A Semantic-rich Framework for Learning Software Patterns

Zoran Jeremić[1], Jelena Jovanović[1], Dragan Gašević[2]
[1]*FON-School of Business Administration, University of Belgrade, Serbia*
[2]*School of Computing and Information Systems, Athabasca University, Canada*
jeremycod@yahoo.com, jeljov@gmail.com, 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.

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 the knowledge about successful solutions for recurring problems that is also known as software patterns [1]. Each software pattern is itself a piece of potentially complex knowledge that requires a degree of mastery to apply to a specific problem context. Besides, different pattern collections (groups of related patterns) use different pattern forms for pattern representation, such as the Gang of Four form or Alexandrian form [2]. Students need to be familiar with all these forms in order to be able to use a specific pattern collection. All the above mentioned imply that teaching and learning software patterns is a complex process that needs to be supported by an adequate learning platform. Aiming to provide such a learning platform, we have identified major requirements for quality learning in the domain of software patterns:

1. Enabling students to learn at the pace and in a place that best suits them. Students should also be provided with content and a variety of course activities that are directly related to learning objectives and students' knowledge, skills and experiences.
2. Software development tools that would enable students to experience patterns-based software development in the context of real-world problems. Therefore, these tools should enable students to do practical examples and experience how the theory they have learned can be applied in practice.
3. Collaborative tools such as discussion forums, chat, and systems 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 makes a successful team.
4. Right-in-time access to the online repositories of software patterns and communities of practice.
5. Tools for providing teachers with feedback about students learning activities and their usage of learning content thus enabling them to improve the learning content and/or chosen teaching approach.

Although the abovementioned 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 integrating data about the activities that students performed within individual learning tools and the learning artifacts they have produced during those activities. Therefore 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 suggest a new approach for learning design patterns that leverages existing Learning Management Systems (LMSs), domain specific tools for software modeling and relevant repositories of software patterns available online in order to provide a complex learning framework, called DEPTHS (DEsign Patterns Teaching Help System) that supports collaborative learning of software patterns.

2. Running example

DEPTHS framework can facilitate many scenarios for learning software patterns. Due to the space limit, we describe only one, in which 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 (e.g., it allows students to review and assess each other's solutions). The

978-0-7695-3167-0/08 $25.00 © 2008 IEEE
DOI 10.1109/ICALT.2008.258

teacher provides an informal description of the problem, a typical scenario, a task that has to be accomplished and a set of learning resources that could help students to solve the problem.

Students are typically supposed to provide a graphical representation of their solution (i.e., software model). A student can draw his/her own solution from scratch, use some other student's solution, and/or partial solution provided by the teacher in the problem's description. If one of the last two options is selected, an appropriate solution is loaded within the student's modeling tool and the tool will keep a track about all changes that the student would make and mark them with other color on the diagram. Based on the student's current learning context, DEPTHS will suggest him/her to consult online resources (e.g., parts of the course content, pattern repositories, best practice portals, and other students' artifacts produced while collaborating such as discussion forum messages) that it estimated as potentially useful for the student's current situation. It will also find and suggest peers that could be contacted to get additional support. As DEPTHS provides seamless integration of all its tools, the student can send a message to or chat with peers regardless what tools they are currently using.

3. Ontological foundation

DEPTHS is based on the Learning Object Context Ontologies (LOCO) ontological framework [4]. LOCO allows one to formally represent the notion of learning object context which is defined as a specific learning situation, determined by the learning activity, the learning content, and the student(s) involved. Accordingly, LOCO integrates a number of learning-related ontologies such as user model, content structure, and domain ontologies. We leverage LOCO in as follows: we use a domain ontology to represent the domain of software patterns and the extended learning context ontology to capture context specific for software modeling tools.

The core part of LOCO is the LOCO-Cite ontology which comprises a number of classes and properties aimed at formally representing learning object context. However, we found that this ontology does not allow for capturing and representation of some specific events typically occurring within software modeling tools. Accordingly, we decided to extend this ontology.

DEPTHS uses an ontology of software patterns as its domain ontology. This ontology is used for annotating relevant resources and extracting metadata that are subsequently used for context-aware recommendation of learning resources. It is also used for annotating the products of learning activities, such as chat messages. In this way, DEPTHS can connect these products with learning resources, and use this information to provide a context-aware support. Rather than developing new domain ontology from scratch, we use an existing one. We experiment with a set of ontologies suggested in [5], but other software pattern ontologies may be used instead of or with this one.

4. System architecture

In this section, we present a high-level architecture of our DEPTHS framework. The framework comprises five basic components: Learning Management System, Collaborative Learning Modeling tool, Teachers' Feedback tool, Online Repositories of Software patterns and Semantic Management System (Figure 2).

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 enable integration of the usage tracking data from different systems/tools students use, which is required for a comprehensive learning framework such as DEPTHS. We address this requirement with LOCO (as explained in Sect. 3). In our approach, data about students' interactions within LMSs are transformed into semantically enriched data compliant with LOCO-Cite and stored in the **Repository of Learning Object Contexts (LOCs).**

We have also decided to extend the collaborative learning support that is usual in most LMSs, with a tool for collaborative tagging and highlighting.

4.2. Collaborative Learning Modeling Tool

We have identified that the framework for software engineers' education should necessary have the support for modeling diagrams, especially UML. We refer here to the set of features that should be supported by these modeling tools, beside those that they usually include:

- An easy way for presenting a description of the suggested solution.
- Social Tagging support module enabling students to create either social or private annotations of learning content (online resources, design diagrams, forum messages). That way, a student begins to create a network of content that can be later accessed and searched through a tag cloud view.
- A chat room and messaging tools that support collaboration with other students even if they are currently using these tools from LMS.
- Ability to keep track of students' actions during learning sessions (**Student's actions handler**). These tracks are sent to the **Repository of LOCs** where they are stored for later analysis.

As a support for context-aware learning, the modeling tool recommends students the most suitable online resources or discussion threads that could be useful for the specific problem (s)he is facing.

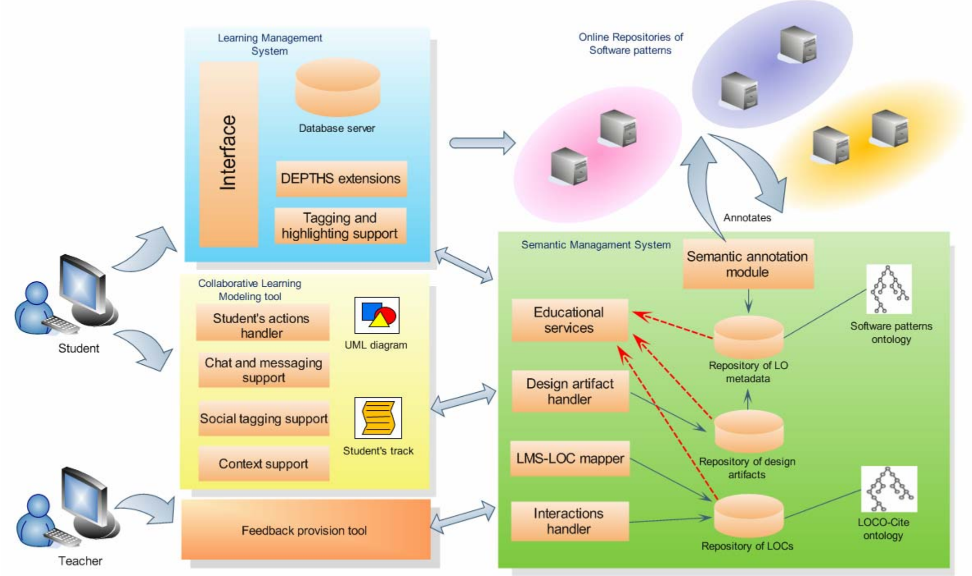
Learning Management System
Interface
Database server
DEPTHS extensions
Tagging and highlighting support
Online Repositories of Software patterns
Annotates
Student
Collaborative Learning Modeling tool
Student's actions handler
UML diagram
Chat and messaging support
Social tagging support
Student's track
Context support
Semantic Managament System
Semantic annotation module
Educational services
Software patterns ontology
Repository of LO metadata
Design artifact handler
Repository of design artifacts
LMS-LOC mapper
LOCO-Cite ontology
Interactions handler
Repository of LOCs
Teacher
Feedback provision tool

Figure 2. The architecture of the DEPTHS learning framework

4.3. Feedback provisioning tool

To help teacher to improve the learning experience of his/her students, DEPTHS incorporates a tool that provides teachers with feedback about all kinds of activities the students performed during the learning session. This tool is also built on top of LOCO, so it can have access to the learning object context data created in all learning tools used in DEPTHS.

4.4. Online Repositories of Software patterns

DEPTHS offers means for leveraging existing learning resources, rather than requiring teachers to create new ones from scratch. There are a plenty of such repositories, such as Yahoo! Design Pattern Library. This is done By leveraging the domain ontology of software patterns.

4.5. Semantic Management System

This module is the integration point of the whole framework. It leverages semantic web technologies to support integration of the abovementioned modules. To acomplish this, it uses a set of repositories and a set of software compentents. In particular, it comprises the following reporitories: i) **Repository of LO metadata** stores metadata extracted based on the software pattern's domain ontology from online repositories as well as from internally created content, such as design diagrams, discussion forum postings, or chat messages. ii) **Repository of design artifacts** keeps students' solution in different formats appropriate for their latter reuse, as well as, their presentation in LMS system. iii) **Repository of LOCs** stores learning context data compliant with the LOCO-Cite ontology. The framework also comprises a set of components (Fig. 2) that provides interaction with repositories and offers a set of services to other modules of DEPTHS.

5. Conclusion and Related Work

This paper proposed the architecture of the DEPTHS framework, a collaborative learning environment designed to facilitate collaborative learning of software patterns. This framework enables the interoperability among different educational tools and online repositories, which employs an ontological framework to form an effective learning environment. DEPTHS can improve students' collaborative 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.

We have started implementing DEPTHS by leveraging open-source solutions and extending them with Semantic web technologies. So far, we have integrated the Moodle LMS in our framework and we are currently working on the integration of ArgoUML software modeling tool and OATS collaborative tagging and highlighting tool.

6. References

[1] E. Gamma, et al., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.

[2] L.Rising, *The Patterns Handbook: Tech., Strategies, and Applications*, Cambridge U. Press, NY, 1998.

[3] J. Jovanovic, et al., "Ontological Framework for Educational Feedback", *In Proc. of the 5th Int'l Workshop on Ontol. and Sem. Web for E-Learning*, 2007, pp. 54-64.

[4] J. Jovanović, et al, "Using Semantic Web Technologies for the Analysis of Learning Content," IEEE Internet Computing, 11(5), 2007, pp. 45-53.

[5] S. Henninger, "A Framework for Flexible and Executable Usability Patterns Standards", In Proc. of the 31st IEEE Soft. Eng., Workshop, 2007, pp. 23-34.