treated as information resource.
- Ontology-driven IS: the ontology is one of the components of the system, which correspond some requirements while reaching overall IS functionality. Ontology is treated as architectural artefact.

Our solution supposes treating ontology as architectural artefact. The conceptual approach for extending existing LMS architecture by intelligent and deeper knowledge layers was proposed in (Dzemydiene, 2006a; Dzemydiene, 2006b). The main idea was to extend the typical LMS by two layers:

- Intelligent layer – intelligent decision support components – which must act as a mediator between the core LMS elements and different types of user interfaces.
- Deeper knowledge layer - domain, users’, learning design ontologies – which can act as foundation for adaptive educational sequencing.

5.2. Learning in a Rich Virtual Environment: the Demand of more Interactivity on Student-Study Material Level

Simplified model of learning as the process of acquiring experience can be depicted as in Fig. 1. Phases also can be named as introduce, apply, summarise.

Figure 1. Learning process (according to Horton, 2006)
In the modern society it is postulated, that absorbing knowledge happens using different sources: lecturers, colleagues, virtual communities, libraries, internet, etc. Also the role of learner is emphasized. In the same time the role of the lecturer transforms into the collection of the following: consultant, expert, facilitator, mentor, etc.

Interactivity is one of the criteria or indicators showing quality of distance studies (Karoulis, 2003). This author emphasizes interactivity with the instructional material, which is described as navigational fidelity, multimedia components, multiple kinds of exercises, facilitation of the active interaction, support for collaborative work and group dynamics. Students’ support is formulated as other indicator and it concerns guidance and encouragement of the student both from the instructional material and from the communication channels, accessibility to the tutor, instructional organisation.

Interaction means primarily the communication between the user and the system. In the interaction framework of Donald Norman (Alan J. Dix, 1993) the following users’ activities are concerned: 1) User establishes the goal; 2) Formulates intention; 3) Specifies actions at interface; 4) Executes action; 5) Perceives system state; 6) Interprets system state; 7) Evaluates system state with respect to goal. There are two different possibilities to support user in completing their tasks. The system can help him to deal with solving complicated problem. Special tricky engineering methods can be employed. Just the opposite, system can support user in performing simple but very often necessary tasks.

Four categories of interaction on student-study material level were distinguished in (Stouppe, 2003):

1) Enriching: Interactions that enable learners to access information but do not convey information in and of themselves.
2) Supportive: Interactions that directly assist the learner in understanding material.
3) Conveyance: Interactions that demonstrate the concept being taught or in some other way let learners try out their knowledge.
4) Constructive: Interactions that call the learner to organize and map their own knowledge and understanding.
Interaction can be not only among student and study material. Students communicate and collaborate among themselves and with lecture. On the other hand, these processes must be additionally supported.

Here we clearly state and analyze some problems, which are seeking partially to solve, to decrease its negative impact.

1) Increasing effectiveness of workload. For us it is actually important not the efficiency of the system, but efficiency of activities of its user, both lecturer and student. Students need different kinds of support: technical, administrative, subject oriented, motivational. Humans, so called instructors, are necessary for supporting students. In order to achieve good learning results, a big amount of workload time is necessary, because instructors supervise and direct the learning process, evaluates not only final achievements, but also intermediate contributions of the learner, the degree of progress. Ontology-based description of required knowledge concludes assumption to shift a part of collaboration processes into student-study material level. Therefore, some functions can be detached from instructor and attached to a computer system. It is important to pick for transfer time-intensive, frequently repeated, maybe not complicated functions. In this way effectiveness will show itself in organisational level, too. Misleading opinion is that self-supporting studies usually happen effectively. For this type of studies people with very strong motivation are necessary. Despite of very big amount of information, it is difficult to find, what is useful. Therefore ontologies and software components, for example agents, let us present study material in a convenient way for learner, avoiding information overload, adapting study material to learning style of learner, readiness of a learner. Practically learning scenarios can be generates and thus personalised learning implemented. This fact resounds modern ideas of individualised learning.

2) Increasing of satisfaction. Achievements of students depend on their satisfaction during gaining learning experience, too.

3) Adaptivity in dynamic context. Adaptive systems are concerned as systems, which offer dynamically built and automatically performed personalisation. Learning materials perform changes in time. New learning objects come; some resources replace the other, some supplement.
Relevant problems are analysed in (Dignum, 2003) and the objectives of knowledge management technologies, considering ontology modelling is one of them, are stated as follows:

- Assist people to generate and apply “just in time” and “just enough” knowledge, prevent information overload and stimulate sharing of relevant knowledge in a dynamic, collaborative environment.
- Preserve individual autonomy and contribute to the creation of an atmosphere of trust between participants.

In summary we can state that the same problems exist in different subject domain, not only e-learning, where users’ work intensively with big amount of information. Therefore, supposed business value of spreading semantic web technologies, and ontologies among them, concerns better support for user in information-intensive environments, including e-Learning.

5.3. Using Domain Ontology in Different Phases of e-Learning

Domain conceptualisation in e-learning system can be represented by the means of: a) Subject domain ontology; b) Instructional ontology (which covers the teaching/learning process).

As context we consider real-time information, in the case of development and delivery of distance studies, their physical, technical/technological, organizational, management, pedagogical, social, ethical/cultural aspects. A big part of context related information is stored in core databases of LMS. Additional resources and services can be discovered using web services technology.

In the global view ontology can be used in the processes, listed in Table 6.

<table>
<thead>
<tr>
<th>Entitlement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development of learning materials/activities using</td>
<td>Constitutes from two different parts:</td>
</tr>
<tr>
<td>subject domain and instructional ontology (optional).</td>
<td>- Content and services discovery and assembly.</td>
</tr>
<tr>
<td></td>
<td>- Development of their own materials and/or activities.</td>
</tr>
<tr>
<td></td>
<td>Tasks that can be realised: verifying completeness, timeliness, compatibility, linking of learning materials.</td>
</tr>
<tr>
<td>Teaching/learning process.</td>
<td>Includes adaptive course delivery, adaptive sequencing, adaptive presentation, adaptive interaction, and adaptive support. For example:</td>
</tr>
</tbody>
</table>
Adapting educational resources for students with different levels of learning progress.
- Different starting point. Concepts belong to overlapped groups, which are important differently to different students.
- Concept mapping, development of topics map (as evaluated task).

5.4. Supporting Adaptnity and Information Retrieval

The definitions of adaptive systems are not similar in different sources. Paramythis defines adaptive learning system as “system, which is capable of: monitoring the activities of its users; interpreting these on the basis of domain-specific models; inferring user requirements and preferences out of the interpreted activities, appropriately representing these in associated models; and, finally, acting upon the available knowledge on its users and the subject matter at hand, to dynamically facilitate the learning process” (Paramythis, 2004).

Gaudioso (Gaudioso, 2005) differentiates definitions of adaptable and adaptive systems. According him “Adaptable systems are systems, which offer personalization that is pre-defined before the execution of the system and that may be modified by users” and “Adaptive systems are systems, which offer personalization that is dynamically built and automatically performed based on what these systems learn about the users”.

More generally adaptivity in psychology and computer science is understood as the ability of a system to adjust to the changing environment, or, in the other words, in context.

Often adaptivity is analysed in the context of learning, oriented towards knowledge transfer. For example: “The main goal of adaptation in educational systems is to guide the students through the course material in order to improve the effectiveness of the learning process” (Gaudioso, 2005). That means, constructivistic view is not taken into account.

Users’ modelling is often accentuated while speaking about adaptivity. Adaptive e-learning systems incorporate knowledge about user’s goals, knowledge level, interests, stereotypes, cognitive preferences, learning styles (Cristea, 2004). But some of the mentioned information can be found in the core databases of LMS and used without extra student modelling. Therefore, we restrict ourselves to subject domain modelling.
For example, having subject domain ontology and information about student’s goals we can adapt educational resources for students with different levels of learning progress (see Fig. 2):

![Diagram showing division/extraction of abstract subject areas](image)

**Fig. 2.** Example of division/extraction of abstract subject areas

The next figure (see Fig. 3) illustrates very popular model for adaptive educational hypermedia sequencing, proposed in (Karampiperis, 2005):

![Diagram showing abstraction layers of adaptive educational hypermedia sequencing](image)

**Fig. 3.** Abstraction Layers of Adaptive Educational Hypermedia Sequencing (by Karampiperis, 2005)
Simple but frequent tasks also are worthy to support. During e-Learning process users face knowledge intensive work. Typical users of e-Learning system and their functions are listed in Table 7.

**Table 7. Users and their functions (according to Tankelevičienė, 2003)**

<table>
<thead>
<tr>
<th>User type</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>System administrator</td>
<td>Controls the system, users, courses, defines users’ rights, adapts system to separate computer or informs users, how to adapt computer to system (what to install, what values of parameters to change).</td>
</tr>
<tr>
<td>Author</td>
<td>Creates new courses, while using previously accumulates materials and forming new learning content, renews content, provide learning scenarios.</td>
</tr>
<tr>
<td>Instructor</td>
<td>Observes learning process, analyzes students’ achievements, consults students, direct students to the proper direct, evaluates tasks. Usually authors become instructors of the course.</td>
</tr>
<tr>
<td>Student</td>
<td>Picks the course according to his/her goals, seeks for realisation of defined goals, connects to the system using user name and password, studies material, evaluates progress, completes tasks, collaborates with others, etc.</td>
</tr>
</tbody>
</table>

We can see, that there are plenty activities, concerning browsing and searching. Simplifying, searching can be treated as goal-directed search, and browsing – as explorative search. Modern teaching methods require supporting both searching types. Therefore, it will be valuable for the user to support them in these activities.

### 5.5. Potential Business Value in e-Learning Context

Distance learning course, as a specific form of e-learning, can be defined as a module of study programme, prepared according to distance learning methodology and delivered in a virtual learning environment using information and communication technologies. Both study module and distance learning course are based on study goals and expected competences. Therefore, subject domain ontology can be developed despite of learning resources and services, which may vary at different moments.

We separate two sources of inputs for adaptation process: domain models and context (Fig. 4).
Domain conceptualisation is represented by the means of: a) Subject domain ontology; b) Instructional ontology (which covers the teaching/learning process).

As context we consider real-time information, in the case of development and delivery of distance studies, their physical, technical/technological, organizational, management, pedagogical, social, ethical/cultural aspects. A big part of context related information is stored in core databases of LMS. Additional resources and services can be discovered using web services technology.

Usually the following benefits of using ontology are listed:
- Ontology presents common conception about domain.
- Ontology lets communicate humans and computers.
- Ontology supports knowledge sharing and reuse.

The utilities of ontologies are also listed in (Ruiz, 2006):
- Clarify the knowledge structure.
- Reduce conceptual and terminological ambiguity.
- Allow the sharing of knowledge.

Davis (2007) characterises the business value of semantic technologies in five critical areas:
- Development – automation in different development steps;
- Infrastructure – enablement and orchestration of core resources;
- Information – semantic interoperability of information and applications in real context;
- Knowledge – knowledge work automation and supporting knowledge workers;
- Behavior – systems knows what they are doing.
Technology itself (Semantic Web and ontologies in particular) provides only with increased possibilities. Therefore, methods, frameworks and tools are necessary for realising practically all mentioned.

6. Conclusions

The concept Ontology still has different meanings among researchers and practitioners in Software engineering/Semantic Web areas. The conception partially depends on the fact, modelling or engineering of ontology is more accentuated.

The methodologies for the manual development of ontology are underdeveloped. There are still much heuristics in the development process.

The approach to using domain ontology in the development and delivery of educational resources enables support for this process, increasing adaptivity and interactivity on student-study material level.

The development and research of adaptive learning environments is struggled with the high investment for development and maintenance problem. Besides other, the lack of interoperability in the field of knowledge management (knowledge discovery, knowledge representation, and reasoning) can be mentioned.

The domain centric learning, in generally, and our approach, in particularly, are more suitable for formal studies, where subjects are unambiguously specified.

Reference


