SWEL’07: Ontologies and Semantic Web Services for Intelligent Distributed Educational Systems

Supplementary Proceedings
of the 13th International Conference of Artificial Intelligence in Education
(AIED 2007)

July 9th, 2007
Marina del Rey, CA, USA
Preface

The SWEL’07 workshop at AIED’07 aims to address issues related to using Semantic Web based knowledge representation and Grid technologies for the design and development of intelligent distributed educational systems. It covers topics related to the use of ontologies, Semantic Web standards, and distributed processes enabled by Grid technologies for learning content delivery, services and knowledge components specification, effective intelligent courseware construction, and learner modelling.

SWEL’07 is the fifth in a series, following the successful edition in 2002, held in conjunction with ICCE’02, the three sessions of SWEL’04 in conjunction with ITS’04, AH’04, and ISWC’04, the three sessions of SWEL’05 in conjunction with AIED’05, ICALT’05, and K-CAP’05, and SWEL’06 held in conjunction with AH’06.

This edition has a special focus on service-oriented Semantic Grid Architectures for e-Learning and is organized together with two SIGs of the European Network of Excellence Kaleidoscope “Artificial Intelligence and Education” and “Learning Grid”.

Two sessions are included in the workshop focused on the following topics.

1. Session on Ontologies and Semantic Web Standards for e-Learning:
   - Building ontologies for e-learning; theoretical issues in ontology engineering.
   - Using ontologies and Semantic Web standards for structuring, representing, indexing, and retrieving shareable and interoperable learning resources.
   - Using ontologies and SW standards for supporting authoring of intelligent educational systems.
   - Using SW-based contexts for personalization of e-learning applications.

2. Session on Semantic Grid Architectures and e-Learning Services:
   - Semantic Grid architectures and systems for distributed intelligent e-learning.
   - Models and tools for the semantic annotation of e-learning services.
   - Automatic composition of semantically described services and processes.
   - Pedagogical approaches and learning models for exploiting the potential of distributed and Grid technologies for e-learning.
   - Application scenarios for dynamic composition of distributed learning services using AIED and GRID technologies and techniques.

The workshop features a keynote talk “Inside a theory-aware and standards-compliant authoring system” by Riichiro Mizoguchi.

July, 2007
Darina Dicheva, Riichiro Mizoguchi, Nicola Capuano and Andreas Harrer
Workshop Committee

Co-Chairs
Nicola Capuano, University Of Salerno, Italy
Darina Dicheva, Winston-Salem State University, USA
Andreas Harrer, University Of Duisburg-Essen, Germany
Riichiro Mizoguchi, University Of Osaka, Japan

International Program Committee
Lora Aroyo, Free University Amsterdam, The Netherlands
Miguel Bote-Lorenzo, University of Valladolid, Spain
Jacqueline Bourdeau, Télé-université, Université du Québec, Canada
Paul Brna, University of Glasgow, UK
Weiqin Chen, University of Bergen, Norway
Hugh Davis, University of Southampton, UK
Thanasis Daradoumis, Open University of Catalonia, Spain
Cyrille Desmoulins, University of Grenoble, France
Vladan Devedzic, University of Belgrade, Serbia and Montenegro
Yannis Dimitriadis, University of Valladolid, Spain
Vanja Dimitrova, University of Leeds, UK
Martin Dzbor, Knowledge Media Institute, UK
Peter Dolog, Aalborg University, Denmark
Angelo Gaeta, Centro di Ricerca in Matematica Pura ed Applicata, Italy
Dragan Gasevic, Simon Fraser University, Canada
Monique Grandbastien, LORIA, France
Marek Hatala, Simon Fraser University, Canada
Yusuke Hayashi, University of Osaka, Japan
Nicola Henze, University of Hanover, Germany
Mitsuru Ikeda, JAIST, Japan
Judy Kay, University of Sydney, Australia
Kinshuk, Athabasca University, Canada
Rob Koper, Open University, NL
Frank Linton, MITRE, USA
Miltiadis Lytras, Athens University of Economics and Business, Greece
Gordon McCalla, University of Saskatchewan, Canada
Tanja Mitrovic, University of Canterbury, New Zealand
Ambjörn Naeve, Royal Institute of Technology, Sweden
Hiroaki Ogata, Tokushima University, Japan
Alexandra Poulavassilis, University of London, UK
Niels Pinkwart, Clausthal University of Technology, Germany
Pierluigi Ritrovato, Università di Salerno, Italy
Demetrios Sampson, Center for Research and Technology - Hellas, Greece
Miguel-Angel Sicilia, University of Alcalá, Spain
Sergey Sosnovsky, University of Pittsburgh, USA
Maria Felisa Verdejo, Univ. Nacional de Educación a Distancia, Spain

SWEL website:
http://compsci.wssu.edu/iis/swel/swel07/swel07-aied07.html
# Table of Contents

Inside Theory-Aware and Standards-Compliant Authoring System  
*Riichiro Mizoguchi, Yusuke Hayashi and Jacqueline Bourdeau*  
1

A Grid Based IMS Learning Design Player: the ELeGi Case Study  
*Nicola Capuano, Roberto Iannone, Sergio Miranda and Marcello Rosciano*  
19

Ontologies, Applications Integration and Support to Users in Learning Objects Repositories  
*Aude Dufresne and Mohamed Rouatbi*  
30

Exploring Semantic Description and Matching Technologies for Enhancing the Automatic Composition of Grid based Learning Services  
*Gustavo Gutiérrez-Carreón, Thanasis Daradoumis and Josep Jorba*  
36

The Unfolding of Learning Theories: Its Application to Effective Design of Collaborative Learning  
*Seiji Isotani and Riichiro Mizoguchi*  
44

Ontological Framework for Educational Feedback  
*Jelena Jovanović, Shilpi Rao, Dragan Gašević, Vladan Devedžić and Marek Hatala*  
54

Mobile Topic Maps for e-Learning  
*John Mcdonald and Darina Dicheva*  
65

Scenarios for a Learning GRID  
*Agathe Merceron, Nicola Capuano, Francesco Orciuoli and Pierluigi Ritrovato*  
74

Supporting Philosophers’ Work through the Semantic Web: Ontological Issues  
*Michele Pasin and Enrico Motta*  
80

Community-oriented Course Authoring to Support Topic-based Student Modeling  
*Sergey Sosnovsky, Michael Yudelso and Peter Brusilovsky*  
91

Using SWRL for ITS through Keyword Extensions and Rewrite Meta-Rules  
*Eric WANG and Yong Se KIM*  
101

Learning a Domain Ontology in the Knowledge Puzzle Project  
*Amal Zouaq, Roger Nkambou and Claude Frasson*  
106

Personal Assistant towards Semantic Information Retrieval  
*Zeeshan Ahmed and Detlef Gerhard*  
115
Inside Theory-Aware and Standards-Compliant Authoring System

Riichiro MIZOGUCHI\textsuperscript{a}, Yusuke HAYASHI\textsuperscript{a}, Jacqueline BOURDEAU\textsuperscript{b}
\textsuperscript{a} ISIR, Osaka University, \textsuperscript{b} LICEF, Télé-université
\textsuperscript{a} 8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan
\textsuperscript{b} 100 Sherbrooke W., Montréal, (QC) H2X 3P2 Canada

Abstract. In their paper\textsuperscript{[14]}, Bourdeau and Mizoguchi foresaw a framework for ontology-based intelligent systems. Although it took longer years than their expectation, the ontology they have been developing is now released for evaluation with the help of the second author. Ontology building is a labor-intensive process and it is rarely perfect. Our enterprise is not an exception. The current ontology is still very preliminary because it has been completely reconstructed from the existing one with a few new ideas. So, we hope the readers be generous when they read the ontology. The ontology presented here is not a light-weight ontology but a heavy-weight ontology. It is built based on philosophical consideration of all the concepts necessary for understanding learning, instruction and instructional design. Although it is full of axioms, the Hozo GUI which is based on a frame structure makes it easier to read it. However, the readers are expected to have basic knowledge of ontology and preferably be aware of the theory of role and of the Hozo way of role representation. Papers \textsuperscript{[6]}[7] would be helpful to grasp what we are doing with this ontology. The prototype system named SMARTIES is a totally ontology-aware system which fully utilizes the merits of ontology computationally as well as conceptually. It is so preliminary that it cannot be open to public, though you can get a rough idea of what it is from the papers.

1. Introduction

Nowadays standard technologies play an important role in the development and delivery of learning contents. Standard technologies provide stakeholders with great benefits; there is however a lack of pedagogical justification of standard-compliant contents. This project focuses on educational theories as a kind of pedagogical knowledge and aims at building an information system that helps users to utilize them for instructional and learning design (Here, the term “instruction” does not have a narrow definition such as lecture but has a broad definition to include anything that fosters or suggest learning in a learning environment). This project takes an ontological engineering approach to grasp fundamental concepts of learning and instruction in order to enable information systems to be aware of the theories on the basis of such concepts.

This article introduces a comprehensive ontology which covers different theories and paradigms about instructional and learning design\textsuperscript{1}. Note that this ontology is still a tentative result of our project. We have plans in the future to continue to make further refinement on it and we welcome your contribution to the refinement. In addition, at the current moment, this ontology focuses only on the abstract design of learning contents and has not been yet related to domain knowledge or learning objects to concretize the abstract design. This is one of the future plans of this project.

1.1 Scope: Objectives and non-Objectives

This project’s objectives include:

- To find an engineering approximation that allows building of an engineering infrastructure that enables instructional designers and educational practitioners to apply knowledge derived from educational theories, and

\textsuperscript{1} This ontology is open to the public on our web site: http://edont.qee.jp/omnibus/.
• To establish a method for building theory-aware systems for education.
This project’s objectives do not include:
• To insist on the scientific validity of the proposed framework for organizing
educational theories,
• To reconstruct existing educational theories on such a basis, nor
• To create new theories

1.2 Expected outcomes

a) Providing a sharable model of instructional design knowledge
Instructional design knowledge includes theoretical knowledge such as instructional
and learning theories as well as empirical one such as heuristics and best practices. The
ontology introduced in this document includes about 100 pieces of “WAY-
knowledge” based on some theories.

b) Increasing theory-awareness in authoring tools
Based on the ontology, an authoring tool can become aware of instructional knowledge
and help authors. This project has developed “SMARTIES”: a prototype system of a
theory-aware authoring tool based on the ontology. This prototype system provides
support functions for making learning/instructional scenarios based on this ontology.
To put it more completely, this system provides a modeling environment and
guidelines for making theory-compliant learning/instructional scenarios. This system
has the flexibility of ontologies. The upper level concepts are built-in but the lower
level concepts can be imported from the ontology built in Hozo² [12].

c) Linking standard-technologies to instructional design knowledge
Instructional design knowledge is expected to enhance the educational justification of
standard-compliant contents. The prototype system supports authors in building
standard-compliant scenarios with theoretical justification because it can export a
theory-based instructional/learning scenario model into IMS LD level A format [9].

1.3 Current state and Future plans

a) Updating this ontology
The current version of this ontology is the version 1.0. Comments received and
discussion done on this workshop will be reflected in the modification as much as we
can. In addition, we want to continue the discussion about ontologies of instructional
design on our website.

b) Expansion of this ontology
We are planning to include concepts related to instructional design theories/processes
and CSCL. In addition, we currently plan to put this ontology to the core of our
infrastructure for instructional design knowledge sharing. This means that the ontology
enables us to utilize theories for contents design through the top-down approach as
well as to build new theories and to share best practices through the bottom-up
approach. We believe such an infrastructure will harmonize theory and practice of
instructional and learning design.

c) Link with learning objects and learning object metadata
In order to implement such abstract design as our scenarios, it is necessary to link it to
learning objects. We have defined attributes of learning and instruction in the proposed
ontology. To consider the relation between these attributes and learning object
metadata (e.g. IEEE LOM [8]) is one of the future directions of this study.

² Hozo ontology editor can be downloaded from http://www.hozo.jp
2. Building a comprehensive ontology for educational theories

The objective of this study is building a comprehensive ontology which covers different theories and paradigms. In this section we briefly discuss the problems and our approach for building such an ontology.

2.1 Basic consideration

In building such a comprehensive ontology, there is one big problem. It is considered that, in the first place, each paradigm or theory has its own definition of “Learning” and hence we cannot organize variety of theories on a common basis. However, for example, Reigeluth [17][18] and Ertmer [4] give some observations about commonality and difference among paradigms and theories. We can summarize these statements as follows; Every theory has some sort of common basis for explaining learning and instruction, and while the assumed mechanism of developing knowledge is different for each paradigm, the idea of states in the learning process is common. In a similar line of the thought, this study sets up a working hypothesis that there must be an engineering approximation of the states where we can conceptualize “Learning” in terms of state change of learners [14].

Note that the purpose of this study is not to expose a scientifically valid basis for organizing theories nor to reconstruct them on this basis, but rather to find an engineering approximation that allows building of an engineering infrastructure that enables practitioners to utilize instructional and learning theories. This paper thus proposes a foundation from the view point of ontological engineering based on the results of previous research in this respect [1][14][16].

2.2 Conceptualization of the interaction between learning and instruction

We have defined a concept that we named I_L event as shown in Figure 1. An I_L event is a concept to link instructional events to learning events. In this study a learning event is composed of state-change and learning action. Learning actions cause the change of learner’s state. On the other hand, an instructional event is composed of an instructional action which affects learning events. The key points of our conceptualization include to emphasize the relations among these three and to model a contribution of instructional action on the change of learner’s state.

2.3 Conceptualization of the abstract structure of instructional/learning scenario

In our modeling framework, a scenario can be modeled as a hierarchical structure of I_L events for achieving a certain change of a learner state. We call it an “I_L event decomposition tree”. The basic idea of an I_L event decomposition tree is to relate a macro-I_L event to the lower (micro) ones that collectively achieve the upper (macro) I_L event as a way of achievement of the change of a learner state (referred to just as “WAY” hereafter). Figure 2 illustrates an example of an I_L event decomposition. This shows that there are two WAYs to achieve the macro-I_L event, which is to introduce a content for making a learner recognize it. WAY1 is based on Gagne and Briggs’s theory [5]. This firstly presents what to learn and then gives guidelines. The other is based on Collins’ [2]. This gives only demonstrations and no explanations. In this case the macro-I_L event is not decomposed.
but concretized. These ways can be thought to have the same goal but achieve it by different strategies. Such relation between WAYs is described by OR relation like between WAY1 and WAY2. In the proposed modeling framework, a scenario is described as a sequence of the leaves in the tree structure and the tree-structure model layered multiply with WAYs accounts for the design intention.

2.4 Conceptualization of strategies suggested by instructional/learning theories

Theories prescribe strategies for planning the instructional and learning process according to supposed situations. In our modeling framework, a learning and instructional strategy is modeled as a WAY in view of generality which can be adapted to the specialized concrete application situations. Such a generic WAY is named “WAY-knowledge”. Currently, we have organized about 100 pieces of “WAY-knowledge” based on some theories: Gagne’s nine events of instruction [5], Dick and Carey’s ID theory [3], Merrill’s Component display theory [13], Keller’s ARCS model [11], Collins’s cognitive apprenticeship [2], and Jonassen’s Design of constructivist learning environments [10]. These are defined as relational concepts (see 3.4).

Organizing “WAY-knowledge” is expected to contribute to the clarification of the conceptual structure of each theory and to theory-eclectic design guidelines for the

<table>
<thead>
<tr>
<th>Categories</th>
<th>Properties</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learner characteristics</td>
<td>Age (type) / Language / Prior knowledge: fact, concept, rule</td>
<td>child, adult / Japanese, English, French / learned, not learned</td>
</tr>
<tr>
<td>Domain/topic characteristics</td>
<td>Concreteness / Complexity / Causality / Prerequisiteness</td>
<td>concrete, abstract / simple, complex / causal, not causal / prerequisite, not prerequisite</td>
</tr>
<tr>
<td>Context characteristics</td>
<td>Context of learning / Testing / Instruction mode / Delivery mode</td>
<td>School, workplace, university / summative, formative, Assessment, certification / individual, group, community / classroom, distance, distributed</td>
</tr>
<tr>
<td>I_L event characteristics</td>
<td>Event kind / Authenticity / Interaction kind</td>
<td>I_L, assessment / authentic, artificial, virtual / action, interaction, social interaction</td>
</tr>
<tr>
<td>Learning object characteristics</td>
<td>Language / Language level / Representation mode</td>
<td>Japanese, English, French / child, adult / text, graphics, image, video, simulation, game</td>
</tr>
</tbody>
</table>
modeling learning and instructional processes. Through this process, we have tried to classify the properties of the theories. Table 1 shows the current classification of the properties. These properties are common to learning/instructional scenarios and models, the pieces of “WAY-knowledge” and concepts of theories in the OMNIBUS ontology. One of the contributions of this is to enable authoring tools to understand the relation between theories and a scenario and to explain it to the authors. Using these properties, such an authoring tool can suggest theories that have the same properties as a scenario to the authors, or can provide the accordance of properties between a scenario and a theory as the justification of a scenario for the authors. We have implemented authoring supports of this kind. This is discussed in 4.4.

3. Concepts defined in the ontology

Among various types of concepts defined in the ontology, we are here concerned only with the main concepts related to the instructional/learning scenario models introduced above. This section describes what are defined in this ontology and how to read it using Hozo ontology editor.

3.1 The basic principle of conceptualization in the Hozo ontology editor

Although the ontology is presented in two versions - Hozo and OWL -, we recommend to read the Hozo version, since it represents the full semantics of the ontology. However, you need to know the basics of the Hozo way of representation.

The Hozo ontology editor handles the following two concepts separately:

- Wholeness concept: A concept of a thing considered as a whole (e.g. bike), which is composed of multiple concepts (e.g. wheel, handle, etc.), each of which makes up a part of the whole,
- Relational concept: Conceptualized relationship between multiple (usually two) concepts.

This distinction is done based on the following consideration [12]. For example, let us consider a “brothers” and a “brotherhood”. Assume that there are two brothers, Bob and Tom. “The Smith brothers” could be a conceptualization as a whole, which is a pair of two persons. On the other hand, “brotherhood between Bob and Tom” is conceptualized as a relation. On the basis of the observations that most of the things are composed of parts and that those parts are connected by a specific relation to form the whole, wholeness concept and relational concept are distinguished in Hozo ontology editor. In this example, the “brothers” can be considered as a wholeness concept and the “brotherhood” as a relational concept. Theoretically, every thing that is a composite of parts can be conceptualized in both perspectives as a wholeness concept and a relational concept.

In the ontology, two types of relations are defined: pure relation such as “same as” and “before-after”, and “WAY-knowledge” (strategy) such as “Educational strategy”. While the former is used as constraints on the wholeness concepts(normal classes), the latter is not used that purpose but used for representing “WAY-knowledge”. Details are explained in relational concept.

3.2 Ontological approach to the systematization of educational theories

3.2.1 Fundamental viewpoint

The relation among theories behind instructional design is considered as a nested structure as shown in Figure 3. The bottom of the structure is the “learning world”. Learning theories
explain processes and events in the world. The "Instructional world" is on top of the learning world. The instructional process influences or facilitates the learning process. Instructional theories prescribe the effective instructional process for the learning process with the desired outcome. The instructional process happens in parallel with the learning process. Moreover, the "Instructional design world" is on top of the instructional world. Instructional design process is the design process of the instructional process. An instructional design theory prescribes the rational process for designing the instructional process. One of the major differences among the three kinds of processes is that while the lower two are real world processes, the other is a planning or design process of real world processes/events. However, thinking along the nested structure, we see an essential characteristic that all the processes rely on the learning process which can be modeled as a state-change in a learner. Therefore, based on our working hypothesis stated in 2.1, we built this ontology with the state-change in a learner as the foundation of the conceptual system.

3.2.2 Upper level structure

The upper level structure of this ontology is shown in Figure 4. Roughly speaking, the OMNIBUS ontology is mainly composed of concepts related to the "Common", "Cognition", "Learning", "Instructional" and "Instructional design/Instructonal system design (ID-ISD)" worlds, and "Event". We would like to emphasize that our policy of conceptual distinction between "Event" and the other process-related concepts is based on context-dependence. Concepts related to each world are defined as those necessary to represent processes in the respective worlds with minimal context-dependence. On the other hand, "Event" and its subclasses are defined as those for representing (1) events with maximal context-dependence on education referring to those defined in other worlds and (2) relations between them. We discuss the distinction and the context-dependence of them in a bit more detail in 3.3.3.

This paper discusses OMNIBUS ontology with a focus only on the learning and instructional worlds and the relations between them. In the following sections, we mainly explain "State" and "Action" that are defined in the common world and that are shared among the learning and instructional worlds in order to describe learning and instructional process, "Educational event" as a contextualized description of process, and "WAY" as the relational concept.
3.3 Main wholeness concepts

This sub-section introduces the main wholeness concepts in this ontology.

3.3.1 State

As discussed in 2.1, states in the learning process are the most important factor for building the comprehensive ontology. Each description of educational theories uses its own terminology. Hence each theory has different concepts of states from the other theories at a glance. However, Reigeluth points out that many theories are described in different terminology although some of them describe or prescribe the same method for the same situation (states as the precondition and the outcome) [18].

In accordance with our working hypothesis stated in 2.1, we have collected such states from several theories and categorized them under an is-a hierarchy from the view point of the conceptual meaning (Figure 5). States in the OMNIBUS ontology are mainly classified into the following two types of classes.

- Internal state: This is a state about the inside of agents. This includes “Cognitive process state”, “Attitudinal state”, “Progression state”, “Developmental state”.
- External state: This is a state locating between internal state and situation and is an aspect of the agent’s engagement/participation in an action.

In this ontology, these states are common to any theory and learning is described by changes of learner state. Therefore the difference between theories is described as the difference of states used or not used in the theories, process of changes of the state supported in the theory, and the relation between changes of states and learning actions.

In the following, we explain the definition of actions and discuss the relation between changes of states and learning actions.

3.3.2 Action

Actions are defined in common with learning and instruction (Figure 6). All actions are decomposed into some subactions and the decomposition can be repeated almost endlessly. However, in most cases, the decomposition should be stopped at a certain granularity level under which...
finer grain actions are meaningless in the context. The finest grain actions are called primitive actions.

There are two kinds of actions: one which has a unique decomposition into subactions like “walk” and the other which has multiple ways of decomposition according to the context/goal under which actions are being performed like “teach”. The former case is a normal case so that subactions are defined using (part-of) slots. The latter case needs special care to grasp its rich meaning properly, that is, to uncover its deep implications underlying the ways of performing the sequence of subactions. This is why we don’t use part-of slots but instead introduce “WAY-knowledge” (way of decomposition) together with “Event” for the latter kind of actions. Needless to say, the distinction between these two kinds of actions is relative. Furthermore, we introduce the idea of “Event” to capture the latter kind of actions. By event, we mean a large chunk of actions full of contextual stuff such as “learning event” and “instructional event”. “Event”s refer to “Action”s at particular situations which require particular actions to achieve their goals.

3.3.2.1 Primitive action
This action changes the “Communicative state” of the doer or of the object. In this ontology, actions of this type cannot be decomposed into some sub actions. That is to say, they are primitive actions.

3.3.2.2 Physical state action
This action also changes Communicative state of the doer or the target object. In contrast to primitive actions, this action can be decomposed into some sub actions to achieve the state change intended in the action. However, the decomposition is not defined in the definition of the action. It is defined as a “WAY” that will be explained in the later section: WAY: prescriptive model derived from strategies defined in theories.

Example 1: Inform (Figure 7)
This class defines the action “Inform” as the state change of the recipient of the “Inform” action, which is an Agent, to the state of “Informed” which is a sub-class of communicative state.

Example 2: Remind (Figure 8)
This class defines “Remind” as the state change of recipient of the action to the state of “Led”, which is a sub-class of communicative state, to “Recall” action. Note that it is out of the scope whether the action led is actually done or not. This definition intends to describe the intention of the doer of the action. The action of “Recall” is defined as a sub-class of the “Cognitive action” which is explained below.

3.3.2.3 Cognitive (state) action
This action changes the internal state of the doer or the object. This action can be also decomposed into some sub actions to achieve the state change intended in the action.

Example 1: Recall (Figure 9)
This defines that the action of “Recall” changes the state of doer of the action to the state of “Have recalled”, which is a sub-class of the internal state. This is illustrated below.
3.3.3 Educational event

“Educational event” is the concept for representing (1) events of learning and instruction, and (2) relations between them. Its is-a hierarchy is shown in Figure 10. “Learning event” and “Instructional event” are the concepts for representing events of learning and instruction. “I_L event” is a concept for relating “instructional event” to “learning event”.

In this ontology, the instructional and learning process is defined as a process with a goal related to the situation. A process can have different meanings in different contexts, which are composed of goal, situation, etc. For example, informing a topic has an intention to afford better understanding of a learner in a context but has an intention to just call a learner’s attention in another context. Based on such considerations, this ontology focuses on describing the instructional and learning process with the context clearly. The concept to describe such a process is “Event”. Basically, “Event” is defined as composition of “process” and the contextual information of it such as “participant”, “time”, and “location”. In “Educational Event”, “process” is specified by “Action”. You may wonder why we define “Action” and “Event” separately. The definition of “Action” also has “participant” as explained in the previous section and “time” and “location” will be fixed when an instance of “Action” is made. But what we can say here is that our policy of conceptual distinction between “Action” and “Event” in this ontology is context-dependence as mentioned in 3.2.2, especially the relation to the goal of “Action”: “Event” is defined to be dependent on a context. On the other hand, “Action” is defined independently of the context, and just defined as the change of states with no relation to any intention.

3.3.3.1 Learning event

A “learning event” is composed of an agent as a learner, a learning action, its objects, effects and conditions of learning, and spatial/temporal attributes. The relation among them is suggested by learning theories.

3.3.3.1.1 Effect of learning

By “Effect of learning”, we mean that a learning theory can tell us what effect is expected after this learning action. This meaning is described by the “Action result” relation among “Learning action”, “Learning effect” and “Learning theory” slots as shown in Figure 11.

![Figure 10: Is-a hierarchy of “Educational event”](image)

![Figure 11: Definition of “Effect of learning” in Hozo](image)

![Figure 12: Definition of “Preparing learning condition” in Hozo](image)
3.3.3.1.2 Preparing learning condition
By “Preparing learning condition”, we mean that when learning conditions are satisfied, the learning theory assures that the learning action should be successful. This meaning is described by the “Guarantee” relation among “Learning action”, “Learning condition” and “Learning theory” slots as shown in Figure 12.

3.3.3.2 Instructional event
An “Instructional event” is composed of an agent as an instructor, an instructional action, its objects and spatial/temporal attributes. The definition of this event itself is defined independently of the definition of “Learning event”; therefore an “instructional event” does not include how the instruction affects learning. The effect of instruction on learning is described within “I_L event” as presented below.

3.3.3.3 I_L event
An “I_L event” defines the relation between a “Learning event” and an “Instructional event”. That is to say, an “I_L event” describes how an “Instructional event” contributes to a “Learning event”. This relation is defined from two points of view. The first is the contribution of “Instructional event” to the change of a learner’s state. The other is the preparation for the following “Learning event”.

3.3.3.3.1 Example 1: Preparing learning condition and Remind event
Figure 13 shows the definition of “Preparing learning condition”, which is a sub-class of “I_L event”, and Figure 14 shows “Remind event”, which is a sub-class of “Preparing learning condition”.

“Preparing learning condition” is composed of one “I event” slot and two “L event” slots (one is constrained by “Effect of learning” and the other is constrained by “Preparing learning condition”). This defines the following two kinds of relation between learning and instruction:

1. An “instructional action” influences a “learning action” that causes an expected “learning effect” (Of course, this never means that the learner always does the intended learning action and changes to the intended state).
This is defined as the “influence” relation between the “instructional action” slot in the “I event” slot and the “learning action” slot in the “L event” slot constrained by “Effect of learning” (“Learning event”).

2. “learning action” is a preparation of the other “learning action”. That is, “learning effect” satisfies the condition of the other learning action (“learning condition”). This is defined as the relation between the “learning action” slot in the “L event” slot constrained by “Effect of learning” slot and the “L event” slot constrained by “Preparing learning condition”.

These are defined as three slots (one “I event” and two “L event”) and by the relation between them (“Influence” and “Prepare-cond”).

In “Remind event”, these are defined more concretely as shown in Figure 14:

1. A “Remind” action of an “instructor” influences a “Recall” action of a “learner”, which causes a change of learner state to the “Have recalled” state. This relation is defined as the “Influence” relation. (In the Figure 14, the “Influence” relation in the “Remind” event appears just in order to make the explanation easy to understand. It is not necessary to define this link in the “Remind event”. “Remind event” is a sub-class of “Preparing learning condition” so that the “Influence” relation is inherited.) You may think that the “learning effect” slot of “L event” slot and the “terminal state” slot of “doer” slot in “learning action” are redundant descriptions. However the “learning effect” slot of the “L event” slot describes only a notable state in this event picked from states defined in the learning action.

2. The “Have recalled” state as the “effect of the learning” is a preparation of the other future learning action of the learner. That is, the effect satisfies the condition of a future learning action. This relation is defined as the “Prepare-cond” relation and “same as” relation.

3.4 Relational concepts

In this ontology, we define the following two types of relational concepts in HOZO:

- Pure relational concepts: This is a concept to define a relation among slots in a wholeness concept. In this ontology, relational concepts other than sub-classes of “WAY” are pure relational concepts.
- “WAY” and “WAY-knowledge”: “WAY” is a special relational concept to describe a way of achievement of the state change in a learner. This type of relation is not used for defining the relation among slots in a wholeness concept. “WAY-knowledge” is a specified concept of “WAY” based on theories as of now. In the future, we are planning to extend this definition not only to theory but also to empirical knowledge.

In OWL, both of them are defined as a sub-class of “RelationalConcept”, which is a sub-class of owl:Thing. Note that they are not defined as sub-classes of owl:Property.

3.4.1 Pure relational concept

In this ontology, some pure relational concepts are defined. For example, “less-than”, “same as”, “Influence”, “prepare-cond”, etc. Please see in detail in the Hozo or OWL-compliant ontology editor. These pure relational concepts are used in order to define the relation among slots of a wholeness concept as shown below.
3.4.2 WAY: prescriptive model derived from strategies defined in theories

As mentioned above, “WAY” is a relation between an “upper (macro) I_L event” and one or several “lower (micro) I_L event”s that achieve the upper one. That is to say, a “WAY” is a description of an educational strategy. Theories prescribe strategies for planning the instructional and learning process according to supposed situations. In our modeling framework, a learning and instructional strategy is modeled as a “WAY” in view of generality which can be adapted to the specialized concrete application situations. Such a generic “WAY” is named “WAY-knowledge”. Figure 15 illustrates examples of the description of “WAY-knowledge” named “Presentation” WAY. This is based on a part of Gagne and Briggs’s nine events of instruction. The key of this strategy is to tell directly to a learner the content and way of learning. This “WAY-knowledge” describes this learning and instructional process as the sequence of two “micro I_L event”. One is to inform the content of learning as a learning item in order to let a learner recognize it and the other is to inform guidelines for learning to the learner in order to let a learner recognize the guidelines. These processes are described by “Guiding event”, which is a sub-class of “I_L event”. “Guiding event” focuses on the continuous interaction between an instructor and a learner. In this “I_L event”, the interaction is taking place from the start to the end. On the other hand, there are the other types of "I_L event", for example, “Enhancing”. This focuses on a non-continuous interaction. In this type of “I_L event”, basically the interaction between an instructor and a learner is taking place only once at the beginning. Then the learner does by him/herself.

4. A theory-aware and standards-compliant authoring system: SMARTIES

This section discusses the application of the ontology of learning and instruction and the framework of function decomposition tree.

Existing authoring environments for learning support systems aim at combining authoring tools and knowledge representation [15]. Most of the systems have functionalities to support instructional and learning design based on some sort of fixed theories (or empirical knowledge). Of course, such systems provide designers with guidelines and
improve the consistency of design on this basis. However, all of knowledge from the theories in many of the theory-based systems is built in the procedures. It is the developer, not the system, who knows the theory. It causes concealment of the relation between the system’s functionalities and the theories they are based on.

Our study aims to build a theory-aware design support system that understands theories. Such a system has the capability of explaining which theory underlies any suggestion the system makes to authors, as opposed to a system in which the theories are implemented as built-in procedures. The following sub-sections present our idea of a design support system called “SMARTIES: SMART Instructional Engineering System”, which we have been developing.

4.1 An overview of a theory-aware design support system

Figure 16 shows a block diagram of SMARTIES, which has been under development in this study. The scope of support is limited to the design phase of ID process, rather than the analysis and development phases.

SMARTIES helps two types of users; one is scenario authors, which includes instructional designers, educational practitioners and occasionally learners. The other is knowledge author, which mainly includes researchers and theorists.

A scenario author makes a particular instructional and learning process model using the authoring interface. The model manager manages a model that scenario authors made. In addition, the model manager provides the author with guidelines for making a model. Based on the ontology, basic guidelines for modeling instructional and learning process are supplied; concepts and a vocabulary representing them, and the basic structure of concepts. In addition, based on “WAY-knowledge”, instructional and learning strategies from theories are supplied. Finally a scenario is generated as the leaves of an instructional and learning process model. The model can also be finally exported according to IMS LD specification [9].

A knowledge author describes instructional and learning strategies as “WAY-knowledge” with an understanding of theories and put them to the Way-knowledge base. The Way-knowledge manager manages Way-knowledge base and provides knowledge authors with the ontology as basic guidelines as well as the model manager. Describing “WAY-knowledge” makes it possible for scenario authors to retrieve strategies for inter-theory cooperation and apply multiple theories to a particular instructional and learning process model.
4.2 Scenario description support

Figure 17 shows a screen shot of SMARTIES. This scene shows how a scenario author makes an instructional and learning scenario using “WAY-knowledge”.

The scenario editor provides a scenario author with an environment to describe an I_L event decomposition tree as an instructional and learning process model. I_L events are represented as nodes and the decomposition of them is represented from top to bottom. In this window, an author decomposes the learning goals of the scenario step-by-step by choosing applicable “WAY-knowledge”.

The Way-knowledge window provides an author with applicable “WAY-knowledge” candidates in order to help him/her decompose each I_L event. It displays applicable pieces of “WAY-knowledge” appropriate to the selected I_L event that he/she wants to decompose. When the author chooses one of them, a proposed decomposition is displayed on the window. If the author decides to adopt the selected Way, the proposal is applied to the main window. By repetition of the process mentioned above, a scenario author makes instructional and learning process model, moving from abstract levels to concrete ones.

4.3 IMS LD export

In order to enhance sharability and reusability of the scenario descriptions, we have mapped I_L event decomposition tree onto IMS LD specifications. Briefly speaking, each unit of

![Diagram of I_L event decomposition tree and IMS LD export process]

Figure 18  Mapping an I_L event decomposition tree into IMS LD
decomposition in an I_L event decomposition tree can be converted to two activity-structures for learner and instructor in an IMS LD description as shown in Figure 18.

In IMS LD, only top and leaf activities have the description of the objective while the others do not have. Therefore only a part of the design intention can be converted to the IMS LD description although it keeps sharability and executability of learning/instructional scenarios. On the other hand, an I_L event decomposition tree keeps the whole design intention together with theoretical justification of it. For these reasons, IMS LD and our modeling approach are complementary to each other.

4.4 Scenario explanation support

One of the characteristics of theory-aware systems is the ability to interpret learning/instructional scenarios in terms of theories. An I_L event as a descriptive concept and “WAY-knowledge” as a prescriptive concept enable information systems to explain theories and scenarios described as an I_L event decomposition tree and to give suggestions for scenario design/improvement.

4.4.1 A classification of explanation types and cases

In this study, we have classified explanation of a scenario into two types. Table 2 summarizes the classification. Each type covers some cases of explanation about interpretations of learning/instructional processes or problems in achievement of the goals. These types of explanation are based on the properties of theories discussed in 2.4. I_L event decomposition tree, “WAY-knowledge” and the definition of theories in the OMNIBUS ontology are characterized by the properties. Such scenario explanation can be done by comparison between their properties.

Interpretative explanations report interpretation result of a scenario based on the ontology and “WAY-knowledge”. Scenario comprehension uses the scenario model and the descriptive concepts in the ontology. Even if an author does not use any pieces of “WAY-knowledge” for scenario authoring, this interpretation can be done. Theory exposition uses only “WAY-knowledge”. This just tells what each theory proposes independently of a particular scenario. A theoretical justification of scenarios is a combination of them. This explains both interpretation of a particular scenario and its

<table>
<thead>
<tr>
<th>Type</th>
<th>Case</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interpretive explanation</td>
<td>Scenario comprehension</td>
<td>Explaining just interpretation of relation among events in a scenario without theoretical justification. E.g. An event is preparation of another event.</td>
</tr>
<tr>
<td></td>
<td>Theory exposition</td>
<td>Explaining theory itself independently of a specific situation.</td>
</tr>
<tr>
<td></td>
<td>Theoretical justification</td>
<td>Explaining interpretation of relation among events in a scenario with theoretical justification. E.g. An event is preparation of another event and the necessity is guaranteed by a theory.</td>
</tr>
<tr>
<td>Suggestive explanation</td>
<td>Insufficiency of necessary goals</td>
<td>It seems learners can’t achieve the goal because necessary (sub) goal is insufficient in the scenario.</td>
</tr>
<tr>
<td></td>
<td>Insufficiency of supplementary goals</td>
<td>It seems learners can achieve the goal but not so effective. If some supplementary goals are added, it will be better.</td>
</tr>
<tr>
<td></td>
<td>Excess of goals</td>
<td>It seems that it is difficult for learners to achieve the goal because there are too much unnecessary goals in the scenario.</td>
</tr>
<tr>
<td></td>
<td>Disproportion in process</td>
<td>The scenario doesn’t have proper proportion of process. E.g. lack of motivating, too much assessment, etc.</td>
</tr>
<tr>
<td></td>
<td>Inconsistency of principle</td>
<td>The principle of learning and instruction isn’t stable. A scenario needs some extent of sustainment of principle. E.g. too many suggestions in inquiry learning.</td>
</tr>
<tr>
<td></td>
<td>Unsustained state</td>
<td>A state doesn’t sustain until when it is required. E.g. an event reminds a learner of prerequisite knowledge but another event that needs it is far ahead.</td>
</tr>
</tbody>
</table>
justification based on pieces of “WAY-knowledge”. These kinds of explanations are expected to be useful for authors to review their own scenario or to know the design intention of those made by others.

Suggestive explanations generate suggestions for improvement of scenarios. This is used when scenario authors did not use applicable “WAY-knowledge” suggested by SMARTIES but decomposed the tree using their own way knowledge in terms of the system vocabulary. In such a case, scenarios would lack theoretical validity but the system can infer the authors’ intention to some extent. This type of explanation is based on the interpretation of a scenario and includes improvement suggestions of a scenario. Cases that suggestive explanations cover are listed in Table 2. Note that it is intended not to force authors to follow but to recommend alternative ways or different viewpoints. These explanations are expected to be useful to check the validity of authors own scenarios.

4.4.2 Generation mechanism of scenario explanation

In order to generate scenario explanation we made message templates whose vocabulary comes from the ontology and whose structure is partly based on an I.L event decomposition tree. Comparing scenario models with the ontology and pieces of “WAY-knowledge” enables a theory-aware system to make interpretation and to generate explanation messages.

Figure 19 illustrates an example of explanation generation. Figure 19 (d) shows an example of explanation message about Insufficiency of necessary goals of Suggestive explanation. This message is generated from the message template (Figure 19 (a)). Italic words in the template are specified by the scenario model (Figure 19 (b)) and a definition of a piece of “WAY-knowledge” (Figure 19 (c)) in the ontology. This message template is composed of two parts, which are a scenario interpretation part and a theoretical justification part. The former is related to a scenario model and the latter related to “WAY-knowledge”. Each part has some blank entries to be filled and each blank is related to a
required model structure or pieces of “WAY-knowledge”. This relationship enables a theory-aware system to generate specific messages using templates according to a scenario.

When a system generates a message, firstly, a system compares descriptions of learner’s state-changes in a scenario model with ones of “macro I_L event” in pieces of “WAY-knowledge”. If a state-change is the same in both but “micro I_L events” are different, the system checks the difference and generates explanation messages to notify the estimated problems. Note that this comparison cannot work if a user describes state-changes by his/her own words instead of the system vocabulary because this function is based on the concepts defined in the ontology.

The first half of the message explains the current state of the scenario model. In this case, it points out that it seems learners cannot achieve the entire goal of the scenario because one of the sub-goals is unlikely to be achieved. The blank entries in the template are filled using the scenario model (Figure 19 (b)). In this case the necessary information is <The goal of the whole scenario> and <The goal not to be achieved>. These goals are detected from the model, and words representing the goal are put into the template.

The last half explains the reason of the problem and an improvement suggestion based on a theory. The template is embodied as a message using the definition of a piece of “WAY-knowledge” (Figure 19 (c)). In this case, this message is based on Gagne and Briggs’s theory. The insufficient goal in the scenario, which is “Motivated”, is identified with the piece of “WAY-knowledge” and fills the blank of the template.

We are currently developing the explanation function in our prototype. Figure 17 (F) is an example of generated explanation messages.

5. Concluding remarks

We have discussed the OMNIBUS ontology together with its application to building a theory-aware authoring tool, SMARTIES. Especially, the stress has been put on the details of design rationale and how the ontology is used at the computational level. We are just in the middle point of our long-term project. Because we already summarized what are left undone in 1.3, we only confirm here that there remain a number of problems to solve. The authors would like to kindly ask your warm help and feedback.

Ontology engineering is the key technology whose roles should be apparent. While the problem attacked in OMNIBUS project looks very AI-oriented, it is essentially different from the traditional AI in which people try to build problem solvers for humans. In other words, people try to make performance systems such as ITSs intelligent. Ontological engineering is different. It helps people solve problems by providing useful, long-lasting and reusable fundamental concepts and knowledge. It tries to make authoring tools (meta systems) intelligent rather than performance systems (ITSs). Hidden goals of OMNIBUS project include showing this new direction of AI technology application in AIED community.

References

A Grid Based IMS Learning Design Player: 
the ELeGI Case Study

Nicola CAPUANO a, b, 1, Roberto IANNONE a, Sergio MIRANDA b, c 
and Marcello ROSCIANO a 

CRMPA, Centro di Ricerca in Matematica Pura ed Applicata, Italy 
DIIMA, University of Salerno, Italy 
MOMA, Modelli Matematici e Applicazioni, Italy

Abstract. This paper illustrates the work done and the results achieved within the 
ELeGI project about the orchestration and the delivery of Learning Services lying 
in the GRID inside an IMS Learning Design (IMS-LD) Unit of Learning and 
running under an enhanced version of the CopperCore Player. The added value of 
GRID technologies for the creation and the execution of dynamic learning 
experiences is evidenced as well as the experimentation performed to overcome 
the original IMS-LD limitation on running services is presented. The aim of the 
ELeGI project is to promote and support a learning paradigm centred on the 
knowledge construction using experiential based and collaborative learning 
approaches in a contextualised, personalised and ubiquitous way through the 
definition and implementation of a service oriented Grid based software 
arquitecture.

Introduction

The aim of the ELeGI project is to promote and support a learning paradigm centred on 
knowledge construction using experiential based and collaborative learning approaches 
in a contextualised, personalised and ubiquitous way through the definition and 
implementation of a service oriented Grid based software architecture.

In the context of the ELeGI project a new approach to the learning design web 
 service integration has been studied. It exploits the CopperCore architecture (engine 
and player) in a distributed learning environment. This paper describes the realized 
framework able to generalise the web service integration inside an IMS LD Player.

In Section 1 the ELeGI project and its goals are briefly described. Section 2 
introduces the IMS Learning Design specification and the actual engine and player 
implementation. CopperCore engine is described as well as the SLeD project, and 
the actual limitations related to the web service integration issues. Eventually, Section 3 
details the realised approach.

1 Corresponding Author: CRMPA, Centro di Ricerca in Matematica Pura ed Applicata, Università di 
Salerno, Via Ponte Don Melillo, 84084 Fisciano (SA), Italy; E-mail: ncapuano@unisa.it.
1. The ELeGI project

1.1. Objectives

The European Learning Grid Infrastructure (ELeGI) project [1], a EU-funded Integrated Project, aims at promoting and supporting the adoption of a learning paradigm focused on the construction and sharing of knowledge in high dynamic contexts using experiential based and collaborative learning approaches in a ubiquitous, contextualised, and personalised way and taking into account informal learning aspects as well. To achieve these goals, ELeGI defines a clear strategy, formalized through the definition of models (supporting formal and informal learning scenarios), methodologies and technologies, enabling to overcome the drawbacks of traditional e-Learning solutions and to advance the effective use of Technology Enhanced Learning (TEL).

1.2. The ELeGI Learning Model

The Learning Model defined in the ELeGI project and fully described in [2] is conceived in such a way to represent different pedagogical approaches and to allow the automatic building and delivery of adaptive Unit of Learning (UoL) that can dynamically change during the learning process according to the learner’s pedagogical needs and preferences. The ELeGI Learning Model produces an operational process so to create and deliver a UoL by using the structures defined by the three underlying specific models: the Knowledge Model, the Learner Model and the Didactic Model. The overall operational process can be divided into three processes, each one addressing a specific phase: Knowledge building process, UoL building process and UoL delivery process.

Unlike the traditional approaches to the creation of UoLs and/or learning courses, the exploitation of the learning model offers several advantages. Actually, examining current tools (IMS-LD tools, LAMS, etc.), a UoL can be considered as a static and monolithic block, since once created, it is rather difficult to change or modify its inner resources and/or to add/remove services and resources at run-time. Moreover, the traditional approaches to create a UoL typically rely on expertises of the institutional designer only, and have no capability to reuse existing blocks. Conversely, exploiting the Grid technology and the virtualisation mechanism (by which each resource is virtualised as a service), the ELeGI approach allows both to add dynamicity during the UoL delivery, enabling the automatic search and late binding of resources and services, and to reuse already developed building blocks (e.g. Ontologies, LEM, Didactic Methods) during the UoL building. Thus, the benefits of the ELeGI approach, from the reusability and pedagogical viewpoints, appear to be noticeable.

2. A Learning Design implementation

2.1. IMS Learning Design

The IMS Learning Design is a specification used to describe learning design scenarios. It can describe a wide variety of pedagogical models, or approaches to learning,
including group work and collaborative learning. It does not define individual pedagogical models, it provides instead a high level language, or meta-model, useful to describe many different models. The language outlines how people perform activities using resources (including contents and services), and how these three components are coordinated into a learning flow.

The IMS Learning Design illustrates how a learning design scenario unfolds making an analogy with a theatrical play. Just as a play can be staged by different actors, in different theatres/places and changing props, so learning design scenarios can be executed again by different learners and tutors, on different systems, with alternative learning resources or tools:

- The play is presented in a series of acts, in which roles are played by those actors taking part in it, for instance: learner, tutor, mentor, and so on.
- People playing the roles undertake a series of activities within an act. For a learner these might include discussing with classmates about the merit of a piece of source material. A tutor’s activity could be to remark on their conclusions.
- Each role is presented together with its own learning objects (LO) and services (e.g. communication tools) within an activity.
- An act is complete after all the activities of a specified role, or roles, are finished. Alternatively, a time limit may be set, which determines an act to be completed.
- As soon as an act is completed, the next one starts. The play finishes when all the acts are complete, the learning design scenario finishes when all the plays are complete.

The IMS Learning Design specifications are structured into three levels:

- **Level A** includes activities, roles and environments. Activities (learning activities or support activities) can be grouped into activity structures and executed into specific environments. An environment is composed of learning objects and services provided to the users during the activity execution. Users are classified into roles (learners, teachers, tutors, etc…). Learning objects are educational contents by which learners acquire knowledge and services are functionalities invoked during the learning process in order to communicate with tutors or other learners.
- **Level B** adds properties (storing information about a single person or a group) and conditions (setting constraints upon the flow of activities) to the first level.
- **Level C** adds notifications (such a mechanism handling messages between users) to the framework.

As observed in [3], even though IMS-LD presents a mechanism to personalize the learning experience at run-time using properties and conditions at Level B, the instructional designer has to provide a fixed set of learning objects and services in which to select contents and tools for final users of the learning experience. So learning objects and services are statically bound to the learning design scenario.
2.2. CopperCore

CopperCore is the world’s first open source Learning Design engine capable to process all three levels of IMS Learning Design. Released from Open Universiteit Nederland (OUNL) it is not designed be used as a stand-alone learning environment but to be integrated in a service oriented framework consisting of different services that are combined to create a complete e-learning system. CopperCore architecture provides three application program interfaces (APIs). The first provides access to validation; the second to services; and the third provides the presentation aspects.

All the APIs are exposed using XML and can be called or manipulated externally to provide file management and student management. This allows the separation of content and services from the engine itself. The CopperCore system also provides a LD player implemented as XSL style sheets applied to the transformed learning design. The core implementation uses Java to provide the combining and integration code.

2.3. Service delivery inside the SLeD project

The SLeD project (Service Based Learning Design Player) [5] was the result of a collaboration between the UK Open University (UKOU) and the OUNL and was funded as part of the Framework and Tools strand of the JISC e-Learning Programme, which is constructed around the concept of a service oriented architecture [6]. The project was built on the CopperCore engine which validates Learning Design packages to check if they conform to the specification, and if not, indicates the problem areas.

The objectives of the project were focuses on the extension of the available tools to the Learning Design community and were also intended to further develop the way a Learning Design approach might be practically realized within an institution. One of the aim of the project was to extend and formalise the approach for integrating services while maintaining the architectural integrity of the system. The Sled architecture is shown in the figure 1.

The CCSI module is a layer that sits between the Sled (or other learning design) player and the other LD services (for example the CopperCore engine, forums, search etc). It provides a single point of contact thank to which the player may access all the services, and also coordinates the communication between the services. For example, with the QTI service, the CopperCore LD engine also needs to know the results from the users’ test, so when the user submits his/her answers, CCSI will pass the submission received from the user to the QTI engine, and then return the test results back to the user and to the CopperCore engine. Communications between player and services work due to the fact that the player is connected to CCSI by a set of predefined methods (for each service used), the CCSI module then transforms/adapts this method call to that actually expected by the service provider.

CCSI also handles the transformation of the information back from the service provider to the player. In this way the player always knows the method to call and that one it should expect back - no matter who is providing the actual service, it is a task of the CCSI layer to ensure that the transformations take place correctly. For each type of service (e.g. LD engine, forum, search) there is an adapter class which provides the actual implementation and transformations for the player to connect to these services. Any parameters required by the adapters (for example URLs to web services) are contained in a configuration file.
2.4. Actual limitations

Even though the SLeD project has considerably improved the service integration into IMS Learning Design player some drawback already persists.

In order to switch between service providers for which an adapter class is available, e.g. to switch between a Google search provider and a MSN search provider, it is necessary to manually modify an xml configuration file in the CCRT folder of the CopperCore installation and then to restart CopperCore for the changes to take effect. Furthermore it is possible only a single connection to a provider for a particular service at a time. For example it could not have a Unit of Learning which uses a Google search engine and another Unit of Learning configured to use MSN search.

Adding a new service type is more complicated than switching service providers. The steps to carry out are the following:

- To think to the generic functions for the new service
- To code the adapter classes to connect up to the actual service provider.
- To update the player so that it will connect to the new service type.
- To create the interface the authors could use to write unit of learning that embeds the new service.

3. Service delivery inside ELeGI IMS-LD Player

The ELeGI Software Architecture is presented in Figure 2. The Grid layer provides a set of Infrastructure Services and other services to create and manage a Virtual Organisation (VO) [8]. The Learning layer is mainly devoted to the execution of the processes related to the Learning Model.

The Learning Services sub-layer, provides services and tools to support the execution of the three processes of the Learning Model including the management of
Ontologies, the Learner Model, and the Didactic Model. The Personalization subsystem aims at dynamically adapting and delivering educational contents and services, matching learner’s needs and preferences according to his/her profile. The Learning Experience Management subsystem allows to access and manage courses, modules, and other learning experiences (e.g. allocating student, staff, etc.), while Contents & Services Orchestration subsystem deals with the execution of the UoL, that are described using the IMS-LD [4].

The Application Layer uses the services provided by the underlying layers, or their composition, to implement applications in the e-learning domain. This layer includes a key component of the ELeGI software architecture, IWT Grid Aware (IWT-GA) [9] Portal that, according to the research on Grid portals, is designed exploiting the Web Services for Remote Portlets (WSRP) specifications [10].

IWT-GA is the Grid version of Intelligent Web Teacher (IWT) learning platform [11], it adopts the concept of portlets as a way to design user-centric portals that can be dynamically adapted to the context.

IWT-GA holds an IMS-LD compliant engine and Player, a modified version of Coppercore [12] that exploits the Contents&Services Orchestration subsystem services of ELeGI, that is the subject of the following discussion.

3.1. Service implementation

To have a service running inside the EleGI LD Player is quite simple, indeed it only needs to follow a few implementation specifications for the service development and its deployment in the GRID. There is a loosely-coupled connection between the service and the UoL or the Player that will run it, indeed no specific reference to the service, or particular installation inside the player must be done, the only requirement is to specify inside the IMS-LD learning activity the metadata of the required service.

In order to implement a new service or to extend an existing one, and made it compliant with and fully executable within the IMS-LD Player here referenced to, some APIs have been developed and made available.

The services must be a WSRF compliant GRID service (the WSRF.Net framework which the University of Virginia Grid Computing Group has adopted) [16] in order to maintain its state during the various phases of the execution.

By simply extending a class provided by the mentioned APIs, the service inherits the state management logics and the methods useful to communicate with the LD Player. A typical communication between the two parts can be summarized by two classes of methods. In short, the Player uses a Setter method to pass to the service its initialisation parameters and two methods to know the references of the service delivery portlet to render. The service instead can use the Getter method to read the parameters passed. All the methods can be overridden in the service if the default implementation behavior is unsuitable, otherwise the simple class extension will be sufficient.

It is important to assign in the code the right WSDL base name to the GRID service, since the LD Player in order to communicate with every service needs to use a general service proxy, therefore the methods to be reachable on the service must have a fixed namespace.
3.2. Services discovery and binding

The web service integration approach described in this paper is slightly different from that adopted in SLeD architecture. The idea is to use a special XML configuration file that follows the XML schema depicted in the Figure 3.

The SLA element follows a schema that is an extension of the Web Service Level Agreements (WSLA) standard [13]. WSLA specifications are agreements between a service provider and a customer defining the obligations/duties of the parties involved in it. Primarily, this is the obligation of a service provider to perform a service according to agreed-upon guarantees for IT-level service parameters (such as availability, response time and throughput) for Web Services, also specifying the measures to be taken in case of deviation and failure and to meet the asserted service guarantees (for example, a notification of the service customer).

This file represents a description of the service which includes the parameters it needs, specified by “param” element in the schema, and the discovery method used to retrieve the service. There are two implemented retrieving methods:

- Static address specification. Simply the service address specified in the “address” attribute of the root element is used to instantiate it;
- Dynamic address specification. This method is adopted if the “address” attribute is not specified. Then the “SLA” element is considered. This element contains all the information needed to retrieve a service registered in a UDDI repository.

The dynamic address specification needs that the desired service is registered using GRASP, a Grid middleware developed in the frame of the homonymous FP5 project [7], and successively re-factored in order to be WSRF compliant by exploiting the WSRF.NET implementation of the University of Virginia Grid Computing Group.

The service description file can be inserted into the desired UoL as a learning object within the learning activity where the service should be used. The ELeGI LD Player (a modified version of the CopperCore Player) is used to parse the description.
file. When in a UoL play the user reaches a learning activity that entails a service support, the LD Player parses the file to search for retrieving the method to exploit.

If the static address specification method is to be used, the LD Player binds the referenced service to the generic service proxy, otherwise it tries to locate and instantiate the service using the GRASP middleware functionalities.

![Figure 3. service description file XML schema](image)

3.3. Services delivery

The main characteristic denoting the IMS-LD Player of the ELeGi project is the ability to discover and run services at run-time (i.e. which are not known prior to the execution of the UoL). Since those services are not installed under the same machine of the IWT-GA Portal server and/or of the Player, but they can be deployed in different places of the VO, the service GUIs are rendered in the Player as portlets through the WSRP.Net implementation (developed in the context of the ELeGi project).

In Figure 4 are outlined the steps executed to render a portlet when a service is delivered in the LD Player.

After that the service is instantiated, the LD Player reads the parameters specified in the service description file and passes them to the service in order to initialize it with all the values needed during the execution. Then the information required to render the portlet are returned while a component called WSRP Consumer, in the LD Player, is responsible to communicate with a WSRP Producer, referenced in the GetWSRPProducerRef() method, in order to get the portlet markup and to render it. The portlet communicates directly with the service to execute the business logic and expose its GUI.

In order to illustrate the above discussion, the “Matchmaking service”, a support service implemented in the EleGi project scenarios, will be used as an example. This service provides a person search support for users that during a learning activity need to look for a tutor or an expert with regard to a specific argument explained during the learning experience.
executeService()

Request for a delivery of a learning service within a learning activity of the Learning Design. The specification of the service are derived from the metadata included in the learning activity environment.

extractServiceSpecifications(String, String)
setParameter(key, value)
generateWSRPProducer()
consumePortlet()
getMarkup()

The LD Player using the service specification create an instance of the learning service and set the configuration parameters.

The LD Player using the information about the returned WSRP producer and portlet reference ask the specific portlet to the consumer.

The portlet communicate directly with the learning service to execute the business logic.

Figure 4. service delivery sequence diagram

Initially, a service description file embedded into the unit of learning (see example in figure 5), is read. Since the address attribute is empty, the dynamic address specification will be used so that all the information needed to retrieve the service URL could be extracted from the SLA element (see the WSLA standard for a better understanding [13]). Than some fixed parameters will be extracted from the “params” section of the service description file; in the example the parameters of interest are “filtered_search” where a value of “true” indicates a search not limited to the people engaged in the same unit of learning of the user requesting the search. and “type_of_role” where a value of “all” specifies to search for learner and/or teacher roles.

Finally the GUI portlet (depicted in the figure 6) will be rendered by the client. It allows to select the concepts and the knowledge level the searched people need to master.

4. Conclusions

In this paper has been presented an approach to integrate web service delivery inside a IMS-LD player. Besides, a modified version of the CopperCore player has been developed in the context of the ELeGI project in order to deliver, within a IMS-LD unit of learning, some dynamically discovered web services exploiting the Grid infrastructure.

The solution proposed seems to be flexible enough to theoretically allow also the delivery of services implemented through the Java language. Indeed the service should only implement an interface (defined in the APIs) and expose it via Web Service. Furthermore, the WSRP Net Consumer implementation has demonstrated in the interoperability tests to allow to consume portlets produced by Java implementations of
the WSRP specification. In the next future the experimentation work will proceed towards this direction.

```xml
<service xmlns="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/XMLSchema-instance"/>

<param>
  <param name="type_of_role" value="all"/>
  <param name="filtered_search" value="false"/>
  <param name="domain" value="Entitlement"/>
  <param name="concept" value="voc01:0002:0000"/>
</param>

<sla xmlns="http://www.ibm.com/wsla" name="WMS_SLA">
  <service provider="Identifier" identifier="Identifier">
    <address>Street</address>
  </service>
  <service provider="ServiceCustomer">
    <name>Street</name>
    <address>Street</address>
  </service>
  <serviceDefinition name="MatchmakingService"/>
  <obligations>
    <serviceLevelObjective name="GlobalResponseTimeObjective">
      <obliged provider="Provider"/>
      <validate/>
      <expression>
        <predicate xsi:type="Less">
          <slaParameter name="globalOperations.CPUUsage"/>
          <value>90</value>
        </predicate>
        <predicate xsi:type="Less">
          <slaParameter name="globalOperations.CPUUsage2"/>
          <value>100</value>
        </predicate>
      </expression>
    </serviceLevelObjective>
    <dependency name="gi"/>
  </obligations>
  <sla/>
</service>
```

**Figure 5.** service description file for the Matchmaking service

References


Figure 6. matchmaking portlet inside the LD Player

Ontologies, Applications Integration and Support to Users in Learning Objects Repositories

DUFRESNE, Aude
University of Montreal
Aude.Dufresne@umontreal.ca

ROUATBI, Mohamed
École Polytechnique
rouatbim@IRO.UMontreal.ca

ABSTRACT We will present how formal ontological representations can be used to bridge across different eLearning applications, to ensure integration and to develop knowledge based support to actors, who are using a Learning Objects Repository. We will present the general framework and how OWL structures are being used to define generic structure and alignment to interlink applications. Structures are used to develop representations of context, concepts, activities, resources and user models, in order to record and exchange them in a SESAME RDF database. Different applications have been interlinked using the system: Explor@Graph, a Computer Supported Experimentation System, a scenario Editor and a Learning Object Repository in the context of support to students and teachers using the LOR.

1. Ontologies and the integration of Learning Objects Repositories

The present research was done in the context of the LORNET effort to develop an integration framework for learning objects repositories. More specifically this research is concerned with designing support to actors and their activities. This meant not only defining support to access resources, but also support to the integration of more complex learning applications and the guiding of activities using them. The architecture to support activities in such a generic and distributed environment was not trivial, it was to offer advice and some control on activities, in a generally flexible environment to suit the typically open tasks like exploring the content or building an eLearning course.

---

1 This research was supported by a CRSNG and a VRQ grants
It was important for the system to offer an open and light integration of components, in order to face the diversity and rapid evolution of eLearning resources. We also wanted to organize it around formal representations, which could be easily changed and made to follow standards being developed in the field (IMSLD) for the exchange of resources. A framework was developed to translate and align specific contexts into this generic form, using OWL representations and a SESAME database to exchange structures and instances of specific ontologies. Using these generic representations, support can be defined to link user overlay models, with contextual navigation activities. Three generic tools are used to make this adaptive integration possible, Explor@Graph that supports generic navigation and retroaction, Explor@GraphNet a more generic web version and the ODIS system which was developed to define the integration between the navigation environment and the representation of a specific context.

![Generic Framework to support users using ontologies](image)

**Figure 1.** Generic Framework to support users using ontologies

This research addresses two problems which are addressed by this research, first on using ontologies to integrate different objects and applications; second, on how ontologies can be used to maintain the models of the domain, of the users, of the tasks, of the context and applications. The solutions to both problems are linked since they rely on the formal specification of communication processes among applications as web services, and on the maintenance of the models and instances described using them.
2. Explor@Graph - A unifying interface for eLearning navigation.

The Explor@Graph interface was designed to be a unifying interface to foster navigation, understanding and the definition of support using conceptual structures of concepts, tasks and documents using typed relations like composition, precedence, resources, production. The conceptual structures may be annotated by the user himself to change his user’s model. Thus each graph or node have static and dynamic metadata associated (time, degree of use and completion, association with messages in the discussion, etc.). These annotations serve as feedback to users and can be used as overlay models by the system to define rule based advices and adaptive support in the interface - open or highlight elements, control elements of the environment, updating and propagation of user models using the structures of relations.

In the context of the LOR, we needed to explore integration with other applications and more complex Learning Objects like computer laboratories. Thus we decided to develop a communication framework between the EG structured description environment, and other applications using ontologies to describe and save the structures, as well as instances and dynamic information on activities using those applications.

3. Using ontologies to integrate applications and support.

As suggested by authors [3-6] ontologies can be very powerful as a description tool but also to support adaptive access, recommender systems or the definition of eLearning scenarios. So to support the activities inside the LORS, we defined an integration framework based on ontologies, among different applications. The ontological structures are used to exchange structured information, across applications. Generally, part of the structure is defined through the various applications of the LOR and functions editor, though to enrich the support, we present a framework where other structures are described for the strategic level and the help in specific applications and domains [7]. Figure 1 presents the general framework where structures are defined:

- The **LOR** is where resources are saved, indexed and grouped. We used the LOR associated with the Explor@ system.
- The **User and Context Manager** where are defined user, groups, roles in relation to tasks, and where resources and privileges are associated to a given context.
- The **Explor@Graph** system (Editor and Navigator) which are used to define graphical navigation structures or functions, relating elements of tasks, concepts or documents, and where support rules can be defined using the properties of the represented structures as conditions and offering as actions MsAgent messages, or control on the described functions (showing properties of a concept or activity, launching resources, etc).
- In the first version [7], to generalize support other structures were defined in XMLSchema to represent other components of support and their instances. The **Generic Rule editor** is then used to create rules linking elements of XML structures.
- The **Generic Support System** is an expert System with a communication interface, that receives events from applications, search for rules and execute them sending back actions to applications, or updating internal models.
The system was used to define contextual support inside a computer supported laboratory [7-9].

In this first implementation of generic support [10], the Explor@Graph system is a fully integrated and usable system to define structures and support, but the integration with other applications is minimal. It proceeds through manual description of XSD structures and XML components of external applications interfaces. Once defined the XSD and XML structures are read into the Generic Rule Editor, where conditions and actions can be defined. Elements of rules may be transferred from the Generic Rule Editor to Explor@Graph, as external actions or conditions, using copy/paste. Elements of tasks and of the control of the External environment can be mixed in rules:

For example a rule may be defined stating that,

\[
\text{If } \text{EG openedGraph = ExperienceOxygen and Plugin sensor = temperature} \\
\text{then MsAgent Prof speak "Good you are using the temperature sensor, In fact, colder water contains more oxygen, let's see if you can measure that.."}
\]

But in this implementation [10], the instances of the structure had to be manually defined. Also, the ontologies were not used to their full potential, without ontology deductions to enrich the support system. Besides, we wanted to be able to persist contextual and user models associated with external applications and to generalize the navigation structures, to make them compatible with other LMS of the LORNET project and especially compatible to IMS-LD description standards.

4. ODIS - Ontological Data Integration System

In the new ODIS framework (Figure 1 lower part) different OWL ontologies are developed using PROTEGE to describe applications and conceptual models that are to be integrated. A composite ontology is developed to define how the ontologies are to be merged and aligned. From this merged ontology, components of interaction between the various applications and a SESAME database are generated using the ShadowGenerator. Those components are also used to define functions to persist information in the database (ShadowPersistor). Once defined conceptually the alignment between the different ontologies, is implemented into the exportation mechanisms using the components generated by ODIS.

A more generic version of Explor@Graph navigator - Explor@GraphNet- has been developed in VB.Net to be used to display structures which are read in SESAME coming from Explor@Graph and other LMS. Thus the functions to help define support inside Explor@Graph can be used to define support in an external environment, using the generic model of the structure and its attributes, using interaction events or user dynamic overlay models. Since the graph structure is generic it can be used to represent different structures of resources, activities, competencies, actors, with the specific properties which have been defined in the ontology.

5. Doctoral Training Environment – Exchanging the structure of activities

The new framework was used to connect the Doctoral Training Environment, a Learning Management System with the Explor@GraphNet graphic interface
The DTE Authoring Tool was built upon CONCEPT@, Télé-université’s authoring environment, in connection with PALOMA, its Learning Objects Repository. Structures of activities and actions are described inside the DTE and are then exported in the SESAME database. The ontology of the DTE had to be aligned to the EGN ontology, so the structure of activity could be presented as a graph. The alignment changes the structures and properties of elements to add visual properties, in the new graph interface, but also to adapt the structure, for example, a property resources in Concept@ will be translated as a new RessourceNode with a specific icon depending on its MediaType, etc.; a completion property will be added to the RessourceNode, which can then be used in the UserModel and the Support Rules. The adapted structure can then be viewed in Explor@Graph and the EGN Graphical Interface. Inside Explor@Graph, generic functions are accessible: spatial organization of the nodes, properties inspection, feedback on navigation and completion, annotations, using the title of the node to search discussions on that subject or resources in the LOR, etc. The Explor@Graph infrastructure to develop and execute support can then be used on those structures to add advices inside the Doctoral Training Environment.

6. Results and discussion

The framework that we present here opens many interesting possibilities. For one, Explor@Graph offers a generic interface for designing support systems. It can be used with its user modeling and support’s rules possibilities. It is now open for importing external structures, IMS-LD structures of activities, conceptual structures, topic maps, which are aligned and exchanged using the ODIS framework. During the exchange process, properties may be added to describe physical properties (position, icons
depending on the alignment model), or user models (completion, importance). Ontologies deductive logic can be used to enrich the structures as they are translated.

The OWL structures which are developed in PROTEGE are easy to visualize and modify. Components (DCM and JCM) which are exported facilitate the development of communication interface between applications and the RDF repository. At run time, each applications can run simultaneously, and update the knowledge base to maintain the contextual and knowledge model. But even more interesting, the use of ontological structures makes it possible to use search and logic reasoning on the information stored and thus augment the power of the Generic Support system.

The framework we have presented is a good step in providing support to activities inside a modular, evolving and opened context such as a network of LOR. In fact, the set of applications we have developed are being used to define support in various context: pedagogical support to students using a Computer Assisted Lab [9], support inside the Adisa course editing system [11]; support to professors for designing and reusing learning scenarios [12]; and support to the Doctoral Training Environment [13].

7. References

Exploring Semantic Description and Matching Technologies for Enhancing the Automatic Composition of Grid based Learning Services

Gustavo Gutiérrez-Carreón, Thanasis Daradoumis and Josep Jorba

Open University of Catalonia, Av. Tibidabo 39-43 - 08035 Barcelona, Spain
{gutierrezc, adaradoumis, jjorba}@uoc.edu

Abstract. In the implementation of e-learning frameworks, a problem still unsolved is how to use and integrate low-level learning services to compose more complex high-level services or tools that make sense to both tutors and learners. In that sense semantic description of Grid learning Services appears like a powerful tool to be used for discovering and matching learning services depending of a set of parameters inside the learning framework. These parameters must represent significant functional characteristics of a learning Grid environment formed by a set of distributed e-learning resources and services. The main objective of this article is to present a review of existing technologies related with semantic description and matching and some techniques used at present to provide Grid Learning Tools and Services automatic composition.

Keywords: Learning Grid, Learning Services, Automatic Composition.

1. Introduction

Distant e-learning emerges as one of promising means for people to learn online. Although there is a substantial increase in computer and network performance in recent years, mainly as a result of faster hardware and more sophisticated software, there are still problems in the fields of integrating various resources towards enabling distant e-learning. Service-based educational systems open new ways in the usability of the Grid as their primary requirements include the provision of adequate services for sharing, syndicating heterogeneous resources and relevant content discovery. The Learning Grid paradigm aims at making use of the collective intelligence and the personalized use of a range of available and potential Grid Learning services. In that sense Grid Learning Services, taking advantage of technological support of Web services in general and Grid services in particular, have fundamentally changed the way that e-learning frameworks were developed. As a learning service example we could consider a Learning Object (LO) [1] from the point of view that a LO is any digital resource that
can be used, reused or referenced to support learning, but not in the sense that it is necessarily designed to explain a stand-alone learning objective.

In the field of Grid services, an important issue is how to achieve the correct integration of inter-organizational and heterogeneous services on the Web. If no single Grid service can satisfy the functionality required by the user, there should be a possibility to combine existing services together in order to fulfill the request. In that sense a lot of efforts have been made to develop techniques and methods for search \([2,3]\), discovery \([4]\), matching \([5]\) and composition \([6,7,8,9,10]\) of grid and web services using semantic description, which showed that important advantages could be achieved if compared with syntactic search.

In general, a framework used for Web service composition \([6]\) (Fig. 1) describes two kinds of participants, service provider and service requester. It contains the following components: a translator, a process generator, an evaluator, an execution engine and a service repository. The service providers propose Web services for use. The service requesters consume information or services offered by the service providers. The translator translates between the external languages used by the participants and the internal languages used by the process generator. For each request, the process generator tries to generate a plan that composes the available services in the service repository to fulfill the request. If more than one plan is found, the evaluator evaluates all plans and proposes the best one for execution. The execution engine executes the plan and returns the result to the service provider.

Fig 1. The framework of the service composition system \([6]\)

The paper takes the above research work into account and proceeds to provide an overall review of those models that mostly contribute to the semantic enrichment of Grid based learning services description and discovery (Section 2). Then, Section 3 goes further and describes technologies for web services matching and composition. Consequently, we propose a conceptual model that explores a new way these models and technologies can be used to enhance the automatic composition of Grid Learning Tools and Services. Section 4 concludes the paper and describes future work.
2. Grid Learning Services Description and Discovery

WSDL (Web Service Description Language) [11] describes the functional information of services such as input parameters, output parameters, service providers and service locations. However, it is limited in supporting the discovery, execution, composition and interoperation of Web services. WSDL cannot provide semantic information of Web services that enable the semantic description of services capabilities.

Currently Globus Toolkit [12] is a common way to implement Grid Services. Globus Metacomputing Directory Service (MDS) implements a standard Web Services interface to a variety of local monitoring tools. Thus, within Globus Toolkit, MDS allows one to register Grid services. Besides it, UDDI has been also used in the web community for business service discovery. Both of them only support keyword based search and are limited in semantic description.

OWL-S [13] is a representative semantic Web service language that arises from the standardization of DAML-S, by integrating OWL-based ontology technology with existing Web service description.

WSMO (Web Service Modeling Framework) [14] provides ontological specifications for the core elements of Semantic Web Services. In fact, Semantic Web Services aim at an integrated technology for the next generation of the Web by combining Semantic Web technologies and Web Services, thereby turning the Internet from an information repository for human consumption into a world-wide system for distributed web computing.

BPEL4WS (Business Process Execution Language for Web Services) [15] provides a language for specifying business processes and business interaction protocols. It can create a composite process by integrating different operations such as Web service call, data manipulation, error report, and process termination.

Nevertheless, these technologies are still immature and incomplete. Moreover, they compete each other; in fact, they still do not provide viable and integrated solutions to the web Services discovery problem.

2.1 Semantic Description of Grid Learning Services

There are some works related to the semantic description of Grid Learning Services. OntoEdu [16] is a flexible platform for online learning which is based on diverse technologies like ubiquitous computing, ontology engineering, Web semantics and computational Grid. It is compound of five parts: user adaptation, automatic composition, educative ontologies, a module of services and a module of contents; among these parts the educative ontology is the main one. The main objectives of OntoEdu are to obtain reusability of concepts, adaptability for users and devices, automatic composition, as well as scalability in functionality and performance. In the near future, this platform aims to be adapted to a Grid environment so that it can carry out its activities based on distributed computing.

The work developed in [17] presents a workflow framework for pervasive learning objects composition by employing a Grid services flow language. The learning objects are distributed in heterogeneous environments which have been used to allow effective collaboration and the reuse of learning objects; this fact can help users learn with no limitations of time and space. This work shows the great opportunities that exist in those research groups which make use of Grid technology to develop innovative, pervasive and ubiquitous learning scenarios. Though this research work is still
encountered at an initial phase, it can be further enhanced by the application of semantic description of learning services.

Finally in [2], the authors have constructed an ontological description for collaborative work tools that allow one to make a manual search of the diverse resources that these tools provide within a Grid environment with the minimum of technical knowledge. This work proposes a Grid-based tool, called Gridcole, which can serve as a basis to implement different conceptual approaches of Grid-based semantic description of learning services, thus extending and endowing it with an innovative, pervasive and ubiquitous projection.

In sum, the works presented above try to provide a solution to the complex problem of grid learning services semantic description, but they are either limited in semantic expressiveness for matching services or they do not face at all the difficult task of using and integrating low-level learning services to compose more complex ones. Both these features could greatly enhance and facilitate the tutor’s and learners’ labor in a complex web-based learning scenario.

2.2 Discovery of Grid Learning Services

Discovery is the process of finding Web services with a given capability [17]. In general, discovery requires that Web services advertise their capabilities with a registry, and that requesting services query the registry for Web services with particular capabilities. The role of the registry is both to store the advertisements of capabilities and to perform a match between the request and the advertisements.

In general, a semantic discovery process relies on semantic annotations, containing high-level abstract descriptions of service requirements and behaviour. Metadata is an essential element in semantic discovery with the capability to expand service descriptions with additional information. The achievement of dynamic composition and automation of services involves discovering new services at run time by software components without human interaction. SOAP provides a description of message transport mechanisms, whereas WSDL describes the interface used by each learning service. However, neither SOAP nor WSDL are of any help for the automatic location of learning services on the basis of their capabilities. Paolucci [18] comments that in order to enable the automation of this process we need a meaningful description of the service and its parameters that can be processed automatically by tools. This implies the possibility to process the context of description by discovery engines.

In this sense, there are some works that aim to improve the semantic services capability of matching. On the one hand, in [19] Paolucci focuses primarily on comparing inputs and outputs of a service as semantic concepts represented in OWL to improve UDDI. This work proposes a way of ranking semantic matching results. This ranking can be used in conjunction with other user-defined constraints to inform of an exact, or potentially useful web-service capability match. On the other hand, there are important lines of research that propose extensions to Web service description WSDL in two ways, annotated WSDL and WSDL-S files [4]. These approaches try to adhere to the current standards while trying to maximize semantic representations required for automation.
3. Grid Learning Services Matching and Composition

There are three principal motivations for Learning Grid Services Composition: build a more powerful service using basic existing services, fulfill service requester’s requirement better, and enhance resource reuse while reducing the cost and time of a new service development. IMS Global Learning Consortium\(^1\) proposes an abstract framework [20] representing a set of services used to construct an e-learning system in its broadest sense. Fig 2. shows the dependencies between the different “layers” of the framework.

The Learning Application composition process consists of identifying sub-tasks of the learning process, locating suitable Learning application Services to construct each process, locating suitable Common Services to construct each learning service, formatting the Learning and Common services into a service flow and executing the service flow to achieve a task which is the goal of the learning process.

The core stage is the composition of learning web services and their adaptation to the needs of a learner or group of learners [21]. Such a composition is carried out by retrieving previously registered objects. Once composed and packaged as learning objects, these composite processes can be executed and then instantiated and adapted to the learner’s particular needs.

These adaptations can be realized, either by predefined rules implemented into the process description and driven by the learner behavior, or in a supervised manner. In the later case, the instructional designer can return to the composition tools to adapt the process.

![Fig 2. IMS Abstract Framework](image)

Dealing with the specific problem of constructing a suitable workflow for a learning scenario, in [22] the authors propose a framework to facilitate automated composition of scientific workflows in Semantic Grids made up of a Manager Service and other supporting services, including an abstract and a concrete workflow generator. They described these components in detail and outlined their interactions. Finally, they described their implementation and its use within the physics domain. The important features of this approach are: an adaptive workflow generation algorithm and the distinction between different levels of abstraction of the workflow in order to allow reuse and sharing.

\(^1\) IMS develops and promotes the adoption of open technical specifications for interoperable learning technology.
Furthermore, there is a detailed development of learning services matching procedures [5] for locating the most suitable Learning Services, combining and integrating a number of matching algorithms, and adopting two principal approaches: the structural matching approach and the linguistic or syntactic approach. This work focuses on the issue of searching a Web Service with required functionalities and addressing a specific application domain, by means of an ontology-based semantic description.

3.1 A Conceptual Model for Grid Learning Services automatic composition

The model we propose for the automatic composition of learning services (Fig. 3) is based on the use of the defined syntactic and semantic characteristics for the different levels of services involved in the Learning Abstract Framework.

![Fig 3. Grid Learning services automatic composition](image)

The first step to carry out the automatic composition is to generate a Semantic schema of the learning tool or learning services that will be composed. This schema will be able to be constructed using the different tools of descriptions at the "syntactic level" through WSDL, or at the semantic level, through service ontologies included in OWL-S, WSMO, SWSF and WSDL-S[5].

Once the semantic schema of the tool or learning service that we want to build is designed, we have to pass it to our discovery process that will locate a set of different level services in the Learning Grid. The operation of these services as a whole allows to carry out the processes defined in the schema. The result of the search will be a group of suitable schemas that conforms to the functional process described in our initial schema.

These resulting schemas will be compared to the initial schema through a Matching process that is based on a structural matching approach and on a linguistic or syntactic approach and whose result will be the best evaluated schema for our learning tool or services.

Comparing our conceptual model with the works presented in [2, 7, 16 and 21], our approach represents a complete alternative since, on the one hand, we provide a multi-level learning services composition method that enables the construction of complex learning services by means of other low level services, depending on the nature of the learning abstract framework. On the other hand, our approach takes advantage of the semantic and syntactic characteristics of learning services, which
facilitates a totally automatic construction of new learning tools based on others previously created.

4. Conclusions and future work

In this work we review some methods and techniques for automatic composition of Grid based Learning Services. In that sense we highlight the importance of defining a contextual based semantic model of the Learning scenario, which is particularly significant in semantic based automatic service searching, discovery, and composition. Future work aims at the implementation of the conceptual model presented in this work in a real, learning collaborative scenario based on Grid.

References


Sycara, Bringing Semantics to Web Services: The OWL-S Approach, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, ISSN 0302-9743, Volume 3387, 2005


The Unfolding of Learning Theories: Its Application to Effective Design of Collaborative Learning

Seiji ISOTANI and Riichiro MIZOGUCHI
The Institute of Scientific and Industrial Research, Osaka University, Japan

Abstract. Although artificial intelligence has been successfully introduced to enhance Education through technologies in the past few years, major challenges still remain. One of them is how to represent the “knowledge” of intelligent authoring systems. To represent the knowledge of authoring systems to support collaborative learning is particularly challenging because it is based on various learning theories and given the complexity of group learning where the synergy among learner’s interactions affects the learning processes and hence learning outcome. The main objective of this work is to introduce an ontological infrastructure on which we can build a model that describes learning theories and to show how we can use them to develop programs that provide a graphical visualization of learning theories and intelligent guidance for an effective design of group activities.

Keywords. Collaborative learning, ontological engineering, intelligent educational system, learning theories.

Introduction

In recent years, with the increasing use of technology, Artificial Intelligence has been gradually and successfully introduced into Education. However, major challenges still remain. Among these, we are interested in how to represent the knowledge of intelligent authoring systems (IAS) and then how to use this knowledge efficiently, especially within the context of collaborative learning.

Usual approaches to such issues provide their systems with a kind of expertise using a set of heuristics and domain theories built in the procedures (programming languages). This means that the programmers, not the systems, have an understanding of the knowledge being used. As a result, these systems cannot share or build new knowledge, ignore the existence of theories on which the knowledge is based, and finally cannot justify their recommendations systematically and scientifically [2; 15].

To develop IAS to support collaborative learning (CL) is especially challenging in view of knowledge representation. Current knowledge concerning CL is based on various learning theories, which are always expressed in natural language and are particularly complex given the context of group learning where the synergy among learner’s interactions affect the learning processes and hence learning outcome. It is in
fact currently difficult for both humans and computers to clearly understand and differentiate between the various learning theories, yet, without their explicit representation, it is difficult to support the design of group activities based on well-grounded theoretical knowledge.

Our approach calls upon techniques of ontological engineering to, at first, establish a common understanding of what a learning theory is by representing it in terms of its explicitness, formalism, concepts and vocabulary. This makes theories understandable both by computers and humans. We then propose techniques of reasoning on these theories which contribute to dynamic guiding and instructional planning. And finally, we present the CHOCOLATO - a Concrete Helpful Ontology-aware COLlaborative Learning Authoring TOol focusing on a sub-system that represents theories graphically to facilitate the design of effective CL activities with theoretical justifications.

1. Collaborative Learning and Learning Theories

Collaborative learning has become a popular method used by teachers in classrooms and in e-learning environments. In spite of that, designing effective CL sessions or analyzing the interaction processes among learners to capture what really happens in each session have been a very complex job due to a lack of comprehensible models for representing what is going on [10]. According to Dillenbourg [7], the key to understanding collaborative learning is to gain an understanding of the interactions among the individuals.

Many learning theories contribute to in-depth understanding and the support of collaborative learning (for instance, peer tutoring, anchored instructions, etc). However, it is not common to find models that allow explicit representation of these theories. One of the reasons is the difficulty to understand the theories due to its complexity and ambiguously. According to [9] different theories can describe the same situation using different terminologies. Moreover, each theory has its own point of view, learning focus, structure, besides many other aspects that need to be considered.

Therefore, to provide systems with theoretical knowledge for collaborative learning we must: a) to establish a common conceptual infrastructure on which we can build a model that describes what a learning theory is and what a collaborative learning is; b) to clarify how learning theories can help the design of group activities and enhance learning outcomes; and c) to propose models and structures that enable the sharing of findings and the use of computers to support the analysis and design of effective CL sessions in compliance with theories.

To deal with the problems presented above we provide a model based on an ontological structure to describe learning theories for collaborative learning and techniques to use it rationally. With that we aim to establish the initial foundations for the development of ontology-aware authoring systems for CL.

2. Graphical Representation of Learning Theories

The use of ontological engineering for knowledge systematization has shown significant results to bridge the deep conceptual gap about how to represent the knowledge of authoring systems and web-based educational environments considering educational theories [6; 9; 15]. In practical terms ontological engineering helps to
achieve the following [5; 16]: (a) A common vocabulary and highly structured definition of concepts; (b) semantic interoperability and high expressiveness; (c) coherence and systematization of knowledge; and (d) meta-models and foundations for solving different problems in a variety of contexts.

In CSCL (Computer Supported Collaborative Learning) research, ontologies have been successfully applied to solve problems such as: group formation [20], CL representation [12], interaction analysis and patterns [10] and modeling of learner’s development [11]. With these achievements it is possible to some extent to successfully identify which kind of collaboration occurs in a CL session, to understand the essence of the group's interactions, and to estimate the expected educational benefits for each learner. Nevertheless, there are some limitations: (a) there is no explicit relation among interaction patterns and learner’s development model; (b) it is not easy to determine what learning theory is appropriate for explaining the learner’s development through a set of events; and (c) it is difficult to propose activities in compliant with the theories to enhance interactions among learners and lead them to achieve desired goals.

To overcome these limitations we re-analyzed seven different learning theories frequently used to support CSCL activities: Cognitive Apprenticeship [3], Anchored Instruction [4], Peer Tutoring [8], Cognitive Flexibility [19], LPP [14], Socio-Cultural Theory [21] and Distributed Cognition [18]. Then, we proposed the Growth Model Improved by Interaction Patterns (GMIP) [13] to unify and to improve the benefits of two successful previous models that offer (a) an explicit representation of typical interactions based on learning theories [10]; and (b) a simplified way to represent the learner’s growth (knowledge acquisition and skill development) [11]. With the GMIP we clarified how learning strategies prescribed by learning theories can help learners to acquire desired goals and explicitly identify the relationships among interactions, learning strategies and learning goals.

The GMIP is a graph model based on an ontological structure to describe an excerpt of learning theory [13]. It represents, in a simplified way, the learner's knowledge acquisition process and skill development process, explaining the relationships between learning strategies, educational benefits and interactions used to achieve these benefits. For such representation, we have to explain more about two processes: knowledge acquisition and development of skill.

The process of acquiring specific knowledge includes three qualitatively different kinds of learning: accretion, tuning and restructuring [17]. Accretion is to add and to interpret new information in terms of pre-existent knowledge. Tuning is to understand knowledge through application of this knowledge in a specific situation. Restructuring is to consider the relationships in acquired knowledge and thus to rebuild the existent knowledge structure.

Considering the development of skills, there are also three phases of learning: the cognitive stage (rough and explanatory), the associative stage and the autonomous stage [1]. The cognitive stage involves an initial encoding of a target skill that allows the learner to present the desired behavior or, at least, some crude approximation. The associative stage is the improvement of the desired skill through practice. In this stage, mistakes presented initially are gradually detected and eliminated. The autonomous stage is one of gradual continued improvement in the performance of the skill.

Using these concepts, the GMIP graph has twenty nodes (Figure 1), which represent the levels of the learner’s development at a certain moment of learning. Each node is composed by two triangles. The upper-right triangle represents the stage of knowledge acquisition, while the lower-left triangle represents the stage of skill
development. The nodes are linked with arrows that show possible transitions between nodes in compliance with [1] and [17]. $s(x,y)$ is the simplified form of representing these nodes in our model: $x$ represents the current stage of skill development and $y$ represents the current stage of knowledge acquisition. For instance, $s(0,0)$ represents the node where the stage of knowledge acquisition and skill development is nothing; and $s(0,1)$ represents the stage of knowledge acquisition is accretion and the stage of skill development is nothing.

Using the GMIP graph, we show the benefits of learning strategies by highlighting its path on the graph and associating each arrow with the interactions. In Figure 1 we show an example of the GMIP graph for the learning strategy “learning by apprenticeship” used by the learning theory “Cognitive Apprenticeship”. Bold arrows represent the transition from one stage to the other, which is facilitated through this learning strategy using the labeled interactions. There are two kinds of interactions: the necessary interactions, represented by a black circle, and the complementary interactions, represented by a white circle. The interactions are linked by ellipses. The dashed ellipse represents a directed link between two interactions and the full ellipse represents a cyclical link between two interactions.

The GMIP clarifies, more precisely, how interactions can affect learner’s development, facilitating the learning design based on events. Thus, it becomes a powerful tool helping designers to select events (interactions) and roles for each learner considering interaction patterns and learning strategies appropriate for desired learning goals and sub-goals (and vice versa). Furthermore, we believe this model is the first step to explain what a learning theory is, making tacit characteristics explicit: for instance, clarifying expected benefits, use restrictions, guidelines for leading/performing activities, in addition to other important aspects of the teaching-learning process.
Another intriguing feature of GMIP that deserve some attention is the possibility of blending learning strategies. Because each strategy is intrinsic represented as paths on the GMIP graph, we can find common points (stages) between strategies, and thus, provide guidelines to blend learning theories by “linking” two or more strategies from different theories to achieve a desired goal. Considering such possibility during the design process a user could choose one strategy to lead learners to obtain some benefits and after change to another strategy to obtain other benefits that the first strategy could not offer. Note that we are not trying to say that it is possible to blend any strategy and any theory, what we want to point out is: if we deeply understand the theories providing formal methods to represent them explicitly, it is possible to identify common points among theories and then propose techniques to blend them rationally. To blend learning theories for CL is a challenging task and will be addressed more carefully and deeply in future research.

In summary, the main contributions of GMIP for CL design are (a) to allow the graphical visualization of theories and their characteristics. Thus, users can quickly interpret the theories, their benefits and propose sequence of activities in compliance with them; and (b) to provide a formal structure based on ontologies which allows systems to reasoning about the theories and the features (actions, roles, strategies, etc.) prescribed by them. Thus, it is possible to offer new alternatives for intelligent guidance (as shown in section 4) providing suggestions of CL activities for users during the design process.

3. Towards a Complete Ontology-aware Authoring System for CL

As we mentioned before to propose a group formation there are many learning theories such as Anchored Instruction, Peer Tutoring, Cognitive Apprenticeship, etc. Then, to assign roles and strategies for members of a group we can select appropriate set of learning theories considering the necessary pre-conditions of learners and the educational benefits we expect to be improved for each learner in the end of a CL session. This flexibility of choosing different learning theories can therefore provide us with many ways to design and conduct learning processes. However, it also suggests the difficulty of selecting the appropriate set of learning theories during the instructional design to ensure learners’ benefits and the consistency of learning processes. Therefore, to help users (instructors, teachers, designers, etc) to design effective group activities we need an elaborated authoring system that considers different learning theories to support the design in compliance with them.

According to [2; 15] there is a deep conceptual gap between knowledge of authoring systems. Because of that these systems cannot share or accumulate new knowledge, usually are based on only one theory that is built in the procedures (programming code) to support the design of learning activities, and do not justify their recommendations systematically and scientifically.

To solve these problems we have been developing a theory-aware authoring system for CL, called CHOCOLATO – a Concrete and Helpful Ontology-aware Collaborative Learning Authoring Tool. It is based on our model GMIP and the ontological structure to describe learning theories, besides previous achievements presented in section 3. Through the use of ontologies, the theories and their features are declaratively and formally represented which (a) prevent unexpected interpretations of the theories; (b) provide a common vocabulary to describe them; (c) enable us to share
and accumulate the knowledge; and (d) provide enough information for computational semantics to provide assistance for users based on theories. Furthermore, through the use of GMIP the system offers graphical and textual support for users providing “intelligent” guidance with theoretical justifications during the authoring process.

The architecture of CHOCOLATO is shown in Figure 2. It is proposed to support different levels of guidance during (a) group formation that maximize the educational benefit considering the individual and group goals; (b) designing of CL activities; (c) recommendation of learning materials; (d) analysis of individual and group outcomes minimizing the difficulties during this process; and (e) proposing new CL sessions based on previous group’s outcomes.


This system assists both novice and expert users. For example, during the design process, for novice users, the design manager of CHOCOLATO provides a structured guidance considering different learning theories. Through an authoring interface using the GMIP it allows users to set initial conditions and goals for a learner or the group and the system automatically recommends theories, strategies, roles and activities to be performed by learners to achieve the desired goals. Furthermore, users can customize the recommendations in order to satisfy requirements depending on particular situations. For expert users, it offers a common language and guidelines to formally express CL activities, the interactions’ flows learner’s roles, strategies and benefits for learners. Thus, it is possible to describe new strategies and roles for learners, reuse and share them, and finally combine sequence of interactions to fit in different scenarios.

Considering the interaction analysis, it is difficult to know when learners acquire the desired benefit because we need to capture what roles the learners played and what kind of interactions occurred in the session. To help such process, the analysis system of CHOCOLATO identify when a CL session proceeds conform the initial scenario designed by the user. Thus, we can predict whether the learners interacted as expected and whether the CL session was successful or not. It is worth to point out that if the initial scenario of a CL session is not established previously it is much more difficult to expect concrete benefits and to analyze (quantitatively and qualitatively) how much benefits were attained by learners.

Thus, focusing on design process, in order to produce effective CL sessions, we would like to present a sub-system of CHOCOLATO (shaded block in figure 2) used to support the design of CL activities. This sub-system is called MARI – Main Adaptive Representation Interface. It is an ontology-aware system that uses ontologies developed in Hozo ontology editor (http://www.hozo.jp) to provide its theoretical
knowledge and represent them on the screen using the GMIP. Through the use of ontologies MARI allows high expressiveness and interoperability among theories and their features. Nowadays MARI has 6 theories and 12 strategies, besides other information in its database.

MARI starts with a neutral network (Figure 3a) that can represent any theory we analyzed by selecting theories/strategies through MARI’s interface. MARI reasons on the ontologies to provide a graphical visualization of them using the GMIP (Figure 3b) offering an easy and quick interpretation of necessary pre-conditions and educational benefits for learners. Furthermore, by clicking in the bold arrows the system can suggest CL activities (interactions) prescribed by the selected theory which help learners in one stage to achieve the next stage as shown in Figure 1. Each interaction will be depicted on the arrows in the next version of MARI.

Another useful function in MARI is to search theories by given a stage of learner’s development. We can select an initial stage of a learner in the GMIP and the system will reason on the ontologies to search for any theory/strategy that has the selected stage in the beginning of the path. As same as before we can select a final stage and the system will search for any theory/strategy that has the selected stage in the end of the path. And finally, the system can search for any theory/strategy that has a path through the selected stage (it means any stage in the path). All these ways of search can be combined, thus, users can select, for example, an initial stage (pre-conditions) and a final stage (expected benefits) of a learner and the system will find the theories/strategies that help this learner to achieve the desired benefits. If more than one theory/strategy is found, users can select one of them and the system suggests activities in compliance with it.

In case we do not find any theory/strategy that helps a learner (or a group of learners) considering his initial conditions (initial stage) and desired goals (final stage), the idea of blended learning theories in the end of section 2.2, could be considered. In such a case a possible solution to help a learner is to use the GMIP to work with
theories at the macro-level (strategies, learner’s stages, etc) to select a strategy \textit{S1}, which help learners in an stage \textit{G1} (initial stage) to achieve a following stage \textit{G2} (sub-goal), and then, to select another strategy \textit{S2}, which help learners in an stage \textit{G2} to achieve a following stage \textit{G3} (final stage) that cannot be achieved by \textit{S1}. After that, working with theories at the micro-level (activities, learner’s roles, etc) the system is able to identify the sets of interactions of the strategies \textit{S1} and \textit{S2}, combining them rationally, to finally, propose a sequence of CL activities that maintain the consistence of the learning process. These steps enable us to connect the strategies \textit{S1} and \textit{S2} and to create suitable set of interactions to help a learner in the initial condition, \textit{G1}, to achieve his desired goals, \textit{G3}.

To completely realize blended learning for CL it is necessary to consider the relationships among many assumptions described by theories (for instance, context, delivery methods, learning preferences, etc), besides the synergy among learners in a group. It is our intention for future research to include a study demonstrating some examples and possibilities to blend learning strategies semi-automatically.

Using ontologies and the GMIP it is feasible for our system to reason on the theories at the macro and micro levels and to create a link between them. This link allows us to select appropriate learning theories and strategies at the macro-level and to suggest consistent sequence of activities for learners in a group at the micro-level.

The suggestions given by our system are only guidelines for users to propose CL activities based on theories which (a) preserves the consistency of the learning process; and (b) guarantees a suitable path for learners to achieve desired benefits. However, expert designers do not need to follow the suggestions. They can propose their own path on the graph and their own sequence of activities. In such case the system also can assist these users providing different kind of information about theories, activities, strategies, learner’s roles and other related information that can be useful in various situations.

4. Conclusions

The main contribution of this research is to introduce our model GMIP based on an ontological structure to describe learning theories for CL and create techniques to use it rationally. This is another step forward in the improvement of ontology-aware authoring systems that offer intelligent guidance to design CL activities supported by theoretical knowledge that solves, at least partially, the problems of knowledge representation presented in [15]. The proposed system MARI supported by our model GMIP and theories described as ontologies allow us to work with theories at the macro and micro levels and to create a link between them. This link clarifies, more precisely, how interactions can affect learner’s development which helps designers to select interactions and roles for each learner with justifications based on the theories. It also allows us to reasoning on these theories semi-automatically to suggest consistent sequence of activities for learners in a group.

We also showed roughly the intriguing possibility of blending learning theories using our model and our system as a feasible solution to deal with the problem of unreachable stages (stages that none of the analyzed theories has a path through it by itself). In such a case, during the CL design the system can suggest for users a set of activities supported by blended theories to find a suitable way to lead learners to
achieve desired benefits. Our future work will deal with many open questions about blended learning for CL to improve our model and our system.

One delicate point we would like to emphasize is the necessity of sophisticated group formation to set strategies, roles and activities for each learner before a CL session starts. We believe that the design of CL sessions is a requisite to maximize educational benefits and to minimize the load of interaction analysis. Such approach creates favorable conditions for learners to perform CL activities and help users to estimate more easily how much benefits learners attain in the end of a session. Our approach uses theory-driven group formation with suggestions of role assignment and sequence of interactions to offer fundamental settings for an effective CL session and essential conditions to predict the impact of interactions in the learning process.

The possibility of clarifying what a CL session is and to amplify its educational benefits has been a great challenge. In this context our approach offers a declarative representation of learning theories allowing computational semantics to support the design of CL sessions in compliance with well-grounded theoretical knowledge and, because it can be explicitly demonstrated, is much more convincing and flexible than usual approaches.

References


Ontological Framework for Educational Feedback

Jelena JOVANOVIĆ1, Shilpi RAO2, Dragan GAŠEVIĆ2,3, Vladan DEVEDŽIĆ1, Marek HATALA2
1FON-School of Business Administration, University of Belgrade, Jove Ilića 154, Belgrade, Serbia
2School of Interactive Arts and Technology, Simon Fraser University, 102 Ave., Surrey, BC V3T 5X3, Canada
3School of Computing and Information System, Athabasca University 1 University Drive, Athabasca, Canada

Abstract. The paper proposes an ontological framework, named LOCO, for automatic provision of educational feedback. The framework is based on ontologies and is applicable to various e-learning systems. The use of ontologies is motivated by the increasing demand for effective mechanisms for knowledge management in e-learning, and in particular for capturing and integrating data about various learning contexts. To address this need, we employ ontologies to formalize the concept of learning object context. We define this concept as a specific learning situation, determined by the learning activity, the learning content, and the (group of) learner(s) involved. To illustrate practical implications of the LOCO framework, we present two educational applications that we have implemented on top of LOCO, namely: LOCO-Analyst – a system for provision of teacher-and-author-centric feedback about learners’ activities and use of learning content; and MICE – a learner-centric platform used for regulating learners’ programming styles when studying a programming language using an integrated development environment.

Keywords. Ontologies, Educational Feedback, Learning Context.

Introduction

Semantic Web technologies seem to be a promising technological foundation for the next-generation of e-learning systems [1]. The pioneering work on using ontologies as the backbone of e-learning systems is presented in [2]. Since that time, many authors have proposed the usage of ontologies in different aspects of e-learning, such as adaptive educational hypermedia, adaptive content authoring, personalization, user model sharing, and context capturing [3] [4]. This is an expected reaction, since e-learning is highly dependent on effective mechanisms for knowledge management capable of integrating the artifacts of various activities that e-learning involves. To illustrate this, let us name some of the activities indispensable for securing high quality of the e-learning process: preparation of learning content; structuring and organization of the content in accordance with the chosen teaching strategy (i.e., instructional design); interacting with students using online communication facilities (such as
discussion forums and chat-rooms); moderating students’ online collaboration; adapting the materials constantly in order to improve the students’ learning efficiency and effectiveness.

Most of the above groups of activities are nowadays largely supported by Learning Content Management Systems (LCMSs) – a widespread learning technology that enables deployment and management of online courses. However, support for adaptation of learning materials is scarce, and most widely used LCMSs include only simple content editing features for this purpose. To be able to adapt the learning content to the students’ needs, online educators have to have appropriate and reliable feedback about the students’ usage of the learning materials. Unlike traditional learning settings where teachers have consequential awareness of what their students are working on and how satisfied they are with the content, most LCMSs provide only simple statistics about the tools the students have used, and only high level interactions with learning content (e.g. page views). Therefore, our goal is to facilitate the job of online educators by providing them with useful feedback about the students’ online activities and their usage of learning content. With such a feedback available, the educators are better able to further improve the content and instructional design of the course(s) they teach. Of course, this is just one side (i.e., teacher-centric) of the story, but another equally important one is to develop learner-centric systems. Such systems observe interaction of students with educational systems, and accordingly provide them with, for example, feedback on how to regulate their learning activities or personalize their search for learning materials or peers.

In this paper, we demonstrate how we employ Semantic Web technologies, namely ontologies and semantic annotation [5], to improve the current state-of-the-art in generating feedback in educational systems. The proposed solution exploits the nature of ontologies to integrate various sources of information covering the same domain knowledge. In our case, we use ontologies to interrelate information about learning objects, learning activities and learners captured from various tools (e.g., LCMSs and various educational tools and environments that learners use in classroom). Furthermore, we use domain ontologies to relate different kinds of learning artefacts (e.g., lessons, tests, chat messages) by exploiting semantic annotation techniques [5] [6]. In particular, we propose the Learning Object Context Ontology (LOCO) framework, based on the notion of learning object context as defined in [7], as a bridge between usage data collection and its meaningful representation for online educators. The LOCO framework can be used to represent formally diverse kinds of learning situations (i.e., learning contexts) typically occurring in modern LCMSs. In addition, its flexible design makes it easily extensible, so that it can support any tool or system used in online education. On top of this ontology framework, we have implemented two educational systems, namely: LOCO-Analyst – a teacher-centric system that generates different kinds of feedback based on learners’ activities done in different learning contexts, and MICE – a learner-centric framework that facilitates learning programming languages by using various software engineering toolkits.

In the next section we present the LOCO framework, the LOCO-Cite ontology as its core part, and the extensions that we have so far built. In section 2, we present two examples of applying the LOCO framework in practice: the LOCO-Analyst tool and the MICE platform. Related work is presented in section 3, whereas section 4 concludes the paper and indicates further directions of our work.
1. Ontology Framework

In our previous work [7], we introduced the concept of learning object context as a specific learning situation, determined by the learning activity, the learning content, and the (group of) learner(s) involved. We have also developed the LOCO ontological framework aimed at its formal representation. Accordingly, the framework integrated a number of learning-related ontologies such as user model ontology, an ontology of learning design, and content structure ontology. Its design reflected our intention to utilize the elements of learning context to facilitate advanced levels of personalization and reuse of learning objects and learning design. Since we had expected that LCMSs would evolve to support the new standard for learning design, namely the IMS Learning Design (IMS-LD) specification, we wanted to develop a framework that would support it on the Semantic Web. However, it turned out that the adoption of IMS-LD is a very slow process and that neither the state-of-the-art LCMSs support it nor it gained wider acceptance among online educators. Therefore, we redesigned the framework to make it more compliant with the requirements of the presently available e-learning tools. We did this in order to make Semantic Web technologies applicable in practice and show the benefits of their usage in real-world settings, and not just contribute to academic discussions on the issues of learning design and the usefulness of its ontological representation. Hence, the redesigned framework reflects more accurately the requirements of the presently available e-learning tools. In the following subsections, we first present the core part of the framework – the LOCO-Cite ontology in its new look-and-feel, and then present its extensions for a mixed-initiative coding environment.

1.1. The LOCO-Cite Ontology

The focal point of the LOCO-Cite ontology is the `LearningObjectContext` class (Figure 1) which is, in accordance with our definition of the learning object context [7], related to the activity (`Activity`) that a learner or a teacher (`um:User`) undertook while interacting with a learning content (`ContentUnit`). An instance of `LearningObjectContext` is always related to exactly one `Activity` instance as well as one `ContentUnit` instance. However, it can be related to more than one `um:User` instances in case of a collaborative activity engaging more users (e.g., various forms of online discussions). Figure 1 presents the core elements of the ontology.

We have identified a few basic kinds of learning activities (i.e., activities typically occurring in every virtual learning environment) and modelled them as subclasses of the `Activity` class. By analyzing the state-of-the-art LCMSs, we have recognized that students are reading some learning content (`Reading`), or doing an assessment (`Quizzing`), or interacting with other participants in the learning process (`Discussing`). In addition, we distinguished between synchronous (e.g., chatting) and asynchronous interactions (e.g., participating in discussion forums) and modeled these by subclassing the `Discussing` class with `DiscussSynchronously` and `DiscussAsynchronously` classes, respectively. Each activity can comprise one or more events (an instance of the `Event` class). For example, we have recognized various kinds of discussion events, i.e., events related to the students, and occasionally teachers, activities in discussion forums. These events are represented in the ontology as subclasses of the `DiscussionEvent` class (which in turn subclasses the `Event` class). Each recognized kind of activity and event is further formally described with a set of ontological classes and properties.
Bearing in mind that there is a plethora of e-learning systems, having different architectures, various purposes, and offering different services and tools, we have designed the LOCO-Cite ontology to be easily extensible, and thus applicable for various learning environments and their usage-tracking capabilities. We have already developed an extension for Mixed Initiative Coding Environment (MICE). It is the topic of the next section.

1.2. Extensions

Technically, MICE is a modelling and reasoning framework providing intelligent support for multiple software engineering toolkits in order to facilitate code development. It integrates a set of external tools, such as tools for communication and collaboration (e.g., code sharing) among users. To enable formal representation of specific types of user-system interactions supported by the tools and systems integrated into MICE, we have introduced an additional set of classes and properties in the LOCO-Cite ontology. The tools currently integrated in MICE are: BlueJ – a Java IDE for beginners; gStudy – a learning environment for monitoring learners’ work with online learning material and their participation in online discussions; and iHelp Courses LCMS [8]. Therefore, we have extended LOCO-Cite to support interactions specific to these tools. In particular, we have defined one extension for each tool integrated into MICE. For example, the gStudy extension introduces new elements to formally represent gStudy’s events related to link-creation, highlighting, browsing, and searching [9].

MICE employs a formal computational representation of the theory of Self-Regulated Learning (SRL) [10] as the context for human-initiated and system-initiated interactions. SRL-based feedback depends on the system’s ability to recognize and respond to the events initiated by the learner. All kinds of events specific to the MICE framework are represented as subclasses of the loco:Event class (Figure 2a). The semantics of the classes shown in Figure 2 can be intuitively inferred from their names; hence we will focus on those classes that are not self-explanatory enough.

The gStudy extension. A highlight event is triggered by the gStudy learning environment when a user attempts to highlight any portion of the currently viewed content. In addition, gStudy allows the user to ‘annotate’ the highlighted portion of the content with terms such as ‘important’, ‘doubt’, ‘to discuss’, and so on. The HighlightEvent class is introduced in the gStudy extension to represent formally such an event. It captures the time when the event took place (via the loco:timestamp property inherited from the loco:Event class), the content that was highlighted (content), and the qualitative clues assigned to the content (typeOfHighlight). The range of the typeOfHighlight property is set to the union of instances of the HighlightType class, each one representing a qualitative clue a user can attach to the highlighted content.
A link event is triggered by the gStudy environment when a user creates a link between two sections of the currently examined content. Accordingly, we have introduced the `LinkEvent` class, which captures the time when the event was triggered (`loco:timestamp`), the ‘source’ (`linkFrom`) and the ‘destination’ (`linkTo`) of the link, as well as the type of the link (`typeOfLink`). The `typeOfLink` property captures the semantics that the user assigned to the link, and is restricted to the predefined set of instances representing various types of links (`LinkType`). Of course, this set of link types can be extended to support other link types found in other systems. The same stands for the abovementioned type of highlight.

The gStudy extension introduces four classes to represent kinds of chatting-related events (subclasses of `ChatEventClass`, see Figure 2a). Even though, we identified chatting as a kind of activity in the LOCO-Cite ontology (`loco:DiscussSynchronously`), we did not explicitly model the events that can occur within it, since the LCMSs that we had analyzed do not provide support for capturing them. However, we did recognize that other learning environments might be able to register events of any learning activity, and therefore established a relation (via the `hasEvent` property) between an activity and its events (Figure 1). In the gStudy extension, we have restricted the `hasEvent` property of the `loco:DiscussSynchronously` class to the instances of the `ChatEvent` class using the `owl:allValuesFrom` restriction.

Figure 2. The hierarchy of classes representing different kinds of events that can be captured in the MICE environment (a); classes of the BlueJ extension (b)
The BlueJ extension. BlueJ (http://www.bluej.org) is a programming IDE for beginners. MICE integrates the SRL principles for providing feedback and suggestions to a beginner programmer based on his/her observed interactions with the IDE (BlueJ in particular). Therefore, the `ProgrammingEvent` class has a number of subclasses representing different kinds of programming related events that can be tracked in BlueJ (Figure 2b). A compile event (modeled with the `CompileEvent` class) is triggered by the BlueJ IDE when a user compiles some Java code. The compilation may pass successfully, or signal either an error or a warning. Accordingly, we have introduced the `CompileOutcome` class (directly under `owl:Thing`) and modeled the possible outcomes of the compile event as its subclasses: `CompileError`, `CompileWarning`, and `CompileSuccess`. We have also added a closure axiom on the `CompileOutcome` class, so that there cannot be a `CompileOutcome` instance that is not an instance of one of the aforementioned subclasses. A compile event is related to its outcome via the `resultsIn` property having the `CompileEvent` class as its domain and the `CompileOutcome` class as its range. In addition, every `CompileEvent` instance keeps track of the time the user compiled the code (`loco:timestamp`), the number of bytes in the code at the moment of the compilation, and the name of the program being compiled.

The BlueJ IDE fires an event each time a user adds or removes a programming language construct. The BlueJ extension enables one to capture data about these events through the `ProgConstructEvent` class and its subclasses. In addition to capturing the time of the event, these classes are modeled to keep track of the type of the construct added/removed (`typeOfProgConstruct`). We use concepts of the Java programming language ontology developed in the scope of the Personal Reader (http://www.personal-reader.de/) project to formally represent the kind of Java programming language construct related to an event.

The LOCO-Cite ontology can capture activities and events common to the great majority of LCMSs. However, almost every LCMS has some specific tools that require extending LOCO-Cite to enable capturing the interaction events these tools can generate. In the case of iHelp Courses LCMS, such a specific tool is the code-sharing facility [11]. We are currently developing an extension of the LOCO-Cite ontology for capturing the specific kinds of events that this tool can generate, and hence enable its full integration in the MICE platform.

All ontologies are developed in OWL ontology language and can be inspected from the following URL: http://lore.iat.sfu.ca/projects/LOCO-Analyst/.

2. Application Examples

In this section, we present two examples of applying the LOCO ontological framework in practice. First, we present the LOCO-Analyst tool that exploits ontological representation of the learning object context data to generate feedback for online educators. Subsequently, we show how the extended LOCO-Cite ontology enabled us to develop the MICE framework that provides real-time feedback to the users to help them reflect upon their programming styles and improve them.

2.1. LOCO-Analyst

LOCO-Analyst aims at helping instructors rethink the quality of the learning content and learning design of the courses they teach. To this end, it provides instructors with
feedback about the relevant aspects of the learning process taking place in the online learning environment they use. The provided feedback is based on the analyses of the log data tracked by the learning environment.

To identify the kinds of feedback online instructors would appreciate we conducted a small-scale survey [12]. The feedback provision functionalities implemented in LOCO-Analyst reflect the results of the survey. In particular, LOCO-Analyst informs instructors about: a) the activities their students performed and/or participated in during the learning process; b) the usage of the learning content they had prepared and deployed in the online learning environment; c) the peculiarities of the interactions among members of the online learning community.

LOCO-Analyst is built on top of the LOCO ontological framework. It generates feedback out of learning object context data represented in accordance with the LOCO-Cite ontology. Furthermore, it exploits semantic annotation and domain ontologies to semantically relate diverse learning artifacts such as lessons, tests, messages exchanged during online interactions. When developing LOCO-Analyst we have used Semantic Web technologies as a toolset, but have put focus on the actual people involved in the learning process (learners and teachers) and the advantages they get from the technology.

The current implementation of LOCO-Analyst uses tracking data from the iHelp Courses LCMS. However, this does not mean that LOCO-Analyst is tightly coupled to iHelp Courses or any other specific e-learning system. Despite differences in the format of the tracked data provided by various learning environments, there are a lot of commonalities in their content and structure (e.g. the history of pages visited and the marks the students received on quizzes). These commonalities formally captured by the LOCO-Cite ontology make LOCO-Analyst a generic feedback generation tool that can be applied to diverse learning environments. The only thing that needs to be adjusted is the mapping between the tracked data format and the LOCO-Cite ontology.

LOCO-Analyst is implemented as an extension of the well-known open-source Reload Content Packaging Editor (http://www.reload.ac.uk/editor.html). This way we have ensured that instructors effectively use the same tool for creating courses, receiving and viewing automatically generated feedback about their use, and modifying the courses accordingly. This further ensures easier and wider acceptance of LOCO-Analyst. Figure 3 presents some screenshots of LOCO-Analyst.

We have already conducted an evaluation study of LOCO-Analyst and the results were generally very positive [12]. It is especially interesting to note that educators who have already used some other systems/tools capable of feedback provision, have recognize the benefits stemming from the use of Semantic Web technologies (such as qualitative over quantitative feedback, an integrated view on the learning process, etc.). Besides confirming the usefulness of our work, this study also helped us identify directions for further work.

2.2. MICE

MICE stands for Mixed-Initiative Coding Environment which aims at assisting student programmers to regulate their development style when programming in Java using an Integrated Development Environment (IDE). Development style for programming, among other aspects, refers to the ability of programmers to follow code conventions, to systematically debug code, to efficiently collaborate with other programmers, and to follow software engineering principles [13].
Unlike other programming environments, MICE (Figure 4) is designed to monitor and capture student programmers’ interactions with various tools and interpret these interactions in order to:

- enable student programmers reflect upon their programming styles as well as provide them with suggestions for improvements;
- provide feedback to teachers regarding students’ performance and programming styles, and thus help them identify the difficulties experienced by each individual student as well as enable them reflect upon their teaching strategies.

Currently, MICE uses BlueJ, an IDE specifically developed for the purpose of teaching object-oriented programming in Java to beginners. We have extended BlueJ with a plug-in that tracks programmers’ interactions with the IDE. This extension listens to BlueJ events such as compiling, clicking on a menu item and moving the mouse cursor. The events are captured in the MICE’s knowledge base as instances of appropriate classes of the LOCO ontological framework (see Section 1.2). In addition, events observed at external systems are also recorded in the knowledge base. For example, we are currently extending MICE with a module that interprets the interaction data captured in the iHelp Courses’ chat rooms and discussion forums. As we have already developed mappings between these iHelp Courses usage data and the learning object context data (defined by the LOCO framework), we are currently focused on interpreting the context data and generating relevant feedback for both student programmers and their instructors. We will also support extraction of some other iHelp Courses’ usage data, such as those collected from its code-sharing facility [11]. A similar plug-in for capturing events specific for the gStudy learning environment (e.g., text highlighting, browsing, and searching) is currently under development.
Changes in the MICE’s knowledge base trigger the execution of its rules. To achieve its objectives, MICE needs, among others, rules that map interaction events into the programming style components and hence enable automatic recognition of the user’s programming style. Another example of the rules that MICE requires are rules that identify situations when the user would benefit from the system’s advice – for example, when the user is spending too much time debugging a piece of code or getting too many errors and warnings regularly.

3. Related Work

For quite a while, a lot of research efforts has been focused on applying Semantic Web technologies to different aspects of e-learning. The Ontology for Education (O4E) Web portal (http://iiscs.wsu.edu/o4e/) is one of good starting points for those who want to learn about the state-of-the-art in this area. However, to the best of our knowledge, no one has tried to use these technologies to facilitate interpretation of actual usage (i.e., context) data. Hence, our work brings in novelty in this research area by demonstrating how Semantic Web technologies enable generation of feedback for both students and online educators, starting from the actual learning activities and interactions that took place in an online learning environment.

Zinn & Scheuer are developing Teacher Tool, a tool which analyzes and visualizes usage-tracking data in order to help teachers learn more about their students in distance learning environments. The development of the tool was preceded by a user study aimed at identifying the information that, on one hand is valuable for teachers, and on the other can be generated from user-tracking data. The results of the study [14] largely overlap with the results of a similar study that we conducted prior to developing LOCO-Analyst and confirms the usefulness of the types of feedback that our system provides. However, unlike Teacher Tool which is bounded to ActiveMath (http://www.activemath.org) and iClass (http://www.iclass.info) learning environments, our solution, thanks to its ontological foundation, is tool-independent.

Our work is also closely related to the work of Mazza and his associates [15] [16], who apply Information Visualization (IV) techniques to help teachers understand what
is happening in their classes. Their approach is first implemented in CourseViz, a tool which works with the WebCT LCMS (http://www.webct.com) to produce various graphical representations of student tracking data [15], and subsequently in GISMO, which does a similar thing for the Moodle LCMS (http://moodle.org) [16]. While these systems exploit IV to present raw data, LOCO-Analyst goes a step further in terms that it analyzes the data and provides teachers with qualitative feedback.

Kosba and his associates have developed the Teacher ADVisor (TADV) framework which uses LCMS tracking data to elicit student, group, and class models, and using these models help teachers gain better understanding of their distance students [17]. It uses a set of predefined conditions to recognize situations that require teachers’ intervention, and when such a condition is met, TADV generates an advice for the teacher, as well as a recommendation for what is to be sent to students. Whereas TADV is focused on the teachers’ day-to-day activities, LOCO-Analyst aims at helping them rethink the quality of the employed learning content and learning design.

Our work is closely related to the work on Massive User Modeling System (MUMS) [18]. MUMS is a framework aimed at supporting just-in-time production and delivery of user modeling information to interested parties. It is built using Semantic Web and web service technologies. In particular, RDF was used to express the semantics of the events being captured and transmitted between event producers and consumers (i.e., user modelers). However, there is no common model for representing user tracking data, and hence ontology mapping is required to enable communication of the parties involved. We address this problem by having the common core ontology (LOCO-Cite) that all involved parties understand.

The Shakya & Kumar’s work on the MI-EDNA system [19] is closely related to our work on MICE platform. Like MICE, MI-EDNA is grounded in the SRL theory and exploits its principles to help learners reflect on and regulate their learning processes. It also exploits an ontological framework for capturing learners’ interactions with the learning environment. However, it is focused exclusively on the kinds of interactions supported by the gStudy learning environment for which it is developed. It does not attempt at integrating data from various tools and systems, as MICE does.

4. Conclusion

The LOCO framework is the first solution in the area of e-learning that uses Semantic Web technologies to facilitate interpretation of actual usage (i.e., context) data. In particular, we have developed the LOCO-Cite ontology that defines the notion of learning object context by abstracting the most relevant concepts from a number of educational systems’ user tracking data. By employing this ontological framework, we have demonstrated how data about various educational activities can be integrated to provide meaningful educational feedback. This is done by developing the LOCO-Analyst teacher-centric system. The LOCO-Cite ontology is later extended for the purpose of capturing learners activities when studying programming languages. This ontology extension is exploited to build the MICE framework for regulation of programming styles when using an (educational) integrated development environment, such as BlueJ. The results of initial evaluation studies of both systems demonstrate the positive effects these systems have on improving the quality of educational process.

The main lesson learned is that the flexibility of ontologies allows us to build layers of abstraction based on the common foundations (i.e., the LOCO-Cite ontology),
and develop extensions that can (potentially) interact to achieve higher intelligence in the future applications. Following this conclusion, we plan to integrate the BlueJ extension into LOCO-Agent to support even more advance feedback for both students and teachers. Another important aspect of the future research is to explore how the LOCO framework can be used to better support learners in LCMSs. We also plan to research potentials of the framework for improving semantic social networking from the perspective of both learners (e.g., personalizing social-awareness of peers in a learning context) and teachers (e.g., stimulating social interaction) in educational systems.

References

Mobile Topic Maps for e-Learning

John MCDONALD and Darina DICHEVA
Winston-Salem State University
601 S. Martin Luther King, Jr. Drive
Winston-Salem, NC  27110

Abstract. Mobile and handheld devices have become an important area of computing technology with great promise for e-learning. As the information on the Web continues to grow at a steady rate, and mobile device capabilities increase, mobile web technology is poised to move to forefront of accessing and obtaining information from the Web. The use of ontology-aware learning environments, such as TM4L (Topic Maps for e-Learning), has shown that the delivery of ontology-based instructional content can provide a rich and efficient learning experience. This paper discusses the challenges of extending TM4L to a mobile computing platform, TM4L Mobile Edition (TM4L-ME) for use in handhelds, cell phones, and/or other mobile devices. We propose a novel mobile architecture for educational topic maps and discuss its implementation. The architecture uses Web Services and other Web technologies needed to provide mobile communication and functionality. It supports an extensible and reusable mobile environment for developing and using educational topic maps.

1. Introduction

Mobile and handheld devices have become an important area of computing technology. They are utilized and relied upon in educational, healthcare, public safety, and numerous other sectors of our society. As the information shared on the Web continues to grow at a steady rate, and mobile device capabilities increase, mobile Web technology is poised to move to forefront of accessing and obtaining information from the web. However, mobile web access today still suffers from interoperability and usability problems. To address these important issues, W3C launched recently its Mobile Web Initiative (W3C MWI) aimed at unifying and concerting efforts of key players in the mobile production chain, including authoring tool vendors, content providers, handset manufacturers, browser vendors and mobile operators. [1]

The primary focus of this project is extending the Topic Maps for e-Learning (TM4L) environment [2] to a mobile computing platform for use in handhelds, cell phones, and/or other mobile devices. We also examine web services and other web technologies needed to provide mobile communication and functionality, in hopes of providing an extensible and reusable mobile environment for developing and using educational topic maps. As part of our core goals we wanted to:

• Design Topic Map-based architecture for mobile device communication and learning content distribution.
• Develop a Mobile Topic Map prototype application capable of viewing and editing educational topic maps.

The idea is to allow a remote browsing and editing of educational topic maps residing on a server from PDA devices. For efficient communication of portions of a
topic map to a mobile device we introduce the concept of a *Topic Map Nugget*. The proposed extension of TM4L to a mobile platform is called TM4L-ME (Topic Maps for e-Learning - Mobile Edition).

This paper is organized as follows. In Section 1 we present an overview of the area of educational mobile computing. Section 2 discusses briefly the Topic Maps technology and the TM4L environment. Section 3 presents the main concepts, and Section 4 - the architecture of TM4L-ME. Section 5 concludes the paper.

2. Mobile Computing in Education

Mobile Learning (or m-Learning) is a new area of research with different interpretations of what constitutes m-Learning. Most of the interpretations focus on using devices that allow the learner to have a high degree of flexibility in location and transport. The Learning Citizen, for example, defines m-Learning as “…the use of mobile or wireless devices for learning on the move.” [3] Authors are split on determining which devices should be considered in m-Learning. Some include using mobile or portable devices, such as phones, PDAs, smart phones, digital audio players, Tablet PCs, Pocket PCs, etc. [4] Others constrain the use to only mobile devices, thus excluding portable devices. According to Lavoie [5] portable are devices, such as Tablet PCs and Laptops that “…can easily be brought anywhere but not accessed any time because of encumbering bulk and relative start up time.” In this line, Anderson and Blackwood [6] suggest that there are three major classes of mobile devices to be considered for use in m-Learning: (1) Personal Digital Assistants (PDAs); (2) Mobile (Cellular) Phones, and (3) Personal Media Players (PMPs).

In recent years we have seen a dramatic growth in the availability of mobile devices. There were 91 million mobile phone subscribers by the end of 2004, around 1.5 billion mobile phone subscribers by June of 2005, and over 3 billion subscribers predicted by the end of 2010 [7]. This is a penetration rate of nearly 43% of the total global population.

Electronic Learning, or e-Learning, presumes learning through digital media that can occur anywhere and at any time. It is typically associated with the availability of a PC connected to the Internet. M-Learning presents essentially the same concept, but with the addition of some capabilities and limitations. The difference between these two forms of electronic learning is more evolutionary than revolutionary [8].

According to Martinez and Garcia [4] the differences between e-Learning and m-Learning belong to three main categories:

- Nature of connection between students and teachers: e-Learning supposes that the user is permanently connected. However, this principle is not valid for m-Learning. Very few mobile devices are always connected and when they are, smaller bandwidth is usually available. Consecutively, there should be support for disconnected users.
- Device features: there is a big differences between the output (screen size, resolution, etc), the input (keyboard, voice, touch screen, etc.) and memory and processing power requirements.
- Mobility of users and devices: involves a new context data to be considered because of the possibility to offer context-aware (incl. location-aware) services.
These differences have to be taken into account when planning learning processes and activities within a mobile environment.

A number of studies (e.g. Corlett [9]; McAllister & Xie [10]; Miertschin & Willis [11]) explore the use of PDAs in the classroom for mobile learning. From elementary to high school, medical schools, etc. the use of specialized software applications has increased the use of mobile devices to reinforce or enhance domain specific learning. Most of these studies focused on the use of a specialized application program to teach domain-specific knowledge. Extending on these uses, Frydenberg [12] proposes the “learning spaces” approach for using a pocket PC as a platform for m-Learning about Information Technology. He describes various learning spaces that integrate the use of mobile devices both inside and outside of the classroom to introduce course topics. In a similar direction is the GEOBOOK project [13], which is aimed at the creation of reusable geo-referenced multimedia learning resources that are created and assembled on the field and then indexed and stored in repositories. During the on-the-field visits, learners can write notes, take photos, register observations and consult previous information about locations they are observing. Similar in nature is the HandLeR personal learning environment for children aged 7-11 [14]. The aim of a HandLeR is to provide the essential tools for daily learning, at school, in the home, or outdoors.

Similarly to these projects we are interested in supporting a learner to collect information about a topic and using his or her mobile device enter it in an ontology-based information pool that is collaboratively built by a group of learners.

3. Topic Maps for E-Learning

3.1. Ontology-aware E-Learning Systems

The need for e-learning systems based on domain ontologies is evident in the way we classify and categorize learning resources [15]. Educational resources are typically not structured in a standardized or widely accepted manner, but rather according to the preference of the author(s). This does not present a huge problem when we consider each resource individually, but gathering and collecting information from multiple resources presents a challenge.

If we wish to explore the relationship between two similar topics in different educational resources, such as two different textbooks on a particular subject, we must to some degree manually infer that the two topics are related and are of a particular type. We are forced to rely only on the resource index, if any, and/or the content description (Table of Contents) for each resource. The time and effort required in making manual (and sometimes incorrect) inferences could be better spent in the exploration of a repository with a pre-defined structure supported by an ontology-aware learning system. This system could create an environment of shared understanding and consistency through the use of domain ontologies [15]. By developing a system that can create, maintain, and use ontology-based repositories we can take advantage of highly structured and interoperable learning tool. TM4L (Topic Maps for e-Learning) [2], is an e-learning environment that encompasses all the features we have mentioned.

The TM4L environment is comprised of the TM4L editor and TM4L viewer. The TM4L editor has a user friendly GUI, which uses the underlying Topic Map engine, Topic Maps for Java (TM4J), an open source Java-based Topic Map API [16]. The TM4L editor allows the user to build learning repositories which are ontology-based
and utilize the Topic Map model to represent their structure. The power and benefit of this learning tool is that the learning content it produces is fully compliant with the XML Topic Maps (XTM) [17] standard and is interoperable with other XTM-conforming systems.

The main goal of the TM4L-ME (Topic Maps for e-Learning - Mobile Edition) project is to extend TM4L to a mobile computing platform for use in handhelds, cell phones, and/or other mobile devices. By combining the convenience and mobility of mobile devices with TM4L’s flexible delivery of learning content, we can create an extended and exciting learning and authoring experience for the user.

In order to describe the intended functionality of the proposed application we first introduce briefly the Semantic Web technology we use, Topic Maps.

### 3.2. Topic Maps

Topic Maps (TM) are an ISO standard [17] that can be viewed as an interchangeable hypertext navigation meta-layer above diverse electronic information sources supporting topical finding of various kinds of resources, such as documents, graphics, images, database records, audio/video clips, etc. This meta-layer models all the topics - persons, objects, concepts, thoughts, etc, which are described “in” the resources and the relations between the topics. The topics and the resources are connected by hyperlinks (occurrence links).

The main Topic Map components are topics, associations, and occurrences. Using those elements, one can create maps in document repositories. The topics represent the subjects, i.e. the things, which are in the application domain, and make them machine processable. They can have zero or more topic types and also have names (a base name and variants for use in specific processing contexts). A topic association represents a relationship between topics. Associations can have types (e.g. ‘illustrated by’, ‘example of’, ‘written in’ etc.) and roles (e.g. ‘example’ and ‘concept description’; ‘prerequisite’ and ‘result’; ‘document’ and ‘language’). Occurrences instantiate topics to one or more relevant information resources. An occurrence can be anything; most often it is an URI or a document (article, picture, video, etc.). Scopes define the extent of validity of a topic characteristic assignment: the context in which a name or an occurrence is assigned to a given topic, and the context in which topics are related through associations. An important concept in TM is this of identity. Two topics are the same if both have the same name in the same scope or both refer to the same subject indicator. The topics and all their characteristics could be merged if this condition holds.

### 4. The TM4L-ME Project

#### 4.1. TM4L-ME Goals and Architecture

Our goal was to preserve the TM4L functionality, which implied the following:

- Editing the e-learning repository
  - Adding a new topic
  - Editing an existing topic
  - Deleting a topic
  - Inserting a relationship between topics
Browsing the e-learning repository.

We investigated different possibilities for the intended mobile Topic Map-based architecture. The idea of converting the full version of the standalone TM4L to a J2ME (Java Mobile Edition) application wasn’t feasible since such an implementation presents a difficult challenge and requires an extended focus on mobile device memory manipulation and development in order to provide persistent storage area for viewing a topic map. An alternative was to develop a server-side topic map engine and storage and a thin mobile client that could only ‘ask’ the server for the information about a particular topic. This strategy has been used in the implementation of a topic map query tool for mobile devices, described in [18]. However, we were not satisfied with such a solution. Our goal required not only to query, but also to update a topic map from the mobile device. Thus we selected client-server architecture with a thick mobile client, where the mobile device can host a small ‘chunk’ of the topic map, which we call a Topic Map Nugget (TmNugget) (see Fig.1). The central point for managing the delivery of topic map functionality to connecting mobile devices is the TM4L Mobile Server. We introduce the concept of TmNugget to meet the challenge for efficient communication of segments of a topic map between a server and mobile devices. The TM browsing can then be seen as a step-by-step process where at each step a TmNugget is transferred from the TM4L Mobile Server to the PDA and browsed there. When the topic of interest is found, the user can ask for topic information: topic name(s), topic type, and/or topic resources.

Another important design decision was the development of a lightweight and portable version of TM4L, which provides the same intended functionality, but reduces development time by taking advantage of the existing TM4L implementation through the utilization of web services.

![Figure 1. TM4L-ME Architecture](image)

The concept of TmNugget supports the following design requirements:
- Extracting a TmNugget containing a specified topic from a topic map.
- Merging a TmNugget to a topic map.
- Sending a TmNugget from the mobile server to a mobile client.
- Sending a TmNugget from a mobile client to the mobile server.
4.2. Topic Map Nuggets

A TM nugget has (is defined around) a center topic. In its simplest case, a TM nugget consists of the center topic only; next it can consists of the center topic along with all directly connected to it topics and the corresponding associations in the topic map. In a typical case, TM nuggets will be defined by extracting a bigger ‘chunk’ of the topic map, for example, such that contains not only the center topic together with its direct neighbours, but also those topics that are indirectly connected to it with, for example, a path of length 2, 3, 3, etc.. The actual ‘depth’ will depend on the available memory of the used mobile device.

More precisely, a TmNugget for a specified topic \( \nu \) and dept \( n \) (\( n = 1, 2, 3, \ldots \)) is a sub-graph of the topic map graph, having a central point \( \nu \) and eccentricity \( \epsilon(\nu) = n \). A TmNugget is represented in an XML format. For that purpose the XTM standard is used, extended with an additional element: \texttt{tmNugget} (see Fig. 2).

Breaking the topic maps into nuggets allows efficient memory utilization on the mobile device. Once adding a new TmNugget or editing a downloaded one is completed, the nugget can be uploaded to the TM4L-ME server. The server will determine the best way to insert or re-insert the changes back into the original topic map.

![Figure 2. TmNuggets](image)

TmNugget extraction involves finding a sub-graph of the topic map graph with a specified center topic and depth. TM4L utilities are used for selecting the relevant pieces of the main topic, which are then serialized in XTM format. Inserting or re-inserting a TmNugget back into the topic map is implemented by applying the standard topic map merge operation with appropriately set parameters.

Creating TmNuggets is delegated to the TM4L Server. Allowing the server to handle the TmNugget generation reduces the performance load on the mobile devices and simplifies deployment as the TmNugget is delivered via a web service. The process of creating a TmNugget begins with a request from the mobile device to the Topic Map Server. This initial request contains the device specifications, and the topic map id being requested. The Topic Map Server acknowledges the device request and compares it specifications to a known list of mobile devices and/or determines the memory and storage constraints from the initial request containing the device specifications.
Server also creates a unique session id which allows the device to make subsequent requests without having to re-identify itself.

To create a TmNugget, TM4L Server first queries the topic map repository to find the requested topic map. Once located, TM4L Server links it to the generated session id and begins the task of creating the nugget. The TM4L Server uses the requesting device specifications to determine the size (depth) of the TmNugget (if not specified).

5. TM4L-ME Implementation

TM4L-ME is built as a service-based application. We utilized web services based on the SOAP [19] standard. The TM4L-ME solution is comprised of a TM4L-ME Server (TMS) (web service) and TM4L-ME mobile client. The TM4L-ME Server leverages the backend functionality of TM4L allowing the user to create, edit, maintain and view topic maps. The TM4L-ME mobile client enables the device to communicate with the server. Both the server and the mobile client are Java applications. We use the Java Micro-Edition 2 (J2ME) platform [20] as the development platform along with the Eclipse IDE. This allows integration with current topic map development projects and toolsets already in use.

5.1. TM4L-ME Server

Web services provide a vehicle for applications to communicate over the web using standardized protocols (such as SOAP). Web services are self-contained, modular applications that can be described, published, located, and invoked over a network, generally, the web. The architectural advantages of using web services allow developers to deliver application APIs in whole or part, based on the need and security constraints. Specifically, for an existing application to start publishing services to the web; there is no need for a significant redesign – a certain adaptation will be only needed, which preserves its functionality and not compromises its security requirements. Much like Object-Oriented development, some of the fundamental concepts in web services development are encapsulation, message passing, dynamic binding, and service description and querying. The fact that we are delivering our API over the web gives another advantage, platform independence. By utilizing web services we ensure that our application is truly interoperable and scalable through enabling communication with any SOAP compliant client or external web service.

The proposed system architecture for TM4L-ME is presented in Fig. 1. The TM4L-ME application operates in two modes, Connected and Disconnected. When connected to an available network, whether an internal intranet or an external internet connection, mobile TM4L will have access to the full spectrum of tools and classes of the TM4L framework, and the ability to download and upload any portion of an existing topic map from/to the topic map repository located on the server. Thus the Mobile TM4L will be able to view, edit or create portions of the topic map and hand-off to the TM4L Mobile server for processing and placement in the TM repository.

5.2. TM4L-ME Mobile Client

We chose the Blackberry 7250 as our primary development platform. The popularity of the Blackberry presents us with a platform that is well known and widely accepted. The
Blackberry device also has several features that make it appropriate for viewing and authoring topic maps, such as a keyboard and scroll mouse.

The TM4L-ME Mobile Client is a J2ME application built using the Connected Limited Device Configuration (JSR 30, JSR 139) [21]. The CLDC provides a pre-defined set of APIs for memory constrained devices, such as cell phones and PDAs. By building our mobile client according to CLDC, we increase the chances that the TM4L-ME Mobile client will function on a wide range of devices, as the CLDC is an industry standard. Conforming to CLDC also ensures that our TM4L-ME client is developed with an open and established architecture that allows for future improvements. The Eclipse development environment that we use offers many useful tools, including simulators for testing the client code. This simulated environment offers the experience of mobile connectivity and functionality from the desktop development platform.

The Web Service portion of the TM4L-ME Server is implemented using Apache Axis, a framework for building and publishing Java web services. Apache Axis is used in conjunction with Apache Tomcat to host web services. Once deployed to the Axis/Tomcat server the web service can be accessed from the web.

We should also note that while our application will run on any Java ME compatible device, the BlackBerry device requires an additional conversion of the mobile application to run on the BlackBerry platform. The tools for converting the Java ME application are included in the Blackberry development toolkit. TM4L-ME implementation is a work in progress.

6. Conclusions

As mobile computing continues to grow, the educational uses escalate as well. The ability to deliver learning content to mobile devices such as PDAs, constitute a hands-on learning tool that is not limited to classroom. TM4L-ME can be used in various ways in educational setting. For example, students can use it to browse course-related learning materials prepared by their instructors in an efficient and highly structured manner. Among the benefits would be that the information distributed would already be categorized semantically, possibly allowing a more intuitive and knowledge rich learning experience. Another possible application of TM4L-ME is for students to collaboratively build a subject-centred knowledge repository in a specific domain. An important future research project will be to evaluate the benefit of using TM4L-ME in a real educational setting.

Some possible enhancements of TM4L-ME include:

- Improved TM4L-ME mobile client viewer for mobile devices that focuses more on the concepts and features of topic maps for times when the device is not connected to a wireless network. This will be possible as the amount memory in the PDA devices continues to grow.

- Improvements in the TmNugget delivery process. The problem of how to break-up and re-build large topic maps for faster performance can be tackled in different ways.
We believe that the use of topic maps in a mobile platform provides many exciting possibilities in the area of e-learning. The process can be both challenging and time-consuming, but we trust that a long-term investment in mobile e-learning tools, such as TM4L-ME, will improve the delivery of educational content.

References

Scenarios for a Learning GRID

Agathe MERCERON a, 1, Nicola CAPUANO b, c, Francesco ORCIUOLI c
and Pierluigi RITROVATO b, c

a University of Applied Sciences TFH Berlin, Germany
b CRMPA, Centro di Ricerca in Matematica Pura ed Applicata, Italy
c DIIMA, University of Salerno, Italy

Abstract. Learning GRID is both a concept and a Special Interest Group of the European Network of Excellence Kaleidoscope. Kaleidoscope is a Network of Excellence funded by the European Commission which brings together European research teams in Technology Enhanced Learning (TEL). The Learning GRID Special Interest Group gathers researchers who want to contribute to an improvement of e-learning practices through the use of the Learning GRID concept. Its key idea reads as Learning Grid technology allows for a direct and personalized experience in realistic contexts and boosts creation of virtual dynamic communities. After introducing briefly Kaleidoscope, this article presents the learning GRID concept and several scenarios that this technology makes possible.

Introduction

Learning GRID is both a concept and a Special Interest Group (SIG) of the European Network of Excellence Kaleidoscope. Kaleidoscope is a Network of Excellence funded by the European Commission which brings together European research teams in Technology Enhanced Learning (TEL). The key challenge that Kaleidoscope is facing is the scientific and structural integration of European TEL research. The challenge lies in the topic itself which is multi-disciplinary. Therefore, Kaleidoscope gathers researchers coming from various disciplines, from learning sciences to computational technology, which is critical since knowledge and meaning making are key to learning. The challenge lies also in the European dimension of the network, since European countries have different educational systems.

The aim of the Learning GRID SIG is to contribute to the achievement of an improvement in e-learning and training practices through the definition of open, distributed and pervasive environments for effective human learning. This view takes into account that (i) effective learning requires an active attitude of learners (ii) and that learning is a social activity. Therefore, future learning scenarios require a technology that allows for active and realistic experiments, personalization, knowledge creation and evolution, as well as autonomous and dynamic creation of communities.

In this contribution we first present what is meant by the Learning GRID concept, then we present pedagogical scenarios that such a concept allows for.
1. Learning GRID Concept

The Learning GRID concept, as proposed by the SIG just mentioned, integrates Grid technology, semantics and learning design.

What is Grid? “Grid is a service for sharing computing power and data storage capacity over the Internet”. Connecting heterogeneous computers necessitates a middleware called a Grid architecture. This middleware comes between the operating systems of the single computers and the application that runs on all the connected computers. A Web Service is a software component that makes itself available over the Internet and that uses a well-defined standard for messages. Examples of popular services are weather reports, plane real time schedules. Grid architectures and Web services mechanisms are merged in the Open Grid Service Architecture (OGSA). This technology provides a transparent access to distributed (stateful) services and resources, provides interoperability between heterogeneous environments, security, trust and ubiquity. With this technology, students’ machines can offer their storage and computing facilities to run parallel algorithms taught in a parallel computing course for example.

What is Semantics? Semantics has to be understood in the Semantic Web context. The Semantic Web is the vision that data on the web are defined and linked in a way that they can be used by machines not just for display purposes but also for integration and reuse across various applications. Nowadays, a Google search with the keywords ‘algebra exercises Grade 8’ gives Internet pages that do propose algebra exercises for grade 8 students, but also Internet pages on the academic content standards for Grade 8 algebra. Search engines know only that such pages contain the pattern or syntax ‘algebra exercises grade 8’ but have no clue on the different meanings or semantics that such pages give to these words. This semantics have to be made explicit by an indexation or annotation of the pages using a well defined vocabulary or ontology. Semantics makes it possible to provide services typical of the learning domain and thus allows for resources discovery, service composition and personalisation. With this technology, it would be easier for a teacher to tailor an algebra course integrating many exercises for practically oriented learners.

What is Learning Design? Learning Design [1] is a language to express many different pedagogies. Learning Design allows for creating scenarios able to capture all the identified pedagogical features. With Learning Design, you can construct a pedagogical scenario for a calculus course with activities on Limit first, then on Derivatives and finally on Integrals. Activities can be individual or collaborative.

2. Scenarios

Various scenarios have been proposed inside the SIG for the Learning GRID. A complete description can be found in [2] and [3] and are summarized in the sequel of this section. Scenarios presented in [2] are either already implemented or are extensions of existing teaching and learning practices. Scenarios presented in [3] are more futuristic. All technologies to implement them are available. The challenge is to make these technologies work together.

Parallel Computing Cooperative Learning. In this scenario, the classical clusters of computers for teaching parallel computing is replaced by a GRID. Grid technology makes it easier to connect students and instructors’ machines to a dedicated cluster, and
to make several institutions collaborate. Such a multi-structure encourages a broad learning experience which includes issues from programming considering communication details to use the networked resources transparently and seamlessly as one supercomputer. Definition of new services would allow teachers to assign tasks to groups of students, to monitor all connected peers, visualize the work of students, access learners’ documents, code and results. It would also allow students to share their code and solutions. Such a scenario encourages simultaneous participation of students because computational power is increased as more students join.

A Collaborative Research Environment. This scenario takes its inspiration from Citeseer, a digital library which organizes scientific literature, allowing flexible searching and citation statistics. The use of GRID technology makes it possible to extend the digital library with new forms of collaborative activities. Apart from papers more resources can be shared such as formal or informal comments, reviews, summaries, videos, code. Virtual organisations can be defined. A virtual organisation can be specified as research students and their supervisor(s), a research group, or a whole lab. Services can be personalized according to users. Access rights for students or supervisors may not be the same. Such a scenario constitutes a new and useful way of supporting research activities, allowing to easily organize and access all sort of documentation/information, and to assess individual and group progress.

A Distance Programming Course based on Practice. This scenario is based on an existing e-learning programming course running in the Engineering School Léonard de Vinci [4]. In this course, students are learning by doing more than by reading learning material. They are provided with many programming exercises that are automatically corrected. Moreover, all their attempts are stored so that students can consult their history. Compilation and program execution are implemented as services, students do not need to worry about downloading themselves any programming environment. Learning GRID would improve the existing system in two main ways. First Grid technology, by allowing a distributed code compilation and execution as well as a distributed storage of students’ answers would make the present system scalable. Second semantics would allow for better personalization and flexible course composition.

Collaborative lab. The traditional lab in computer science is a collection of isolated individuals each pulling their own material and exercises from a dump pedagogical server, and deploying activities invisible for the tutors. Usually informal mails and forums are the only groupware metaphors used by students to break this isolation. The Learning GRID could lead to richer distributed system metaphors enabling teachers to perform actions such as visualizing the activity of students, actively gathering results of students, navigating through these results and making them visible to selected groups.

Using GRID Computing for Processing and Analysing Information from on-line Collaborative Learning Teams. This scenario is implemented and working at the Open University of Catalonia [5]. Computer supported collaborative learning (CSCL) applications are characterized by a high degree of user-user and user-system interaction and hence generate a huge amount of data usually collected in log files. In order to extract information useful to the group activity, these files must be processed in real time. Grid technology is used to process group activity log files and, thus, makes it possible to provide timely feedback to users and compute different statistics.

Networking Course by E-learning. The networking e-learning course teaches basic concepts and allows students to design and configure simple networks. On a trip train,
a student can access the course from her PDA and perform network simulations. The output of the simulator is automatically adapted to the device and gives numerical results to a PDA and graphical results to a laptop for example. Using the public WLAN infrastructure available in the train, the student can contact a tutor and asks questions, or can chat with other students, share documents and perform collaborative activities. This scenario allows for ubiquitous, active and social learning. GRID technology makes communication between heterogeneous devices and use of simulators possible. Simulations could be completed/replaced by more ambitious virtual lab experiments.

Featuring immersive Virtual Reality. A student is learning water and aquifer behavior by using an e-Learning platform. She can access introductory books to gain initial knowledge on the topic. She can also use a dynamically generated set of services that, according to her user profile and current PC capabilities create an immersive virtual reality. The service also informs the student about immersion requiring special equipment like gloves and glasses and about the nearest locations where such immersions are possible. Such immersions allow her to go deeper on the aquifer behavior and characteristics of water. In these virtual sites, she also meets other students and interact with them. Learning occurs naturally as a result of experiments and interactions with other students. GRID technology combined with user profile is needed to make this scenario work.

Featuring the virtual laboratory. Students can access a virtual remote chemistry laboratory using different terminals (PDA, laptop, PC etc.). Students are identified by the system and a status of their work is stored at their user profile. Such information makes it possible to break and resume experiments that may require distributed simulation or aggregation of distributed information. Supercomputer facilities allow specifying and testing complex analysis, typically performed by groups of three students. Output of experiments may be numerical or graphical and adapted dynamically to the devices used by students. From time to time a synchronization period allows for comparing results between groups and getting partial marks. As above, this scenario uses user profile and GRID technologies.

The Field Trip [6] (see Figure 1). Students, equipped with PDAs, go on a field trip. While performing their activities, they store information under the form of photos, videos, text notes, audio comments, etc. Students can also access digital libraries and documents necessary for their work. The PDA, using user profile and context dependent information, automatically indexes these contents using appropriate meta data. All the contents stored by all students are sent, via an appropriate network to the Field Trip (FT) GRID service created before hand by the teacher. The PDA uses biometric data for secure access and for data ciphering. The FT GRID service uses external services like high performance computing services to provide 3D reconstructions of photos, or to orchestrate speech to text and for text interpretation. Using an ontology based knowledge representation, the FT GRID service can compare students’ work with learning objectives and prepare a summary for the teacher in terms of progress and weakness. Using semantic based service searching and location capabilities, the FT GRID service makes it possible for students to contact other students with similar interests or other students who are geographically near. This scenario uses the typical GRID approach in combination with mobility. Semantics is required for knowledge based services.

English plan. Students with a similar English level are sent to an immersion program that consists of a 1-week-trip to New-York. Each day they have to send a report about the activity planned by the tutor that they have performed. Let us take the
‘visit to the stock-exchange’ activity as an example. The night before the visit, students prepare themselves using their PDAs and the hotel WI-FI network to consult documentation, contextual vocabularies, access on-line dictionaries as well as a pronunciation service. They can also perform collaborative work. After sending all reports, a student can access the evaluation virtual network, authenticate himself and take an official exam. This scenario uses primarily mobile GRID technology.

*How to elaborate a business plan.* While flying to a meeting to Japan, a business man has to complete parts of a business plan. As these parts are quite tricky, while at the airport he downloads on his PDA a course about business plan using the WI-FI network of the airport. During the flight, he seeks further advice by connecting to a GRID service that allows him to locate a tutor and discuss with him via a video-conference service. He also accesses a profitability simulator service to complete his work. As above, mobile GRID technology is essential for this scenario.

![Figure 1. The Field Trip Scenario](image)

**Acknowledgements**

This work has been partly supported by the European Network of Excellence Kaleidoscope.

**References**


Supporting Philosophers’ Work through the Semantic Web: Ontological Issues

Michele Pasin, Enrico Motta
Knowledge Media Institute, Open University
m.pasin, e.motta @ {open.ac.uk}

Abstract. As the Semantic Web is increasingly becoming a reality, the availability of large quantities of structured data brings forward new challenges. In fact, when the content of resources is indexed, not just their status as a text document, an image or a video, it becomes important to have solid semantic models which avoid as much as possible the generation of ambiguities with relation to the resources' meaning. Within an educational context, we believe that only thanks to these models it is possible to organize and present resources in a dynamic and contextual manner. This can be achieved through a process of narrative pathway generation, that is, the active linking of resources into a learning path that contextualizes them with respect to one another. We are experimenting this approach in the PhiloSURFical tool, aimed at supporting philosophy students in understanding a text, by presenting them “maps” of relevant learning resources. An ontology describing the multiple aspects of the philosophical world plays a central role in this system. In this paper, we want to discuss some lessons-learned during the modelling process, which have been crystallized into a series of reusable patterns. We present three of these patterns, showing how they can support different context-based reasoning tasks and allow a formal conceptualization of ambiguities that are primarily philosophy-related but can be easily found in other domains too.

1. Introduction

The need to specify and separate the information about the context of usage of a learning resource, from the resource itself, is one of the main reasons behind the creation of various kinds of metadata schemas. In the past years, this work has focused around the notion of learning object (LO) [1], as the technology capable of guaranteeing interoperability to the rapidly growing number of Web-based educational applications. However, increasingly researchers are now arguing that LOs’ metadata are not fine-grained enough to non-trivial composition of resources, e.g. when constructing a curriculum [2]. As a result, as attested by a series of workshops held worldwide [3], the e-Learning research community has begun looking at the potential for e-Learning of the emerging Semantic Web technologies. Our approach, in compliance with this new research direction, addresses the standard LOs’ metadata limitation through the usage of ontologies in order to represent the knowledge used in the metadata.
In the PhiloSURFical tool\(^1\) this approach is realized through the formalization of a humanistic discipline, philosophy. An ontology to describe and organize theories, schools of thought, arguments, problems and their relations to other philosophical concepts will allow the annotation of the learning material, and, subsequently, its dynamic reorganization with a degree of accuracy and flexibility well beyond the one provided by standard LOs metadata. The aim of this paper is to present some ontological lessons-learned which emerged as fundamental in supporting these functionalities. Under this respect, the modelling patterns we are presenting are quite different from the patterns discussed in other works such as [4], where the focus is more on the architectural issues involved in the ontology creation process. In particular, the patterns we are describing in the next sections represent some modelling decisions that are meant to guide the interpretation of philosophical texts, so to have formal models that are applicable for providing non-trivial navigation mechanisms. We believe that such a modelling can have a significance that goes beyond the specific domain of philosophy and can be reusable within more generic areas of interest.

In the following sections, we will first describe the pedagogical perspective motivating our work (section 2.1) and the functionalities of the PhiloSURFical tool used to test bed our assumptions (section 2.2). In the second part of the paper, we will instead give an overview of the approach taken in constructing the ontology (section 3.1) and present three important modelling patterns which emerged during the work: one about a natural language ambiguity regarding events, schools of thought and actors (section 3.2), one regarding the multiple view-types we identified in the domain (section 3.3) and one regarding the subtleties involving the representation of problem areas (section 3.4). Finally, in the conclusion we sum up the scope and contribution of this work.

2. Learning through discovery of relevant resources

2.1. Constructive Learning in Philosophy

It is nowadays common-sense knowledge that for a student to really understand something, it is necessary an active style of learning, not just, for example, a passive reading and remembering of what is read. In educational theory, this thesis (and others related to it) is one of the central tenets of doctrines such as constructivism and situated cognition [5]. Their importance and academic relevance, beyond the various and inevitable debates, is widely acknowledged. According to this position, students are usually advised to engage directly with a subject matter (e.g. an author's text), in order to obtain their own understanding and actively "construct" a meaning out of it.

However, this picture is quite a simplified one. While an active style of learning is relatively easy to foster in “natural”, everyday situations (for example, when learning how to ride a bike or how to speak a language), this is not the case for the more artificial, “academic” learning. The learning and teaching of philosophy, for instance, is a very delicate matter; philosophy, as other subjects such as theoretical physics, mathematics and logic, deals only with abstractions, that is, in Laudrillard terms, “descriptions of the world” [6]. As a consequence of this, its learning cannot be situated in a natural context, but it’s intrinsically de-situated and linked to other

\(^{1}\) A prototype of the tool is available at http://philosurfical.open.ac.uk
processes of second-order nature. This de-contextualized learning is also necessarily mediated: philosophy students do not engage directly with the world, but they act on descriptions of the world. Precisely, with the descriptions emerging from the teacher’s world. In such an academic and abstract context, what are the ideal students’ activities, to the end of a successful learning experience, and what are the best methods and situations to support them, is undoubtedly the object of much debate [7].

But even if a general agreement on this matter will hardly be reached, we can still define some requirements to achieve: for example, in the same specific case of philosophy, following a recent discussion on the topic we agree that the three most important skills to develop in a student must be (a) analysis, (b) argument and (c) interpretation [8]. As the author remarks, the “three skills are interwoven as analysis requires interpretation, and argument depends on the prior abilities to analyse and interpret correctly other philosophical positions”.

With the PhiloSURFical tool we aim especially at supporting the (a) analysis and (c) interpretation skills development, through an environment which allow constructing advanced strategies to present annotated resources to the user, in the form of browsing facilities and narrative generation (figure 1). The active involvement of the student in a process of semi-structured navigation (the structure being provided by the ontology) guarantees her engagement with the subject matter in a “constructivist” manner.

![Figure 1. Screenshot of the PhiloSURFical application](image)

### 2.2. The PhiloSURFical tool

The PhiloSURFical’s tool functionalities, and in general, the envisaged context of usage which has been guiding the ontology engineering process is the following: the semantic model should support the reconstruction of the history of ideas, by relying on structured information about the practical domain and the theoretical domain of thinkers. Our approach takes the notion of a learning pathway as a “system of specially stored and organized narrative elements which the computer retrieves and assembles according to some expressed form of narration” [9] and attempts to transpose it within the specific scenario made up of philosophical entities.
For example, within an educational scenario where young philosophers try to understand domain notions (in a wide sense, comprising ideas and events), these functionalities will exist in the form of mechanisms for contextual navigation and linking of relevant resources. As a result, we expect such a service to facilitate the discovery of (related) unknown resources, which can be used by students and scholars during the process of answering difficult problems.

The PhiloSURFical application is being prototyped with Wittgenstein’s Tractatus Logico Philosophicus [10] and it allows the navigation of a semantically enhanced version of the text. By relying on an ontology created to describe the philosophical domain at various levels of abstraction, users can benefit from multiple perspectives on the text and on related resources. For example, they can reorganize the same text according to the relevance of a single metadata, e.g. the concept of “logical-independence” (tab 2 – metadata based sequence); they can query the knowledge base or other repositories in the Semantic Web, such as the DBpedia [11], by choosing an object of interest (i.e. a topic) and using it to trigger a theoretical narrative (i.e. meta-historical, tab 3), a historical narrative (tab 4), or a geographical one (tab 5).

This is achieved by using simultaneously the knowledge encoded in the ontology, an initial knowledge base of resources and metadata built by a philosophy teacher, and the SPARQL [12] query language to gather information from other sources in the SW.

However, the perspectives mentioned above are only grouping a larger set of possible “narratives” that can be derived from an appropriate metadata description of the domain. The division into theoretical, historical and geographical seemed to us a valid initial breakdown of the domain’s features, which also generic users’ would easily grasp. In the table below, for example, we show other generic narratives existing within the meta-historical narrative (table 1).

<table>
<thead>
<tr>
<th>Theoretical Narrative</th>
<th>Ontology classes mainly involved</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disambiguation</td>
<td>Concept</td>
<td>Visualization of the concept of “god”, as defined by different competing conceptions</td>
</tr>
<tr>
<td>Contrast</td>
<td>View</td>
<td>Visualization of competing theories, the problems involved and the supporting arguments</td>
</tr>
<tr>
<td>Resolution</td>
<td>Problem, Argument</td>
<td>Visualization of the path from the birth of a problem, and its solution</td>
</tr>
<tr>
<td>Analogy</td>
<td>Concept, View</td>
<td>Visualization of concepts that play similar roles in different theories</td>
</tr>
<tr>
<td>Causation</td>
<td>View, Problem</td>
<td>Visualization of a chain of problem-theory-problem etc.</td>
</tr>
</tbody>
</table>

Table 1. Theoretical narratives

As the complete formalization of these “narratives” in the form of SW-queries is still an ongoing work, and the PhiloSURFical prototype has not been evaluated yet, we will not discuss more these aspects, but instead focus on the analysis of some aspects of the ontology. Remarkably, we soon realized how the latter constitutes the primary source for the creation of interesting and valuable narratives. In fact, as we will see in the next section, the philosophical domain conceals various ambiguities and implicit
assumptions that have to be made explicit and formalized. Only by doing so, Semantic Web agents can perform reasoning tasks that offer a valuable alternative to more classic types of search.

3. Engineering Philosophical Knowledge

3.1. Overview: an event-centred design

The specific approach used to realize this system has at its centre the decision to employ the CIDOC Conceptual Reference Model [13] as a starting point for our formalizations. The CRM ontology was originally an attempt of the CIDOC Committee of the International Council of Museums (ICOM) to achieve semantic interoperability for museum data. Since 1996, the formal model has improved considerably till becoming in 2006 an ISO standard. It is now (version 4.2 [14]) in a very stable form, and contains 75 classes and 108 properties, both arranged in multiple is-A hierarchies. The choice of using the CRM was motivated by two reasons. Firstly, for its widely recognized status as a standard for interpreting cultural heritage data. In fact, by reusing and extending an existing and internationally recognized ontology, we will give our tool's users more chances to benefit from the emerging Semantic Web infrastructure.

Secondly, for its extensive event-centered design. This design rationale, in fact, appeared to be appropriate also when trying to organize the history of philosophy: even if it is common to see it as an history of ideas, stressing the importance of the theoretical (i.e. meta-historical) dimension, we believe this cannot be examined without an adequate consideration of the historical dimension. That is, a history of the events related (directly or indirectly) to these ideas. In figure 2 we can see an example of an event-centered representation in the PhiloSURFical ontology. The persistent-item

Figure 2. Reasoning with philosophical events
class, which is one of the five classes composing CIDOC’s top layer (together with
time-specification, dimension, place and temporal-entity) subsumes thing and actor.
The two branches of the ontology departing from them can have various instances,
which are related by taking part (in various ways) to the same event. This kind of
modelling, in the context of the PhiloSURFical tool, is extremely useful because of the
multiple navigational pathways it can support (e.g. we could move to another event
having the same topic, or to another topic treated during the same event, etc.)

From the implementation point of view, the ontology has been prototyped by using
the Operational Conceptual Modelling Language (OCML) [15], which provides rich
support for both specification and reference. Import/export mechanisms from OCML to
other languages, such as OWL and Ontolingua, ensure symbol-level interoperability.

At the time of writing, the ontology counts 348 classes, partly integrated from
other relevant semantic models and partly identified through various knowledge
acquisition techniques (formal and informal). In the next sections we will present three
ontological issues we encountered during the modelling process, together with the
solutions we contrived in order to solve them. As in the example above, the derived
modelling patterns aim at taking advantage of the multiple meanings a philosophical
entity (e.g. an idea, a text or an event) can have, by making these meanings explicit and
employable when building novel exploration mechanisms. In other words, according to
our approach “ambiguities are good” because, if properly identified, they let us explore
the domain in different and interesting ways.

3.2. Pattern #1: is rationalism a school of thought or an event?

The first pattern origins from the fact that in our everyday language we refer to belief
groups, intellectual movements and schools of thought ambiguously, often using the
same word.

According to CIDOC, Period (which is a direct subclass of Temporal-entity)
should subsume prehistoric or historic periods, or even artistic styles. This is motivated
by the fact that "it is the social or physical coherence of these phenomena that identify
a Period and not the associated spatio-temporal bounds" [14]. This seemed to apply
quite neatly also to cultural and philosophical periods, thus we have added Intellectual-
movement and its subclass Philosophical-movement to the hierarchy. So, for example,
we can describe the “enlightenment movement” in the following way (note that the
temporal relations are specified here as slots, but are usually inferred whenever the
appropriate time specifications of the other periods were provided)\(^3\):

\(\text{(def-instance enlightenment intellectual-movement}
\hspace{1cm}\text{((has-time-specification 18th-century)}
\hspace{1cm}\text{(overlaps-in-time-with scientific-revolution renaissance)}
\hspace{1cm}\text{(meets-in-time-with French-revolution American-revolution romanticism)}
\hspace{1cm}\text{(overlaps-with Age-of-Reason neo-classical-art)}
\hspace{1cm}\text{(took-place-at Germany France Britain Spain)}
\hspace{1cm}\text{(has-related-group-of-people enlightenment-group-of-people)}
\hspace{1cm}\text{(is-typified-by enlightenment-conception)))}\)

The last two properties in the previous example have a special importance. Periods, in fact, appear to be tightly connected to the abstract ideas defining them and

\(\text{\(^3\)The latest version of the ontology can be found online at http://philosurfacal.open.ac.uk/ontology/}. \text{\(^1\) OCML has a simple frame-like syntax, with sequences of slot-name and slot-values.}\)
to the group of people that often carries the same name. The slots has-related-group-of-
people and is-typified-by specifically serve this purpose. This issue is better understood
is we just consider how often this feature of intellectual events generates ambiguities,
since in natural-language expressions it is not clear what entity we are referring to. For
example, let us consider the following three statements:

“Throughout history, the attacks of rationalism against empiricism has diminished”
“Descartes was one of the founders of modern rationalism
“This theory is clearly a new and re-shaped rationalism”

At a first examination, all three sentences refer to “rationalism”. However, at a
deeper ontological analysis, we came to the conclusion that in a) “rationalism” is the
label referencing to a group of people, in b) we are meaning an event, while in c) we
are probably referring to an abstract idea.

A modelling pattern (figure 3) involving actors, periods and views (a type of
abstract philosophical idea, as we shall see later, expressing a standpoint) attempts to
tidy things up. The ambiguity of a term such as "rationalism" can be clarified, since the
semantic model keeps the three different ways to intend the concept into a consistent
representation. By doing so, we are providing a context of usage for such ambiguous
concepts, and a direct way to navigate coherently among entities that are ontologically
quite distinct (i.e. from temporal-entity to actor and propositional-content, which
belong to separate branches of the ontology). Moreover, such a context-specification
could be used for by a reasoner to derive inferences from incomplete or inconsistent
data sources, or for performing information extraction.

3.3. Pattern #2: not all views are theories!

The second pattern is related to the fact that people often employ the term “theory” in a
loose manner, over-classifying views with different characteristics.

In our ontology, view has been defined as a generic class referring to philosophical
ideas expressing a viewpoint. That is, propositions picturing a perspective on the world
in the form of more or less structured interpretations of things and events. Examples of
view are “solipsism”, “theory of evolution by natural selection”, “philosophy of Plato”
or "a name has a meaning only in the context of a proposition" (i.e. Frege's context principle). Because of their "categorical" attitude, views usually define concepts and, in general, create the context for the definition of other meanings too (e.g. problem-areas, problems, methods etc.). A number of properties connect views to the other philosophical-ideas: e.g. views can use other ideas, tackle problems, influence and support/contrast each other, be-supported by arguments. However, the feature we want to highlight here is how views can have different granularities: from our analysis of the literature, we identified four of them. This classification is mainly related to the degree of generality they exhibit, and the level of complexity they have. So, we can have (as shown in figure 4):

- Thesis: is the least structured view, as sometimes it consists only of a standpoint in the form of a statement (i.e. an assertion). So, for example, in the context of Wittgenstein's "picture theory of language", a thesis can be the "independence of the state of things".
- Theory: is a systemic conceptual construction with a coherent and organic architecture. A theory explains a specific phenomenon (or a class of phenomena) and typically answers to an already existing problem. Examples can be Darwin’s “theory of evolution” or Quine’s “verification theory
- Philosophical-system: it might appear as a theory, at first sight, but it differs from it essentially for its generality. That is, because it spans over various problem-area, while a theory is usually confined to one problem-area only. As a consequence, theories are usually part-of philosophical systems. We can therefore define a system as the set of a person’s views that are consistently connected to each other, in such a way to form a unity (in a way, this class refers to what is normally called the "philosophy" of a thinker).
- School-of-thought: this class refers to the set of theory-types, or generic standpoints, which in the history of thought have acquired a particular significance and, seemingly, a life on their own. They correspond to widely known conceptions, or standardized intellectual trends that hint at typical ways to answer a problem (or a set
of problems). Examples are “pacifism”, “animism”, “expansionism”, “empiricism” or “monism”. A school-of-thought, compared to the other views, is not as formalized and specific as a theory, and not as general and systematic as a philosophical-system.

3.4. Pattern #3: “problematic” problem-areas

The third pattern we are presenting wants to provide a way of expressing the distinctive features of fields of study. As we will see, the difficulty here arises from the fact that we can interpret them in two different ways: a generic one (e.g. the field of “physics”) and a specific one (e.g. “Newtonian physics”). The pattern models the relations between them.

Our starting point is a problem-centred approach, that is, the decision to see the activity of philosophers as essentially an ongoing process of specifying and giving solutions to problems. Consequently, we consider any recognized area of study, of whatever type or dimensions, as a problem-area. In its simplest version, a problem-area is composed by a set of problems linked by different relational schemas, but in general, tying around a main theme. This theme, in our ontology, can be represented through a problem (has-central-problem property) or thanks to a thesis functioning as a criterion (has-criteria property). For example, “psychology”, when treated as a problem-area, can gather problems tied to the “mind-definition” problem, to the problem of “relating human behaviour to brain activities”, or to the thesis that "brain and mind can be investigated with the methods of natural sciences”.

Other features of problem-areas are that they can be related-to each other (e.g. “mathematics” and “philosophy of mathematics”) and that they can be organized into simple hierarchies (e.g. “internet-ethics” is a sub-area of “ethics”). However, we realized soon that "psychology" has a role and significance in our world that goes beyond a mere problem area. In a similar fashion, "ethics" or "cognitive science" would not be properly characterized only as instances of problem-area, for they also refer to theories or methods which have become intrinsically related to the definition of the area.

Moreover, if we consider the history of thought, the topic and description of problem areas have always been subject of many debates: different views aspire at having the ultimate vision about what the central issues to look at are, or the right methods to take. In this respect, problem-areas are not very different from other ideas that can be defined by multiple views. For example, we can just consider how different was the sense given to “philosophy of language” by the first philosophy of Wittgenstein and the second one.

In order to catch these subtle differences, we defined the class field-of-study as a problem-area that has been socially and historically recognized as separate from the others (and from being a mere agglomerate of problems). In the ontology, this is reflected by the fact that a field-of-study is not just specified by a criteria, but is defined-by a view. It is also characterized by the fact that it collects not only problems, but also ways to solve or tackle them (i.e. theories and methods). The distinguishing properties are therefore defined-by-view, has-exemplar-theory and has-methodology.

Finally, a last tricky issue regarding fields of study must be addressed (see figure 5). This does not emerge when treating relatively isolated entities such as “phrenology”, but it clearly is an issue if we consider, say, “physics”. In our everyday language, and also in the organization of academic programs, we usually refer to “physics”, “psychology” or “philosophy of mind” as generic fields of study. What this
means, is not really clear. In fact, when we delve into them (or even more, if we ask for clarifications to a practitioner), we discover quickly that there are many “physics”, “psychologies” and “philosophies”, at least as many as the views defining them. From our ontological perspective, these would all be separate instance-candidates of the field-of-study class. However, we also need to represent the fact that they are all part of a more generic (and probably emptier, for that regards its meaning) type of field of study.

Our solution to this problem consists in the creation of a generic-field-of-study class, which has no defining view but the views defining the specific fields-of-study that are claimed to be part of it. In other words, we are formalizing the fact that generic fields of study such as “physics” or “philosophy” can be defined only extensionally. So:

```lisp
(def-class generic-field-of-study (Problem-area) ?GF
  (defined-by-view :type
    (setofall ?V (and (has-sub-area ?GF ?F)
      (defined-by-view ?F ?V))))))
```

In the formula, the variables ?GF, ?V and ?F refer respectively to generic-field-of-study, view and field-of-study. Therefore, doing so we can maintain the interoperability between specific thinkers’ definitions of classic problem areas, and the generic but useful ways to refer to them. In figure 5 we give a graphical overview of this modelling pattern, highlighting the important relationships among the classes involved.

Figure 5. Problem areas and fields of study
4. Conclusion

In this article we have presented three important modelling patterns that emerged during our work with the PhiloSURFical tool. This is an application built to support students in understanding a philosophical text, through contextual navigation mechanisms based on Semantic Web technologies. The application is being prototyped with the Tractatus-Logico Philosophicus written by Wittgenstein, using a philosophical ontology we created and instantiated with the relevant data. The ontology modelling process has demonstrated to be crucial to the aim of providing valuable and non-naïve navigation mechanisms. In the paper we show how the usage of solid modelling schemas can serve to solve ambiguities in the philosophical domain, and possibly to tidy up poorly or wrongly structured data in the quickly improving Semantic Web.

5. References

Community-oriented Course Authoring to Support Topic-based Student Modeling

Sergey Sosnovsky, Michael Yudelson, Peter Brusilovsky
School of Information Sciences, University of Pittsburgh, USA
{sas15, mvy3, peterb}@pitt.edu

Abstract. Content authoring is an important part of adaptive educational system development process. The success of the large-scale transfer of these systems to the real classrooms highly depends on how actively classroom teachers will be involved in the content authoring. This paper suggests an approach to the adaptive content authoring based on the coarse-grained topics. While providing sufficient adaptation to student knowledge topics do not put an additional authoring burden on the classroom teachers. The utilization of common metadata standards and application of semantic web technologies for knowledge description facilitates the sharing and reuse of created models and enables reasoning both within a single model and across related models. On the interface level topic sharing and reuse is supported by the community-oriented learning portal. The learning portal communicates with the user modeling server, which employs newly created topic-based course structures as the basis for automatic overlay student modeling and adaptation to student knowledge.

Keywords. Content Authoring, Learning Portal, Topic, Adaptive Educational System, Content Shareability and Reusability, RDF.

Introduction

Authoring of adaptive content is one of the most important and labor-intensive activities of content-based adaptive educational system (AES) development workflow. Traditionally it relies on the design of a fine-grained domain model and careful indexing of various learning objects (LO): tutorial pages, problems, test questions, examples, etc. – with multiple domain concepts [1, 2, 3, 4]. Such approach requires from the authors the expertise in both: the domain of discourse and knowledge engineering, as well as a considerable time investment. Naturally, this fact is one of the main reasons why adaptive learning technologies are so slow to transfer from university labs to the education market. The primary author of the learning content – a classroom teacher – does not have a necessary experience in the development of sophisticated domain models and enrichment of learning content with domain knowledge, while the AES developers are physically unable to create sufficient volumes of intelligent content that can suit the needs of multiple teachers.

The complexity of content authoring in modern AES primarily originates in the complexity of the domain models used in these systems. The more detailed and precise the modeling is, the more accurately an AES can assess student knowledge, and

* By adaptive content we understand here the content of adaptive systems, enriched with the domain knowledge components
consequently the more effectively it can potentially adapt its content to the individual student. However, the important question is: where does the “golden mean” lie, what is the best tradeoff between model precision and model complexity, effectiveness of adaptation and ease of development?

The known approaches to solve the problem of content authoring in AES do not question the need for complex domain models and sophisticated content indexing, but attempt to deal with it. The promising methodology here is to rely on domain models developed by experts and provide teachers with dedicated and friendly authoring tools supporting effective indexing of learning content [5, 6, 7].

In this paper we employ an alternative approach called topic-based knowledge modeling. This approach inspired by the instructional design practice [8] is based on using coarse-grain domain models and simple indexing schemes. Our experience with adaptive hypermedia system QuizGuide [9] based on this approach shows that the domain model does not have to be very detailed to ensure the effective adaptive behavior and usability of the system. The original domain model of QuizGuide consists of only 22 coarse-grained topics describing the domain of C Programming. To compare, the formal ontology we have developed for the same domain, defines 574 fine-grained concepts. Each topic represents a fairly large chunk of knowledge and instead of indexing learning material it is used for aggregation of individual LOs. For example, the left part of figure 1 demonstrates how QuizGuide groups quizzes by topics. From the authoring point of view, the development of the topic structure of a course is a traditional teaching activity. Any classroom teacher subdivides the course curriculum into large categories and assigns to them available LOs. Several classroom studies demonstrated that despite its coarse-grained domain model QuizGuide has been an extremely efficient as an AES. Both, the increase in knowledge gain [9] and the increase in student motivation [10] were significant.

Figure 1. Topic-based Adaptation in QuizGuide

* Of course, the quality of assessment and the quality of adaptation depend on how effective the corresponding components of AES work. The model granularity sets the limit defining the possible accuracy of the system adaptation.
The paper presents our recent work on scaling the topic-based approach from a specific AES to a teacher-oriented authoring framework. The goal has been to develop a tool that allows teachers to develop AES based on their own sets of topics with minimal efforts. Our design is based on three core ideas:

- the natural for teacher hierarchical organization of learning content defined by the topic structure of a course;
- community-driven sharing and reuse of learning content facilitated by semantic web technologies for knowledge representation;
- and the automatic student modeling on the basis of newly created course model to support the immediate topic-based adaptation.

We have developed the authoring framework implementing the above principles, which allows a community of teachers to design topic-based curricula of their courses, associate topics with relevant LOs, share and reuse each other’s topics and even entire course structures. This functionality is implemented on the new version of KnowledgeTree learning portal [11].

The rest of the paper has the following structure. Section 1 gives the details of our topic-based authoring approach. Section 2 describes the authoring interface that allows a teacher to design a topic-based course structure and categorize LOs. Sections 3 and 4 explain the internal functionality of the system. Section 3 concentrates on the model representation, while the fourth section talks about the roles of the Ontology Server and the User Model (UM) Server as well as the system communication flow. Finally section 5 discusses the directions of future work and concludes the paper.

1. Topic-based Knowledge Modeling and Adaptation

In a content-based AES the main purpose of a coarse-grained topic is the same as of a traditional fine-grained concept – to represent an element of knowledge in the domain of discourse, to serve as a basis for identification of student understanding of a corresponding part of learning material and to support a proper adaptation. Yet, topic-based approach to knowledge modeling is different in several ways:

- Topics provide a useful way of learning material aggregation, but not indexing. As a result the relationships between topics and LOs are “1-to-many” (1 topic corresponds to many LOs) and topic-based “indexes” do not exceed manually manageable numbers.
- Topics organize a natural approach for a classroom teacher to divide the course into logically separate units and assign to them appropriate pieces of content. Consequently, the authoring of the topic-based domain model can be easily done by a classroom teacher while s/he is developing the course structure.
- In adaptive systems topics can play two roles: as knowledge components for the student modeling and content-based adaptation, and as interface elements for the content structuring and navigation.
- Topics are coarse-grained; therefore, when relying on topic-based knowledge modeling we sacrifice the model precision. The results of our evaluation of the quality of topic-based student modeling in QuizGuide [12] show that the precision and predictive validity of such models are not high. Nevertheless,
several classroom studies have demonstrated that the adaptation stays efficient, and the overall learning impact of the system is significant.

- Topics are subjective. If we compare topic structures of the same course developed by different teachers, with a high degree of certainty we would expect them to be different. The presence or absence of a single topic, the naming labels, the size of particular topics, the inter-topics relations, and the scope of the entire set can vary from one structure to another based on the personal decision made by a teacher.

To summaries, topics are unique knowledge components, which in the framework of adaptive learning have some pros and cons comparing to smaller concepts. However, the main advantage of a topic is that while ensuring sufficient adaptation quality it provides a classroom teacher with a natural approach to adaptive content authoring from the beginning to the end.

2. Community-oriented Authoring of Topic-based Course Models

Virtually in every course management system teachers organize their content hierarchically, by breaking the course material into fairly large chunks (lectures, themes, topics) [13, 14, 15]. Our approach follows this, natural for teachers, procedure; the entire process of learning content authoring revolves around topics. From the point of course structuring topics could be considered as containers for LOs. The interface of the KnowledgeTree learning portal provides necessary functionality for topic creating and editing as well as for categorizing available LOs with the topics (fig. 2).

Figure 2. Adding/Editing Topic (a), Adding/Editing LO – Quiz (b), and Specifying Degree of Contribution of LOs to a Topic (c)
When adding a new topic to his/her course structure a teacher needs to specify the topic title and description. Both of these could be altered later using identical interface (fig. 5a). When associating a LO with a particular topic a teacher enters its title and the URL where the LO is deployed (fig. 5b). If a teacher wants, he/she also specify a degree of the contribution made by the particular LO to a topic (fig. 5c); otherwise all LOs associated with a topic will have equal weights. Essentially, a weight designates the percentage of work towards mastering a topic that students perform while completing a LO. The topic-based adaptation will depend on these weights. The bigger the weight is the stronger would be the influence of the student activity with a particular LO on this student’s knowledge model for the topic.

We try to minimize the intervention of the administrators of the adaptive systems and services. Teacher performs the bulk of the authoring activities. A typical course-authoring scenario on KnowledgeTree learning portal consists of the following steps:

1. Portal administrator initializes a course. A new course node is assigned to a designated teacher that would author the course.
2. A teacher authors the course. He/she can choose to create the course from scratch or to reuse the content and structures the courses already created by the community. The course can be replicated entirely and modified afterwards if needed by adding, removing and editing topics, or the teacher might just copy some of the existing topics to his course.

At the stage of course initialization on the portal the administrator specifies the course name (that can be changed by the course author later) and a URL of the course RDF model (fig. 3).

Once the course entity has been created and necessary access rights granted to the author s/he could start working on the course structure. KnowledgeTree provides teachers with the possibility to reuse the material of existing courses. The learning portal will become a sharing point, where multiple teachers giving the same course over the years accumulate solid amount of learning content that could be reused for future iterations of the course. Besides, many courses share parts of the material. For example, courses on C and C++ are likely to have similar topics on Loops and Conditions. The LOs from such topics could be used in both courses. A teacher creating a course on KnowledgeTree can take a previously authored course as a starting point and replicate its entire structure or borrow only some of its topics with their LOs (fig. 4). If the teacher is fully satisfied with the pieces s/he has copied from pre-authored courses; that is all s/he has to do. However, if
there are topics and subsequent LOs the teacher wants to add, change or remove, s/he can use portal tools to do that. Such functionality allows any new teacher entering the portal to have an access to the “community wisdom” and reuse learning content provided by multiple authors on the desired level of granularity.

3. Portable Representation of Content Models

KnowledgeTree supports sharing and reuse of created models not only on the level of interface. To ensure potential portability beyond the system and to facilitate the import of foreign models to the portal, for knowledge representation we use RDF [16] along with the common metadata standards, such as Dublin Core [17] and IEEE LOM [18]. RDF has been chosen for its wide spread in the framework of semantic web initiative. It has become a “de-facto” standard for knowledge description on the Web. One more argument in its favor is the availability of tools for design of RDF documents (such as Protégé [19] and Altova SemanticWorks [20]) and for implementing inference on RDF graphs (such as Jena).

When a teacher authors a course, s/he does not have to understand the internal representation of underlying models. Copying or creation of a topic, binding of a new LO, even adjustment of weights between a topic and corresponding LOs do not require specific skills beyond traditional Web-based Course Management. On the system level any authoring activity in KnowledgeTree utilizes three models: domain model, LO model, and course model.

3.1 Domain Model

As mentioned in section 1, from the point of context-based knowledge modeling the main purpose of a topic is to represent a chunk of the domain. When a teacher structuring a course defines a topic, he/she categorizes not only some amount of learning material, but also a coherent piece of domain knowledge. As a result during course authoring the teacher indirectly specifies the domain model adjusted for his/her course. This model is used automatically by the UM server as a basis for overlay student modeling and topic-based adaptation. Figure 5 demonstrates an example of topic-based domain model for C programming, which is, essentially, a set of rdfs:Class’s. Sometimes the author of the
course might want to specify relations between topics. For example, on fig. 5 topics \textit{IfElse} and \textit{ComplexIfs} are connected with the prerequisite-outcome relation.

![Figure 5. Topic-based Domain Model (C Programming)](image)

3.2 LO Model

KnowledgeTree portal is not used for authoring any type of LOs, hence it does not need to modify the LO description model. However, it is capable of interpreting certain information about LOs represented using standard RDF vocabularies. For example, fig. 6 visualizes the model of a quiz (\textit{lom-edu:Exercise}). It specifies the author of the quiz (\textit{dc:creator}), its url (\textit{dc:identifier}), the quiz metadata document (\textit{rdfs:isDefinedBy}). It also shows that the quiz consists of 4 questions ordered linearly.

![Figure 6. LO Metadata](image)
3.3 Course Model

The main model behind the course authoring is the course model. It utilizes topics from the domain model and LO descriptions from the LO metadata. The primary purpose of the course model is to specify the schedule of topics and association between topics and LOs. Figure 7 visualizes a fragment of the C Programming course model demonstrating such associations. Here topic *IfElse* includes two quizzes; contribution of quizzes to the topic is specified as well.

![Figure 7. Topic and Associated LOs](image)

From the point of content-based modeling these three models provide traditional representation of the adaptive content: the domain model specifies knowledge elements, the LO metadata describes available learning material, and the course model supplies the “index” of pieces of learning material in terms of domain knowledge.

4. The Back-End Communication Mechanisms

The topic-structure of a course plays several roles in our approach. The learning portal uses it to categorizes the content, however it also serves as a representation of the domain model and, hence, as the basis for student knowledge modeling and adaptation. UM server is responsible for inference, representation and reporting of student knowledge of the specified topics. The models described in the previous section are stored separately on the ontology server. This section explains how separate models are processed and how different systems communicate to each other to ensure the immediate plugging-in of a created topic structure and modeling of student knowledge in terms of newly created topics.

4.1 Communication with SEDONA Ontology Server

While the course authoring interface is served by the learning portal, the underlying processing of created models is mostly done on the ontology server. SEDONA, which stays for Server of EDucation ONtologies for Adaptation [21], communicates with KnowledgeTree learning portal via HTTP protocol. When a teacher designs the course structure, SEDONA discovers necessary models, merges them together and sends the enhanced course model to the portal. Whenever the teacher creates a new topic or adjusts an existing one, the course model and domain model are modified on the ontology server.

The sharing and reuse of topics is handled by SEDONA as well. If an author of a course wills to use an existing topic, SEDONA does not create a duplicate of this topic in
the domain model corresponding to the course. It simply uses the topic URI from the donor model. However, if any modification has been performed, a new version of the topic will be created.

4.2 Modeling of Student Knowledge

The course structure that has been authored via portal interface and is stored on SEDONA is reused for modeling student knowledge. Our user modeling server CUMULATE [22] is aware of the courses authored on KnowledgeTree and stored on SEDONA. Whenever a KnowledgeTree course is created or modified, CUMULATE replicates its topic structure as a domain model. It also retrieves from SEDONA associations (possibly weighted) between topics and LOs and processes them as an index.

This information is enough for CUMULATE to populate overlay topic-based models of student knowledge. Whenever a student works with a LO associated with a topic, her/his knowledge level for this topic is updated and can be reported by CUMULATE upon request from the learning portal or other interested certified applications. Hence, the entire process of adaptive content development for a teacher is now performed during structuring of the course on the learning portal. Once a topic is created it automatically becomes available for knowledge assessment and adaptation.

5. Conclusions and Future Work

In this paper we have described an approach to effective authoring of adaptive content for AES, based on three principles:

- a teacher-friendly approach to authoring of adaptive topic-based learning content;
- a community-oriented framework allowing topic sharing and reuse enhanced with semantic web technologies for knowledge representation;
- automatic modeling of student knowledge and adaptation on-the-fly based on newly created course structures.

One of the prospective work directions is to implement a mechanism allowing a user modeling facility to benefit of topics sharing. While on the level of interface, course authors can use each other’s topics, CUMULATE is not aware that two topics from different courses are actually the same. By implementing the inter-model inference of student knowledge we will allow the reuse of modeling information for similar topics, so that the work of a student in one course could be taken into account in the relevant course for the topics shared by the courses.

References


Using SWRL for ITS through Keyword Extensions and Rewrite Meta-Rules

Eric WANG and Yong Se KIM
Creative Design and Intelligent Tutoring Systems Research Center
Sungkyunkwan University, Suwon, Korea
{ewang, yskim}@skku.edu

Abstract. We have established a teaching strategies engine using rules in SWRL, with automatic conversion to Jess for execution. We identify limitations of SWRL that make it unusable as-is for ITS applications. To overcome these, we devise a generic approach of extending SWRL by adding new keywords as OWL ontology classes and properties, and post-translating them using rewrite meta-rules. The resulting “extended SWRL” language is sufficiently expressive for practical ITS development. This work supports the adoption of some of these extensions directly into SWRL; more generally, it establishes the meta-approach of customizing SWRL itself.

Introduction

Within an ITS, one module generally applies the teacher’s pedagogical strategy. From the learner’s current state (including recent activity), this module schedules content selection and presentation, tutoring feedback, learner assessment, and other teaching actions. We call this module the teaching strategies engine.

Teaching strategies are a kind of procedural knowledge, including assessment algorithms and decision procedures. We explore the premise that teaching strategies can be expressed using existing IF-THEN rule standards such as SWRL. We have developed a concrete implementation of a teaching strategies engine using SWRL [2] and a standard rule engine. The teaching strategies are written as rules in an “extended SWRL”, as described in this paper. The engine itself is a running instance of Jess [1]. We have previously established an automatic conversion from SWRL to Jess, using Protégé SWRLTab [3], SweetRules [4], and other standard tools [6]. In this paper, we describe the extensions to SWRL that we found necessary to implement practical rule sets for an ITS.

1. An Assessment of SWRL for ITS

At a generic level, an ITS interacts one-on-one with a human learner over a finite session of activity, maintains a model of the learner, and performs teaching actions. We identify intrinsic attributes of ITSs that are relevant for teaching strategy rules.
1.1. Attributes of an ITS

A. **Stateful.** An ITS maintains a model of the learner, which is generically a set of dynamic data fields.

B. **State-changing.** Since the human learner evolves over time (by learning the course content), the ITS’s learner model must also evolve over time. We distinguish two kinds of state change: (B1) adding new knowledge only, and (B2) changing or retracting existing knowledge.

C. **Calls system services.** An ITS typically interacts with the learner by using underlying system services, esp. window management, sound, etc.

In addition, an ITS is typically **interactive**, **persistent** over a finite session, and **event-driven**. These secondary attributes typically don’t need to be directly managed by teaching strategy rules, so we assume they are provided by the ITS framework.

1.2. Limitations of SWRL

The current SWRL 1.0 specification omits some typical rule language constructs, sacrificing some expressiveness to ensure decidability and/or efficiency. We have previously identified some limitations and workarounds in writing teaching strategies in SWRL [7]; in this work, we directly address these limitations.

D. **Flat list.** A SWRL rule body (or head) is a flat list of atoms. No block structure is supported (as there are no constructs that would need them). Note that either of #E or #F would also entail adding block structure.

E. **Conjunction only.** Disjunction (E1) and negation (E2) are excluded.

F. **No quantifiers.** Explicit universal and existential quantifiers are excluded. All atoms are implicitly universally quantified.

G. **No user-defined functions.** SWRL provides a library of built-in functions for primitive math, string, and date operations. It doesn’t allow external functions to be defined or called (which fails 1.1#C).

H. **Assertions only.** A SWRL rule head “makes” its atoms true; the standard implementation is to add these atoms as new facts to the knowledge base. Existing knowledge cannot be changed.

Some omitted language constructs, e.g. negation and existential quantifiers, can be simulated through clever use of OWL DL workarounds. But current OWL + SWRL tools are not yet mature enough to fully handle these techniques. We note that several of these constructs are already planned for inclusion in the next iteration of SWRL 2.0. Given that these constructs are already within the scope of OWL DL + SWRL, we may add them in an easier-to-use form without decreasing SWRL’s decidability – hence we explore the approach of adding them ourselves, as new keywords.

OWL DL + SWRL can achieve desired attributes 1.1#A and 1.1#B1 only. We need a way to achieve 1.1#B2 and 1.1#C. For this, we extend SWRL with our own keywords, and define our own semantics for them.
2. Extending SWRL

We have devised a generic approach for adding new language constructs to SWRL. Briefly, we add new keywords as OWL ontology classes or properties, convert SWRL rules to Jess format, and apply meta-rules to rewrite the keywords before evaluating the rules in Jess. We give one example; other keywords are summarized in Appendix A.

- **State change.** Jess allows modification of existing facts by using a bind-modify pattern pair. This overcomes limitation 1.2#H. In a rule body, ?R <- P is an assignment-like statement that matches any fact P and stores (a reference to) it in the Jess local variable ?R. In a rule head, (modify ?R Q) updates that fact with expression Q, which is usually a new <slot, value> pair.

We use standard tools as much as possible. We edit teaching strategy rules in Protégé SWRLTab, save them in Protégé’s OWL file, and use XSLT and SweetRules to convert them to CLIPS (Jess) format [6]. These tools are agnostic to SWRL rule atoms, and translate them verbatim, without needing to know their semantics.

2.1. Adding Extension Keywords

Every (non-trivial) SWRL atom must be an OWL class or property name. Conversely, any OWL class or property can immediately be used in writing SWRL rules. Class atoms are always arity-1 (they take 1 argument), and property atoms are always arity-2; other arities are not permitted.

- The Jess <- operator and modify keyword are both arity-2. Their 2nd argument is recursively a SWRL property atom P. We separate these into two-atom sequences, e.g. __bind(?R) ^ P. __bind (respectively, __modify) is then arity-1 in SWRL, so we add it as an OWL class. The “__” prefix distinguishes our keywords from domain ontology elements.

We define our SWRL extension keywords as OWL classes and properties in a separate ontology, a portion of which is shown in Figure 1(a). Each teaching

![Figure 1. SWRL rule writing using the __bind extension keyword](image)

We define our SWRL extension keywords as OWL classes and properties in a separate ontology, a portion of which is shown in Figure 1(a). Each teaching
strategies ontology imports the keywords ontology. We may then freely use these keywords in writing SWRL rules, as shown in Figure 1(b). After conversion via SweetRules, our keywords appear verbatim in the CLIPS output (with the default SWRL semantics); an excerpt from our translation output is shown in Figure 1(c).

2.2. Rule Rewriting in Jess

Once the rules are in Jess format, we apply our own meta-rules to rewrite those rules that contain any of our keywords. Jess (equivalently, Java) has a Lisp-like ability to treat unevaluated rules as data, manipulate them using list and string operations, and evaluate a list or string as a rule.

An unevaluated Jess rule has a tree structure, representing parenthesis nesting depth, whose leaf nodes are Jess lexemes (tokens). We represent a tree-of-tokens as a Vector-tree of String, using standard Java classes, i.e. the elements of each Vector are either String for a leaf token, or recursively a Vector-tree for a sublist. Converting a Jess rule string from/to its Vector-tree representation is straightforward.

We define one rewrite meta-rule for each keyword, as a tree rewriting operation, which examines a rule’s Vector-tree, modifies any elements, splices any new subtrees into place, and so on. Each meta-rule deletes its own keyword from the rule (or replaces it with the corresponding Jess keyword), which ensures that our SWRL extension keywords are never actually evaluated in Jess.

- __bind: If this Vector matches (see below) a __bind keyword, then replace this Vector (in entirety) with the simple string “?R <-”, where ?R is the variable name at this Vector’s position 2.

Keyword matching is done with respect to SweetRules’ output. Our __bind keyword is an OWL class, so SWRL deems it to be a class membership test. SweetRules, being agnostic to our keywords, converts __bind to the Jess rule pattern for a class membership test. In SweetRules’ RDF triple terminology (with the ontology’s namespace URI elided), this is written as (triple rdf#type ?R __bind). Keyword matching checks whether the current Vector-tree is of depth 1, length 4, and matches at positions 1 and 3 (using Java 0-origin indices).

The rule rewriting step uses a simple iteration that applies every meta-rule to every Jess rule string. Translated rule strings that have changed are then re-evaluated in Jess.

3. Discussion

This research has established a working teaching strategies engine using current standard Semantic Web tools, which we use as a module in two ITSs. We represent teaching strategy knowledge as SWRL rules in an ontology. This has benefits in openness and reusability of teaching strategy knowledge.

We assess that SWRL (in its current 1.0 state) isn’t expressive enough to write rule sets for an ITS. We solve this by adding the missing language constructs as new keywords in OWL/SWRL, and a translation step to apply our own rewrite meta-rules. This is tolerable, since most SWRL usage today requires some translation regardless.
Our current implementation lifts some Jess constructs into SWRL, resulting in a “SWRL-j” that is expressive enough for our ITSs. This isn’t surprising: we know we could write our ITSs in Jess directly [5], and our “SWRL-j” is equivalent to the subset of Jess that our rules actually use. While this approach has been effective, it poses the question of whether it is parsimonious – perhaps it strays too far from the intent of SWRL? We suggest that this keyword-extension technique could be a viable mode of SWRL usage, allowing each user community to customize SWRL as needed.

Future tasks for this work include (a) implementing a library of teaching strategies from educational learning theories, (b) exploring meta-strategy control mechanisms, and (c) adding an interface for teachers to customize their own teaching strategy rules.

Appendix A. Extension Keywords for SWRL-Jess

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Jess Construct</th>
<th>SWRL Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_name(&quot;str&quot;)</td>
<td>Sets the Jess rule name to str. SweetRules ignores Protégé’s rule names.</td>
<td></td>
</tr>
<tr>
<td>_test(&quot;str&quot;)</td>
<td>Jess test(str) pattern. SweetRules omits some SWRL built-in functions.</td>
<td></td>
</tr>
<tr>
<td>_not(?P)</td>
<td>Jess negation construct: (not P), for an arbitrary atom P. Overcomes 1.2#E2.</td>
<td></td>
</tr>
<tr>
<td>_bind(?R)</td>
<td>Jess pattern-binding construct: ?R ← P.</td>
<td></td>
</tr>
<tr>
<td>_modify(?R)</td>
<td>Jess state-change construct: (modify (?R P)). Overcomes 1.2#H.</td>
<td></td>
</tr>
<tr>
<td>_salience(?, v)</td>
<td>Jess rule priority mechanism.</td>
<td></td>
</tr>
<tr>
<td>_logical(?)</td>
<td>Jess modal-effector construct, roughly “if P then assert X, else retract X”.</td>
<td></td>
</tr>
<tr>
<td>_call(?)</td>
<td>Jess function call construct: (P…). Overcomes 1.2#G.</td>
<td></td>
</tr>
<tr>
<td>_all(?A)</td>
<td>“If all” (unanimous vote) test: “Q is true at least once, and never false”.</td>
<td></td>
</tr>
<tr>
<td>_nand(?A)</td>
<td>Limited block structure: (not (and …)). Partially overcomes 1.2#DF.</td>
<td></td>
</tr>
</tbody>
</table>

Acknowledgements

This research was supported by the Korean Ministry of Science & Technology through the Creative Research Initiative Program.

References

Learning a Domain Ontology in the Knowledge Puzzle Project

Amal ZOUAQ¹, Roger NKAMBOU², Claude FRASSON¹

¹Univeristy of Montreal, CP 6128, Succ. Centre-Ville, Montreal, QC, H3C3J7
²UQAM, CP 8888, Succ. Centre-Ville, Montreal, QC, H3C3P8
{zouaq, frasson}@iro.umontreal.ca
nkambo.roger@uqam.ca

Abstract. In this paper, we present a (semi) automatic framework that aims to produce a domain concept map (DCM) from text and to derive a domain ontology from this concept map. This methodology is particularly aimed at the educational field because of the need of such structures (Ontologies and CM) within the e-Learning and AIED communities to sustain the production of e-Learning resources tailored to learner’s needs. This paper details the steps that transform textual resources (and particularly textual learning objects) into a domain concept map and explains how this abstract structure is transformed into a more formal domain ontology. The paper also shows how these structures make it possible to bridge the gap between e-learning standard learning objects and Intelligent Tutoring Systems.

Introduction

The importance of automatic methods for enriching knowledge bases from free text is acknowledged within knowledge management and ontology communities. In fact, static knowledge bases are hard-to-maintain, time-consuming and expensive. This is especially true in the domain of online training. Learning object repositories represent rather static repositories and suffer from various shortcomings:

- First, learning objects are represented as “black-boxes”. Metadata such as SCORM or LOM are used to describe various characteristics of the learning objects but do not model their inner content;
- Second, they suffer from their lack of adaptability to a learner model;
- Third, they do not obey to a computer-understandable pedagogical framework;
- Finally, they are restricted to the e-Learning community whereas they should offer a base for various kinds of training systems, such as intelligent tutoring systems.

In fact, there should be gains from integration and cooperation between the e-Learning and Intelligent Tutoring System (ITS) communities [3]. On one hand, eLearning-based environments focus on the reusability of learning resources, but they are not adaptable to suit learner needs, they do not really use instructional strategies and they do not have a rich knowledge representation. On the other hand, ITSs exploit rich knowledge structures, provide adaptive feedback and implement pedagogical strategies. However, their knowledge base is generally not reusable because it suffers from its dependence to the application domain and from proprietary programming. If
we think about learning objects as resources to dynamically build an ITS Knowledge base, then it should be possible to benefit from both worlds.

This paper is at the crossroad of several disciplines, including ontology learning from text [7, 10], concept map generation [19] and learning object generation and adaptation [4, 12]. From a knowledge acquisition point of view, one of the specificities of the approach presented here lies in the generation of a domain concept map from grammatical typed dependencies and the transformation of this domain concept map into a domain ontology. From a knowledge exploitation point of view, both domain concept map and ontology sustain the aggregation of learning objects and are usable in training environments. Other approaches to learning object generation did not base their domain knowledge on an automatically-generated domain ontology and did not handle the whole knowledge acquisition and exploitation process.

In this paper, we focus on the Knowledge Puzzle’s framework for extracting concept maps from text and transforming them into a domain ontology. We present an experimentation on manuals about the SCORM domain and we analyze preliminary results. The paper is organized as follows: First, we present the Knowledge Puzzle’s foundations and provide some definitions. Second, we describe the (semi) automatic knowledge acquisition from text and the domain concept maps and ontology generation. Third, we underline the interest of the approach to the educational community by explaining how the knowledge base can be used to generate Learning Knowledge Objects. Finally, an evaluation of various results is explained before a conclusion.

1. The Knowledge Puzzle Approach: Foundations

This paper describes a semi-automatic methodology for building domain ontologies from concept maps generated through natural language processing. One of the specificities of the Knowledge Puzzle’s approach lies in the use of intermediate knowledge models (concept maps) due to the dedication of this ontology to training.

*Domain Ontology* refers to a specific vocabulary used to describe a certain reality. It presupposes the identification of the *key concepts* and *relationships* in the domain of interest.

A *concept map* represents a semantic network showing the domain entities and their relationships. As such, it constitutes a skeleton on which to build more complete domain ontologies.

Human validation is essential at each step of the knowledge acquisition process. The design of a domain ontology is not a linear process; it involves many revisions before a final consensual solution is developed. Moreover, because the Knowledge Puzzle Platform is designed with the final aim of training, it is very important to validate the results obtained at each step of the mining process. A human expert has to confirm and complete the results to guarantee the quality of the ontology.

1.1. Domain Knowledge Acquisition

In computer-based education and particularly in intelligent tutoring systems, the domain knowledge is defined as the representation of the expert’s knowledge. Domain
knowledge is usually divided into two components: declarative and procedural. The Knowledge Puzzle focuses on mining declarative knowledge for the moment. It relies on natural language processing and machine learning.

The learning process in the Knowledge Puzzle’s approach is based on a syntactic and domain-independent approach. The Knowledge Puzzle instantiates a set of extraction rules from a set of generic domain independent templates. Contrary to many ontology learning approaches, it does not guide this extraction and it does not rely on a supervised method that guides the learning process.

The domain knowledge acquisition process relies on a number of steps: term extraction, relationship extraction, concept extraction, instance and attribute extraction, and finally OWL conversion.

1.2. Terminological Extraction

Terminology extraction refers to the discovery of terms that are good candidates for the concepts in an ontology. It can be facilitated by the exploitation of learning objects as the primary source of knowledge: learning objects are purely didactic documents, providing definitions and explanations about concepts to be learned. These concepts share the properties of low ambiguity and high specificity, due to their natural engagement in learning.

First, the Knowledge Puzzle needs to determine the content of each document. It works on plain text documents and partitions them into a set of paragraphs, containing a set of sentences. This is performed with annotators developed with the IBM’s Unstructured Information Management Architecture (UIMA) [17].

Second, the Knowledge Puzzle uses a machine learning algorithm, Kea-3.0 [6], to extract representative n-grams from documents. We slightly modified the initial algorithm to process one document at a time, instead of working on a collection of documents. The extracted seed expressions (one or more words) are then used to collect the sentences containing them. A natural language processing parser, the Stanford Parser [9], processes these sentences to output typed dependency representations: a set of grammatical links describing the relationships between different words. This process has been described elsewhere [5].

Each sentence is represented as a grammatical concept map, i.e. a set of terms linked by the Parser’s typed dependencies.

We elaborated a set of rules that exploit the grammatical concept maps to retrieve particular predefined patterns. These patterns serve to extract a semantic concept map from the grammatical one (semantic terms + relationships). A Pattern is represented by a set of input links and a set of output links. These links are represented by the various grammatical relationships that are output by the Stanford Parser [5]. Once a Pattern is identified, a method is fired to retrieve the semantic structure associated to this pattern.

Terminological patterns rely on modifiers such as adjectives or nouns to restrict the meaning of the modified noun, e.g. is-a (intelligent tutoring system, tutoring system). They constitute a very accurate heuristic for learning taxonomic relationships [10].

As an example, Figure 1 depicts a grammatical concept map that describes the different dependencies for the sentence: “An asset can be described with asset metadata to allow for search and discovery within online repositories thereby enhancing opportunities of reuse.”
The terminological extraction patterns allow to define the following domain terms: Asset, Asset metadata, Opportunities, Reuse, Search, Discovery, Online repositories. All these terms are candidate terms to express domain concepts.

1.3. Relationship Extraction

Domain terms must be related in some way. Relationship extraction refers to the identification of linguistic relationships among the discovered terms. Central to this extraction are verbs and their auxiliaries, which generally express also a domain knowledge. Again, the same pattern structure is used to extract relationships of interest.

Figure 2 describes the final terms and relationships for the sentence in figure 2. As you can notice, the following verbal relationships are extracted and shown in italic:

- Asset can be described with Asset Metadata
- Asset Metadata to allow for search
- Asset Metadata to allow for discovery
- Asset Metadata thereby enhancing opportunities

Other non-verbal relationships are extracted to complete the meaning of the previous tuples:

- Search within online repositories
- Discovery within online repositories
- Opportunities for reuse
1.4. Domain Concept Maps Generation

The process described above is repeated over all the selected sentences. It is then possible to retrieve a term or a particular relationship and to be automatically directed to the source sentence, to the source paragraph or to the source document. This allows an enhanced retrieval of the appropriate knowledge.

Each sentence is associated with its main subject (the term Asset in the previous example).

In the Knowledge Puzzle, a term is considered as a concept if:

- It is the main subject of various sentences, thus being a frequent subject in the domain of interest.
- It is linked to other domain terms through semantic relationships.

It is worth noting that a single concept may be expressed in different ways in the text. The Knowledge Puzzle is able to recognize the base form of a concept through stemming. The Knowledge Puzzle uses a java version of the Porter Stemmer [14] to produce the stem associated with each concept. For example, the words “stemmer”, "stemming" and "stemmed" have the same root form: “stem”. This allows particularly recognizing plural and singular forms. Another way of expressing a concept is through an abbreviation e.g.: Sharable Content Object = SCO. The Stanford Parser outputs abbreviation links as typed dependencies but they are not always correct. So, the Knowledge Puzzle implements an algorithm to identify if an indicated abbreviation is correct and if true, stores the abbreviation as an acronym of the current concept.

1.5. Domain Ontology Generation

Besides domain concepts and relationships extraction, domain ontology generation consists of instance and attribute extraction, and OWL conversion. The domain concept maps constitute the knowledge base for this generation. The Knowledge Puzzle uses a linguistic approach to determine attributes and instances.
Instance extraction enables to find objects that are instances of a particular concept. Hearst [8] was the first to talk about linguistic patterns for identifying hyponyms (“is a kind of”). Particularly, the pattern “NP1 such as NP2, NP3 and NP4” indicates a hyponymy relationship.

It seems sometimes a little bit difficult to differentiate linguistic expressions indicating an “instance-of” relationship from expressions indicating a sub-class relationship. Let’s suppose that NP1 is a concept. We established the following rules:

• If NP2, NP3, or NP4 are also concepts, then they are considered as sub-classes of NP1.
• Otherwise, if NP2, NP3 and NP4 are not considered as concepts, they are stored as instances of NP1.
• Finally, if NP1 is not a domain concept as previously defined, then we leave the relationships “is a kind of” between these terms and the human evaluator is free to modify it to a sub-class or an instance.

As far as attributes are concerned, two kinds of relationships can be extracted:

**Inner Attributes** that describe the concept itself. Term attributes can be extracted using contextual information or relies on nominal modifiers as expressing a potential property.

**Verbal Relationships** that describe the relationship of the concept with other ones. We find occurrences of these patterns in the text, and we create lists of attributes for all the terms in the corpus.

The Knowledge Puzzle uses the following patterns to extract concept attributes:

• `<attr> <C> <verb> ...` where C is a concept and attr a modifier. An example of text matching this pattern is: … inline metadata is … where metadata is a concept.
• `<attr> of <C>` (e.g., “identifier of asset”) or `<C>’s <attr>` (“asset’s identifier”).
• `<C> have/possess <attr>`

Similar techniques for identifying concept’s attributes can be found in [1, 13].

Verbal relationships express more specialized relationships that are important in the domain. Basically, all verbal relationships between pairs of concepts are considered as potential ontological relationships.

Concepts are exported as OWL Classes and verbal relationships as domain properties (OWL Object Property). Attributes are considered as data type properties. Instances are stored as individuals of their classes. Taxonomic relationships are converted into sub-class links. The whole is exported as an OWL file.

2. The Interest of the Approach for the AIED and e-Learning Community

One of the more acute problem of knowledge-based systems is the famous “knowledge acquisition bottleneck”. This is also an issue in training communities (e-Learning and Intelligent Tutoring Systems). Like we previously said, Intelligent Tutoring Systems have richer and more fine-grained knowledge representations than e-Learning systems.

Given the large amount of learning objects, it is possible to annotate their content semi-automatically through the generation of domain ontologies. From one side, this enables an accurate representation of learning objects and their contexts. It permits also
a more focused answer to learners’ needs and provides them with navigation facilities through the learning resources. From another side, this semantic annotation creates a common knowledge base representing a bridge between Intelligent Tutoring Systems and e-Learning systems: the automatic acquisition of a domain ontology enables to capture the semantics of the subject domain. Moreover, the idea of generating a domain ontology from concept maps is particularly useful for the educational field.

2.1. Context and Constructivism

Concept maps are not considered only as intermediary templates for building the domain ontology. In fact, concept maps alone have proved their usefulness for training within constructivist environments [11]. The Knowledge Puzzle defines a formal relation between a concept and the generated concept map to enable its reuse in training sessions. This formal relation represents the concept’s context. This is very important within constructivist environments where prior knowledge is used as a framework for understanding and learning new knowledge.

Constructivist learning should allow exploration with learning material and related domain concepts. It should foster meaningful learning by relating previous knowledge with a new one. As previously shown, the Knowledge Puzzle offers such capabilities.

2.2. Learning Objects Types

We underlined the importance of more fine-grained retrieval of learning content. We believe that an ontology of instructional roles is essential to reach this goal [18]. In fact, learning objects can include various types of learning materials such as lectures, examples, exercises, etc. Due to its pedagogical natural aim, a learning object itself contains a set of instructional roles. In order to provide a standard model for binding knowledge and instructional roles to documents or parts of them, we propose to model an OWL ontology: The instructional role ontology. It defines instructional roles that can be found in documents and that can be used to provide a template to a learning material structure. This ontology can evolve according to expert’s desires. We provide him with a tool to manually annotate the document according to instructional roles [20]. Instructional roles are applied on domain concepts, thus providing an inter-related meaningful structure (e.g.: an explanation about a concept X, a definition of concept Y, etc.)

2.3. Generating Learning Knowledge Objects

Because concept maps are extracted from domain learning objects, they constitute an interdependent structure of domain concepts and a library of learning materials. These learning objects must also be adapted to fit individual needs at various levels. The Knowledge Puzzle focuses on the definition of competencies as instructional objectives. It relies on a Competence Ontology that describes a competence as a set of skills on domain concepts. Skills are classified according to the Bloom taxonomy [2], which defines six levels of intellectual behaviour important in learning and associates a
set of verbs with each level. We also use action verbs to qualify the skills involved in a competence.

We adopt a lazy aggregation approach for learning knowledge objects generation: we push back the aggregation process until there is a need. This gives enough flexibility to the aggregation process and enables individual adaptation. This also means that learning knowledge objects are not stored, but automatically assembled on demand. A Competence Gap Analyzer compares the learner profile with the competence definition to detect training needs. Learner’s instructional objectives are then indicated to the Instructional Plan Generator (IPG).

The IPG must follow an instructional strategy in order to provide efficient training through instructional theories. The IPG can be configured to use various instructional theories. For the moment, we created a basic instructional theory editor that indicates the different steps of a theory and that map the aforementioned instructional roles to each step of the theory. For example, the theory can state that an introduction about a concept X must be presented first. Instructional theory steps are stored as instances of an instructional theory ontology.

A theory-aware learning knowledge object can then be generated according to the available theories. In fact, any instructional theory could be applied to constitute a Learning Knowledge Object. Learning Knowledge Objects are then ready for presentation in a training environment. It is also possible to standardize their structure in order to make them compliant with an e-Learning standard such as SCORM and IMS-LD [20, 21].

3. Evaluation and First Results

In this paper, we evaluate two facets of the Knowledge Puzzle: the domain concept maps and the domain ontology (domain concepts and relationships).

We ran a small-scale evaluation on the Knowledge Puzzle’s domain concept maps using a set of training manuals that present and explain the Sharable Content Object Reference Model [16]. Because SCORM manuals contain not only declarative knowledge in natural language, but also codes and algorithms, we extracted manually the declarative parts of the manuals into a set of 32 plain text documents.

After having run the structure annotators, we obtained a set of 157 paragraphs with an overall number of 1302 sentences. A set of 1354 domain terms was obtained after the concept maps generation as well as 2351 semantic relationships. A threshold I was specified to consider a domain term X as a domain concept. I is the number of semantic relationships where X is the source concept (the out-degree of X). Three thresholds were considered: 2, 5 and 8.

In the case of threshold=2, domain terms have been reduced to 477 domain concepts and 628 properties with specified domain and range.

In the case of threshold=5, we obtained 76 domain concepts and 228 properties.

In the case of threshold=8, we obtained 38 domain concepts and 156 properties.

Several questions arise to evaluate the domain ontology quality, which is a challenging issue:

- Estimating the pertinence of the domain concepts;
- Estimating the correctness of the relationships (label, arguments);
- Estimating the pertinence of the domain relationships;
Domain concepts and relationships were automatically exported to an OWL ontology.

Domain concepts were exported as classes of the new domain ontology. Only plausible relationships between pairs of domain concepts were considered as valid OWL object properties. For example, if we consider the tuple: Assets can be described with Asset Metadata, then the relationship “can be described with” is an OWL Object Property whose domain is the class Assets and whose range is the class Asset Metadata. Moreover, we exported the taxonomical links (is-a links) as subclasses, and instance links as OWL Individuals.

Note that the plausibility of the domain ontology is closely linked to the plausibility of the domain concept maps.

The authors of this paper inspected the domain ontologies in order to evaluate the correctness and plausibility of domain concepts. Moreover, because SCORM manuals include a glossary of important domain terms and acronyms, the authors were able to compare them to the extracted concepts.

The following table summarizes the author’s evaluation of domain concepts and relationships’ pertinence for each generated domain ontology.

<table>
<thead>
<tr>
<th>Thresholds</th>
<th>Domain Concepts Pertinence (%)</th>
<th>Domain Relationships Pertinence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>86.79</td>
<td>70.7</td>
</tr>
<tr>
<td>5</td>
<td>92.10</td>
<td>77.19</td>
</tr>
<tr>
<td>8</td>
<td>97.36</td>
<td>87.82</td>
</tr>
</tbody>
</table>

Table 1. Human evaluation of concept and relationship pertinence.

We noted that the higher the threshold is, the lower is the generated noise. However, some pertinent knowledge is occulted. So a balanced value should be found, which takes into account the number of domain sentences that are available for processing. As far as relationships are concerned, each author judged if the domain concept map relationships were plausible. A relationship r between C1 and C2 is considered as plausible if it contains a plausible label and two plausible arguments. Plausible means that this relationship is true with respect to C1 and C2. For example: Asset can be described with Asset Metadata is a plausible relationship.

4. Conclusion and Further Work

In this paper, we presented a solution to semi-automatically generate concept maps from domain documents. These concept maps are useful to support meaningful learning and serve as 1) a knowledge base for building domain ontologies, 2) a skeleton for a more focused learning object composition. We based these extractions on lexicon-syntactic and semantic patterns.

In the knowledge acquisition side, our goal is now to tackle the issue of over-generation and of noise resulting from the overall process. The problem is to be able to automatically ignore non-domain knowledge. We also work to enhance the algorithm of the extraction patterns. It is not trivial to be able to interpret grammatical relationships semantically. Confidence rates should be attached to the extracted knowledge. Moreover, more thorough ontology and concept map evaluation techniques
must be performed. The question of the pertinence of a threshold related to the outcome-degree of each domain concept remains opened and should be further investigated.

In the knowledge exploitation side, we would like to realize the generation of more complex learning objects that exploit the concept maps as well as available instructional theory ontologies [15] to fulfill a learning objective.

References

Personal Assistant towards Semantic Information Retrieval

Zeeshan Ahmed and Detlef Gerhard

Mechanical Engineering Informatics and Virtual Product Development Division, Vienna University of Technology, Getreidemarkt 9/307 1060 Vienna, Austria
{zeeshan.ahmed, detlef.gerhard} @tuwien.ac.at

These days Web technology is the most successful source of spreading information throughout the world but deficient because there are still some areas which require improvements. Most of the unstructured information published on web is formatted in a way that it can be viewed but can not be processed. This deficiency leads to the problems of knowledge based information search, extraction and maintenance. Currently it is not possible to search and extract desired concrete level information using full text queries because there is no such mechanism which can publish data over the web in the form of machine readable and processable format or can extract the actual semantic from full text query and then look into the data warehouse for the particular knowledge based information. The mechanisms of presenting information over the web in a format that humans and machines can understand the context leads to the concept of Semantic Web introduced by Tim Berners Lee [1]. “Semantic Web” aims to create technologies capable of reasoning on semi structured information and produce meshes of information which can be linked up in machine processable way. “Semantic Web” right now is standing on two important building blocks Ontology and Semantic Web Agent [2]. Ontology is the explicit representation and description of already available finite sets of terms and concepts used to make the abstract model of a particular domain by including information in the form of properties containing values, restrictions and relationships, where as the Semantic Web Agent is an advanced implemented form of object oriented concepts having autonomous behavior in solving miraculous problems by self processing and communicating with different sources. The main and currently not achieved goal of Semantic Web is to structure the meaningful contents of unstructured data published over Web to take advantage in improving the information retrieval process [1] and to involve knowledge management in making some more advanced knowledge modelled management systems. A number of semantic based solutions have been proposed and developed but still the problems are not solved. Narrowing the scope of our research on major currently existing Semantic Web challenges and keeping in mind some already existing in time proposed and implemented semantic web and desktop based solutions we proposed an approach called Semantic Oriented Agent based Search (SOAS). The proposed architecture of the approach as shown in Figure.1 consists of an agent called Personal Agent (PA) and five dynamic processing units i.e., Request Processing Unit (RPU), Agent Locator (AL), Communication, List Builder (LB), Result Generator (RG) and Database.
The architecture of SOAS is designed to work in a sequence that user’s unstructured full text query based search requests will be received by PA, which will simply forward the context to RPU for further processing. RPU is the most important component of the SOAS divided into three subcomponents i.e., Full Text Reader, Lexer & Parser and Reconstructor to read, parser, analyse and structure the contents of unstructured full text in semantic Ontology based models which will be used by AL to find out the contact information of particular domain’s available agents using forward changing principle. Using extracted contact information Communication will contact and communicate with agents and store obtained results in database. Information stored in the repository will be extracted by LB, if results are more than one then LB will rank them on the basis of their weights and create a prioritized list of results which will be forwarded to RG to convert it into the final resultant information in acceptable human readable format which will be then forwarded to the user via PA.

We do not claim that SOAS is the only feasible solution because right now it is in an initial stage having a wide range for the improvements but still it can be a new way in looking towards the semantic search problems. In future, we will try to improve the quality of the currently proposed architecture and implement it in a software application to evaluate its effectiveness.

References
