Towards a Semantic-Rich Collaborative Environment for Learning Software Patterns

Zoran Jeremić[1], Jelena Jovanović[1], and Dragan Gašević[2]

[1] FON-School of Business Administration, University of Belgrade, Serbia
jeremycod@yahoo.com, jeljov@gmail.com
[2] School of Computing and Information Systems, Athabasca University, Canada
dgasevic@acm.org

Abstract. Current approaches to learning software patterns are based on individual use of different learning systems and tools. With this ‘fragmented’ approach it is very hard to provide support for context-aware learning and offer personalized learning experience to students. In this paper, we propose a new approach to learning software patterns that integrates existing Learning Management Systems, domain specific tools for software modeling and relevant online repositories of software patterns into a complex learning framework that supports collaborative learning. This framework is based on the semantic web technologies.

Keywords: Semantic web, ontologies, collaborative learning, software patterns.

1 Introduction

The major concern of today’s software engineering education is to provide students with the skills required for solving different kinds of software problems both on their own and as members of a development team. In addition, it is essential that students learn how to exploit previous successful experiences and knowledge of other people in solving similar problems. This knowledge about successful solutions to recurring problems in software design is also known as software patterns [1],[2]. Software patterns are becoming increasingly important in software engineering.

However, teaching and learning software patterns is not an easy task. It is rather a complex process that requires significant efforts from both educators and students. In order to secure high-quality learning in this complex engineering field, a learning platform needs to meet the following requirements:

1. Enable students to learn at the pace and in a place that best suits them as well as provide them with the content and learning activities that are related to the learning objectives and students’ characteristics, knowledge, skills and experiences.
2. Integrate software development tools that would enable students to experience patterns-based software development in the context of real-world problems. Such tools should enable students to do practical examples and experience how the theory they have learned can be applied in practice.

P. Dillenbourg and M. Specht (Eds.): EC-TEL 2008, LNCS 5192, pp. 155–166, 2008.

3. Include collaborative tools such as discussion forums, chat, and tools for software artifacts exchange. Since software development is intrinsically a team oriented work, students should get used to collaborative style of work as well as learn what constitutes a successful team.
4. Enable seamless access to online repositories of software patterns and communities of practice that will provide students with right-in-time access to the relevant online resources, that is, to the software patterns relevant for the problem at hand.
5. Provide tools for informing teachers about students learning activities, their usage of learning content and other valuable information that could help them improve the learning content and/or the chosen teaching approach.

Even though the above mentioned kinds of tools do exist today, they are not used in an integrated way [3]. Instead, current approaches to learning software patterns are based on individual use of these tools. The major problem with this 'fragmented' approach is in its lack of means for enabling exchange of data about the activities that students performed within individual learning tools and learning artifacts they have produced during those activities. Besides, with such an approach it is very hard to provide support for context-aware learning services and offer personalized learning experience to students.

In this paper, we propose a new approach to a learning environment for software patterns that leverages existing Learning Management Systems (LMSs), domain specific tools for software modeling and relevant repositories of software patterns available online. All these elements connected together establish an integrated learning framework, called DEPTHS (DEsign Patterns Teaching Help System) that supports collaborative learning of software patterns. We propose the use of Learning Object Context Ontology (LOCO) framework [4] as an ontology base for the integration. The framework integrates a number of learning-related ontologies, such as user model ontology, a learning content ontology, and domain ontologies. We leverage the LOCO framework in the following manner: domain ontology is used for representing the domain of software patterns, whereas the learning context ontology is extended to allow for capturing and representation of learning contexts of different kinds of systems and tools that DEPTHS integrates.

This approach promises to be beneficial for all participants in the learning process:

- Teachers would spend less time developing lessons. Instead they would (re)use the existing online resources. DEPTHS would secure the ease of locating relevant online resources for the course the teacher develops.
- DEPTHS will improve students' learning effectiveness and efficiency by recommending relevant resources from online repositories as well as by recommending the most suitable peer(s) to collaborate with.
- Integration of knowledge about all learning related activities performed within any educational tool integrated in this framework should provide solid base for improving the quality of important educational aspects (e.g., adaptation and context-awareness).

The following section presents a typical usage scenario of DEPTHS. Section 3 gives an overview of the LOCO ontological framework as well as the extensions we made to meet the specific needs of DEPTHS. An extensive description of the DEPTHS architecture is given in Section 4, whereas Section 5 focuses on the implementation details. Before concluding the paper, we outline the related work (Section 6).

2 Running Example

It is possible to develop many scenarios for learning software patterns in the DEPTHS framework. Due to the limited size of this paper, we describe only one, which is based on a problem-based learning approach with collaborative learning support. In this scenario (Fig. 1), a teacher defines a specific software problem that has to be solved in a workshop-like manner. Workshop is a peer assessment activity with a huge array of options, such as allowing students to review and assess each other's solutions. The teacher provides an informal description of the problem, a task to be accomplished and a set of learning resources that could help students to solve the problem.

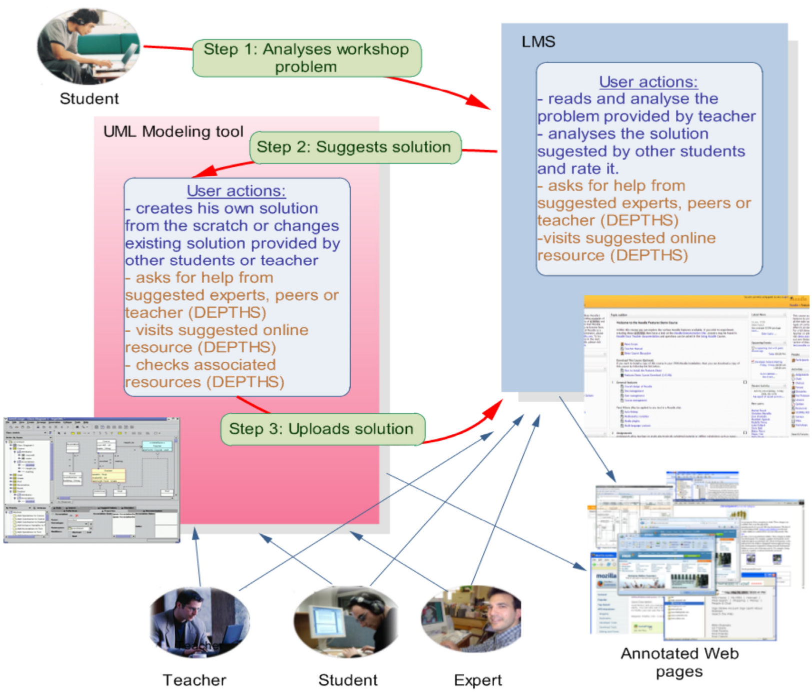

Fig. 1. An example learning scenario with DEPTHS: problem-based learning with collaborative learning support (DEPTHS in parenthesis indicates DEPTHS specific functionalities)

Students are typically supposed to provide a graphical representation of their solutions (i.e., the designed software models). A student can draw his/her own solution from scratch, use some other student's solution, and/or use a partial solution provided by the teacher in the problem's description. If one of the last two options is selected, an appropriate solution (often in the form of a diagram) would be loaded within the student's modeling tool and the tool would keep track of all changes that the student would make and tag them with other color on the diagram. Based on the student's current learning context, DEPTHS would suggest him/her to consult online resources that it estimated as potentially useful for the student's current situation. It would also find and suggest other students, experts and/or teachers that could be contacted in order to get additional support. The system would do this both proactively and on the student's request. As DEPTHS provides seamless integration of all of its tools, the student is able to send a message to or chat with peers regardless what tools of the DEPTHS framework they are using at that moment.

3 Ontological Foundation

DEPTHS is based on the Learning Object Context Ontology (LOCO) ontological framework [4]. LOCO allows one to formally represent the notion of learning context which is defined as a specific learning situation, determined by the learning activity, the learning content, and the student(s) involved. To support DEPTHS, we use ontologies of the LOCO framework to interrelate information about learning objects, learning activities and learners collected from various tools relevant for learning software patterns, as specified in the introduction.

The core part of the LOCO framework is the LOCO-Cite ontology, which comprises a number of classes and properties aimed at formally representing learning context. In addition, this framework integrates a number of learning-related ontologies, such as: User Model ontology, Learning Design ontology, Learning Object Content Structure ontology, Domain ontology [4]. To address the requirements of the DEPTHS framework, we fully adopted the LOCO-Cite ontology, and connected it with the ontology of software patterns domain.

3.1 The LOCO-Cite Ontology

The LOCO-Cite ontology allows for semantic representation of the data about a student's overall interactions with learning content and other students during different learning activities. Based on this data, DEPTHS can perform context-aware retrieval of software patterns resources from online repositories and its own repository of software artifacts (which contains artifacts produced and shared by other students); identify and draw students' attention to the related threads in discussion forums; and identify peers that could help in a specific situation.

Activities are very important part of the learning process in DEPTHS as they lead to realization of learning objectives. Examples of such activities are reading lessons, visiting online pages, participating in a workshop or doing an assignment, solving design problems, quizzing and collaborating with other participants. In the LOCO-Cite ontology, these activities are recognized and grouped as three basic types of activities: reading, doing an assessment, and collaborating. However, we found that the LOCO-Cite ontology does not allow for capturing and representation of some specific types of activities and events typically occurring within software modeling tools. Accordingly, we extended this ontology with a set of classes and properties, to enable collecting data about user interactions in software development tools, in a similar way as described in [3].

3.2 Domain Ontology

Since DEPTHS is devised as an environment for teaching/learning software patterns, it leverages an ontology of software patterns as its domain ontology. DEPTHS uses this ontology to annotate relevant learning resources and extract metadata that is subsequently used for finding resources appropriate for a student's current learning context. In this way, DEPTHS can easily connect diverse kinds of learning resources, and use this links to further improve its context-aware support by being able to mash-up knowledge scattered in different activities.

Rather than developing new design pattern ontology from scratch, we decided to (re)use an existing ontology. Among the available ontologies of the design patterns domain [5],[6],[7],[8], we have chosen the set of ontologies suggested in [8] to serve as the domain ontology of the DEPTHS framework. Comparing these with the other ontologies, we found that they provide a very intuitive and concise way to describe design patterns and patterns collections, and more information on usability knowledge and the contextual factors that impact this knowledge. This approach to pattern representation allows for federating distributed pattern collections.

4 System Architecture

In this section, we present a high-level architecture of our DEPTHS framework. The framework comprises five basic components: a Learning Management System (LMS), a Collaborative Learning Modeling tool, a Teachers' Feedback tool, Online Repositories of Software patterns and a Semantic Management System (Fig. 2). In the rest of the section, each of these components is addressed in turn.

4.1 Learning Management System

Today's LMSs have an extensive set of tools and features aimed at facilitating the learning process (e.g., quiz, assignment, chat room, discussion forum, and glossary). However, they do not fully address the requirements of a comprehensive learning framework such as DEPTHS. One of those requirements is the integration of the usage tracking data from all the systems/tools students use. We address this requirement with the LOCO framework (see Section 3). As most of the existing LMSs use classical databases for data storage, it is necessary to transform the data stored in their databases into semantically enriched data compliant with the LOCO-Cite ontology. The transformed data is stored in the **Repository of Learning Object Contexts (LOCs)** which is fully based on the LOCO-Cite ontology**.**

Apart from the existing collaborative learning support that is usual in most LMSs (such as discussion forums and chat-rooms), we found that it would be very useful if student(s) had a tool for collaborative annotation of learning content (such as tagging, commenting and highlighting). The more the content is annotated, the easier it becomes to later find it and retrieve it. Accordingly, we decided to integrate such a collaborative tool in the LMS that is used in our framework.

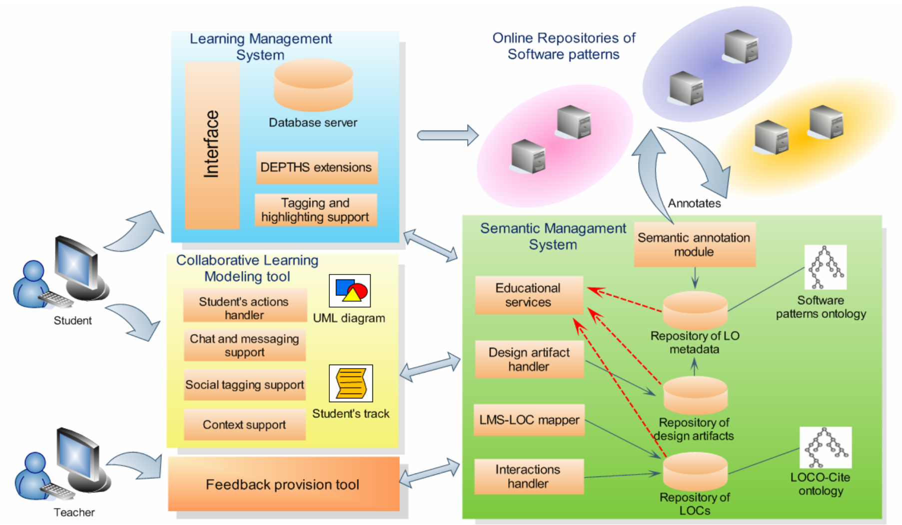

Fig. 2. The architecture of the DEPTHS learning framework

4.2 Collaborative Learning Modeling Tool

We have identified that the framework for software engineering education should necessary have the support for software modeling using diagrams, especially UML (Unified Modeling Language) diagrams. However, most of the existing software modeling tools does not provide all necessary support for collaborative learning. We refer here to the set of features that should be supported by these tools, beside those that they usually include:

- An easy way for presenting a description of the suggested solution.
- Collaborative Tagging support module enabling students to create either social (public) or private annotations of learning content (e.g., publicly available online resources on the Web, design diagrams, and forum messages). That way, a student begins to create a network of content that can be later accessed and searched, for example, through a tag cloud view.
- A chat room and messaging tools that support collaboration with other students even if they are not using the same tool in the given moment.
- Ability to keep track of students' actions during learning sessions (**Student's actions handler**). These tracks are sent to the **Interactions handler** (see section 4.5) which is responsible for integrating them into the **Repository of LOCs** where they are stored for later analysis.
- Context-aware learning. Based on a student's learning context the system should suggest him/her the most suitable learning content, publicly available online resources on the Web, similar problems, or discussion threads that could be useful for the specific problem he/she is facing. This should basically help students to better comprehend relations between the acquired theoretical knowledge and experiences of others with the practical problems at hand.

4.3 Feedback Provisioning Tool

In order to help a teacher to improve the learning experience of his/her students, DEPTHS incorporates a tool that provides teachers with feedback about all kinds of activities their students performed during the learning session. This tool is also built on top of the LOCO framework, so that it has access to the learning context data created in all learning tools used in DEPTHS. The feedback tool provides teachers with contextualized feedback and relevant information about students' learning.

4.4 Online Repositories of Software Patterns

One of the main advantages of this framework is that it leverages existing online learning resources, rather than requiring teachers to create new ones. There are a plenty of such repositories which could be used, such as Yahoo! Design Pattern Library[1], Portland Pattern Repository[2], and Hillside.net Pattern Catalog[3].

DEPTHS leverages the domain ontology to provide both teachers and students with resources from these repositories that are relevant for the current teaching/learning context. In particular, the domain ontology is used for annotating semanticaly the resources available from these repositories and the resulting semantic metadata is stored in the **Repository of LO Metadata** (see Fig. 2). This metadata is used for indentifying the resources relevant for any given learning situation. An additional advantage of DEPTHS is that these resources are made accessible from which ever tool of the DEPTHS framework a teacher or a student is using. Moreover, students are able to tagg and highlight these resource. We believe that these content annotation activities can not only improve some typical activities in learning (e.g. revisiting learning material, personal note taking, and connecting with peers), but also provide valuable data that can quaranty higher quality of context-aware learning.

4.5 Semantic Management System

This module is the integration point of the whole framework. In particular, it leverages the semantic web technologies to support integration of all the above mentioned modules. In order to acomplish this, it uses a set of repositories and a set of software compentents. In particular, it comprises the following three reporitories:

- **Repository of LO metadata** stores semantic metadata about online resources available from online repositories, as well as about internally created content, such as software design diagrams, discussion forum postings and chat messages besides regular lessons used in the courses under study. This metadata consists of topics defined in the software pattern ontology and we refer to it as semantic metadata as it formally defines the semantics of the learning content it is attached to.
- **Repository of design artifacts** keeps students' solutions in different formats: a reusable format, and a format suitable for presentation in the LMS.
- **Repository of LOCs** stores learning objects' context-related data in accordance with the LOCO-Cite ontology.

---

[1] http://developer.yahoo.com/ypatterns/index.php
[2] http://c2.com/ppr/
[3] http://www.hillside.net/patterns/onlinepatterncatalog.htm

In addition, Semantic Management System integrates the following components:

LMS-LOC mapper has the role to transform data from the LMS database of logs of learners's activities into the format compliant with the LOCO-Cite ontology and to store the resulting data in the Repository of LOCs. This data mapping is performed throughout each learning session in order to keep the semantic repository updated (with data about the events occuring durign that session).

Semantic annotation module is used for annotating online repositories of design patterns, as well as, diagrams (created by students) stored in the Repository of design artifacts. This module automatically extracts metadata based on the domain ontology and stores them in the Repository of LO metadata.

Design artifact handler manages the diagrams in the Repository of design artifacts. It takes diagrams from the modeling tool and stores them in the Repository of design artifacts in different formats. It is also responsible for keeping track of different versions of the same diagram.

Educational services provide all necessary support for context-aware learning and are accessible from all tools integrated in the DEPTHS framework. These services are based on Semantic web technologies, and include (but not limited to):

– Finding web resources relevant for the student's current learning context.
– Recommending learning artifacts (discussion posts, chat messages, workshop submission…) related to the current context.
– Finding potential collaborators among experts and peers.

5 Implementation of DEPTHS

In this section, we describe the tools that we are using to implement the proposed framework and argument our decision to use specifically these tools.

5.1 Learning Management System

As the LMS component of the DEPTHS framework (Section 4.1), we have decided to use Moodle[4] LMS for many reasons. First, Moodle is a popular open-source LMS, which requires only hosting costs, and thus provides us with an inexpensive but reliable learning environment. In addition, the open source nature of this system enables us to extend it with Semantic Web technologies. Moodle also has an extensive set of tools and features.

However, one of the most eminent advantages of Moodle that influenced our decision is that it facilitates collaborative work, that is, it is designed under the social constructivist theory. This theory argues for a student-centered environment where learners are able to work independently, reflect on their own work and on the work of other students, while at the same time being connected to a group of learners with whom they can share ideas and reflect on each other's work [10]. As we indicated in the introduction, getting the students involved in the learning process is essential to effective learning of software patterns.

[4] http://moodle.org/

However, the manner in which Moodle stores data about students' interactions with the system is inappropriate for DEPTHS. Rather, DEPTHS requires semantically enhanced interactions data, that is, RDF data stored in a format compliant with the LOCO-Cite ontology. In order to resolve this issue, during the initialization of the system, we do the mapping of the interactions data stored in Moodle's database into the required ontological format and store the resulting RDF data into the Repository of LOCs (see Section 4.5). This task is performed using D2RQ[5] – an open source platform that facilitates the mapping of relational databases into ontological models. This way a lot of valuable data that currently resides in Moodle database are made accessible to DEPTHS in the form of RDF statements. Later, throughout each learning session, DEPTHS uses Sesame[6] Java API to update the semantic repository with data about the events occurring during that session. Apart from updating the Repository of LOCs, DEPTHS uses Sesame API to query this repository in order to retrieve the data required by its educational services (see Section 4.5). As two distinct technologies are used (PHP for Moodle and Java for DEPTHS), we use PHP/Java bridge[7] to provide the connection.

We have decided to integrate OATS (The Open Annotation and Tagging System) [11] in Moodle in order to provide students with a tool to collaboratively create and share knowledge, by using highlights, tags and notes. OATS is an open source tool which was created to further enrich the functionalities provided by an LMS. The aim is to motivate students to enagage more and move beyond passive consumtion of e-learning content towards active production [12]. We have already made an extension of the LOCO-Cite ontology to enable formal representation of events occuring in collaborative content annotation tools such as OATS and we intend to use it to capture and store data about students interactions with OATS into the Repository of LOCs.

This way DEPTHS employs Moodles advantages but also adds some new possibilities provided as DEPTHS's educationsl services, among which the most important are: finding web resources that could be useful in the current learning context, finding relevant internaly produced resources stored in DEPTHS repositories, and finding appropriate peers.

5.2 Semantic Annotation of Learning Content

Among many available tools for content annotation that we have tested, we decided to use the KIM framework[8], a semantic annotation platform, that provides an automatic semantic annotation, indexing and retrieval of documents. Semantic annotation in KIM is based on the provided domain ontology (i.e. the ontology of software patterns, see Section 3.2) that makes KIM aware of the concepts from the software patterns domain. As a result, we use KIM annotation facilities to automatically annotate diverse kinds of learning artifacts with relevant domain topics. This further facilitates

[5] http://www4.wiwiss.fu-berlin.de/bizer/D2RQ/spec/
[6] http://www.openrdf.org/
[7] http://php-java-bridge.sourceforge.net/pjb/
[8] http://www.ontotext.com/kim/index.html

semantic interlinking of diverse kinds of learning artefacts: online resources, lessons from the LMS, students software models, and exchanges messages. Thus, enables us to integrate previously fragmented knowledge artifacts students used or created in learning activities.

5.3 Domain Modeling Tool

ArgoUML is a Computer-Aided Software Engineering (CASE) tool suitable for the analysis and design of object-oriented software systems. It allows for designing all kinds of UML diagrams. Another advantage of ArgoUML is that it supports open software standards which facilitate exchange of UML diagrams among students, as well as presentation of these diagrams in Moodle.

Due to its open-source nature, ArgoUML can be extended to enable capturing of user interaction data and storing that data in the common ontological format of the DEPTHS framework (i.e. the LOCO-Cite ontology). We are using Sesame Java API to extend ArgoUML, so that it continually updates the Repository of LOCs with data about events generated during students' interactions with the tool. Subsequently, we are going to extend ArgoUML, so that it can make use of the DEPTHS's educational services, such as finding solutions to the similar problems suggested by other students or finding appropriate online resource about design patterns that could be used in the student' current context.

5.4 Feedback Provision Tool

Finally, we have decided to integrate LOCO-Analyst in the DEPTHS framework. This tool provides teachers with contextualized feedback regarding all kinds of activities their students performed during a specific period of time [4]. It is built on top of the LOCO framework so we can easily use it in this framework without any intervention on it.

6 Related Work

The framework proposed in this paper is related to two favored research fields: collaborative learning of software engineering and the Semantic web. Even though extensive work has been done in both research fields, to the best of our knowledge there were very few attempts in developing collaborative learning environments through integration of existing tools based on the semantic web technologies.

The approach proposed in [13] presents an intelligent tutoring system, called COLLECT-UML, the goal of which is to support the acquisition of both problem-solving skills and collaboration skills. In this environment, students construct UML class diagrams that satisfy a given set of requirements. COLLECT-UML supports collaborative learning and provides feedback on both collaboration issues and task-oriented issues. Our framework provides similar approach to the learning process, that is, students learn through the practical problem-based examples in collaboration with other students. However, our framework offers higher learning potential as it provides access to the relevant learning resources and facilitates context-aware learning.

Many authors have indicated the great advantages that Semantic web technologies can bring to education. For example, Devedzic [14] stated that for semantic interoperability of educational content and applications on the Web, it is necessary to root them in the technologies of the Semantic Web. He has proposed educational servers which exploit learning technology standards, ontologies and pedagogical agents to support interaction between clients (authors and students) and servers (hosting educational content and services). A similar approach is described in [15] where the author suggests the use of semantic web technologies for representing knowledge about resources, learners and services. The author in this work suggests a service-based architecture for establishing personalized e-learning, where personalization functionality is provided by various web-services. While strongly agree with this approach, we are going a step further. In DEPTHS we provide a domain specific tool which is necessary, especially in software and other technical engineering education.

In [3], the authors suggested the approach similar to the one presented in this work. They have developed MICE – a learner-centered platform for regulating learners' programming styles when studying a programming language using an integrated development environment. It also integrates an LMS and a set of tools for communication and collaboration among users. Even though MICE follows a similar approach to integration of existing tools, it still lacks access to the online resources that is available in our framework. Besides our framework promises additional support for collaborative learning as it offers social tagging support.

7 Conclusion

This paper has introduced DEPTHS, a collaborative learning framework that we are developing to better support education in the domain of software patterns. We have argued for the integration of existing, proven learning systems and tools in this framework in order to provide an effective collaborative learning environment. Such a framework requires a flexible underlying ontology-based model to support such integration. We have found that the LOCO ontological framework exactly addresses this requirement. Currently, we are using two kinds of ontologies of the LOCO framework: learning object context ontology (the LOCO-Cite ontology) and domain ontology (an ontology of software pattern).

DEPTHS can improve student's collaboration work by recommending resources that are related to the goal the student is currently working on, by providing tools for collaboration in solving practical exercises and by suggesting peers to collaborate with. By allowing for knowledge sharing among different educational tools and online repositories, we provide students with a context-aware learning environment. We believe that this is a promising direction that will contribute to further improvements in software engineers' education.

Our present and future work is primarily focused on working further on the implementation of the DEPTHS framework. So far, we have been primarily working on data collection and integration and the next step is to develop educational services (explained in Section 4.5) that will leverage that data to provide students with enhanced learning experience (such as the one described in the learning scenario of Section 2).

References

1. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press, New York (1977)
2. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995)
3. Jovanović, J., Rao, S., Gašević, D., Devedžić, V., Hatala, M.: An Ontological Framework for Educational Feedback. In: Proc. of the 5th Int'l Workshop on Ontologies and Semantic Web for Intelligent Distributed Educational Systems, USA, pp. 54–64 (2007)
4. Jovanović, J., Gašević, D., Brooks, C., Devedžić, V., Hatala, M., Eap, T., Richards, G.: Using Semantic Web Technologies for the Analysis of Learning Content. IEEE Internet Computing 11(5) (2007)
5. Dietrich, J., Elgar, C.: A formal Description of Design Patterns using OWL. In: Proc. of ASWEC, IEEE Comp. Soc. (2005)
6. Montero, S., Diaz, P., Aedo, I.: Formalization of web design patterns using ontologies. In: Proc. of 1st Int'l Atlantic Web Intell. Conf. (AWIC), Spain, pp. 179–188 (2003)
7. Kampffmeyer, H., Zschaler, S.: Finding the Pattern You Need: The Design Pattern Intent Ontology. In: Engels, G., Opdyke, B., Schmidt, D.C., Weil, F. (eds.) MODELS 2007. LNCS, vol. 4735, pp. 211–225. Springer, Heidelberg (2007)
8. Henninger, S.: A Framework for Flexible and Executable Usability Patterns Standards. In: 31st IEEE Software Engineering Workshop (SEW-31), USA, pp. 23–34 (2007)
9. Coplien, J.O.: Software Patterns. SIGS Books, New York (1996)
10. Dougiamas, M.: A journey into constructivism (1998), `http://dougiamas.com/writing/constructivism.html`
11. Bateman, S., Farzan, R., Brusilovsky, P., McCalla, G.: OATS: The Open Annotation and Tagging System. In: The Proc. of the 3rd Annual Int'l Scientific Conf. of the LO Repository Research Network, Montreal (2006)
12. Fischer, G., Ostwald, J.: Transcending the Information Given: Designing learning Environments for Informed Participation. In: The Proc. of ICCE Int'l Conf. on Computers in Education, New Zealand (2002)
13. Baghaei, N., Mitrovic, A., Irwin, W.: Supporting collaborative learning and problem-solving in a constraint-based CSCL environment for UML class diagrams. Int'l Journal of CSCL 2(2-3), 150–190 (2007)
14. Devedžić, V.: Next-generation Web-based Education. Int'l Journal for Continuing Engineering Education and Life-long Learning 11(1/2), 232–247 (2003)
15. Dolog, P., Henze, N., Nejdl, W., Sintek, M.: Personalization in Distributed e-Learning Environments. In: Inter'l WWW Conf., pp. 170–179. ACM Press, New York (2004)