Semantic Web Technologies for the Integration of Learning Tools and Context-Aware Educational Services

Zoran Jeremić[1], Jelena Jovanović[1], and 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. One of the main software engineers' competencies, solving software problems, is most effectively acquired through an active examination of learning resources and work on real-world examples in small development teams. This obviously indicates a need for an integration of several existing learning tools and systems in a common collaborative learning environment, as well as advanced educational services that provide students with right in time advice about learning resources and possible collaboration partners. In this paper, we present how we developed and applied a common ontological foundation for the integration of different existing learning tools and systems in a common learning environment called DEPTHS (Design Patterns Teaching Help System). In addition, we present a set of educational services that leverages semantic rich representation of learning resources and students' interaction data to recommend resource relevant for students' current learning context.

Keywords: Semantic web, ontologies, collaborative learning, project-based learning, software patterns, context-awareness.

1 Introduction

The major concern of today's software engineering education is to provide students with the skills necessary to integrate theory and practice; to have them recognize the importance of modeling and appreciate the value of a good software design; and to provide them with ability to acquire special domain knowledge beyond the computing discipline for the purposes of supporting software development in specific domain areas. 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 design patterns (DPs) [1].

Apart from learning individual DPs and the principles behind them, students should learn how to understand and apply patterns they have not seen before, how to integrate different DPs, and how to use this knowledge in real-life situations.

This indicates a rising need for the social constructivist approach in software engineering education. In particular, an active learning paradigm is needed which recognizes that student activity is critical to the learning process. The basic philosophy of active learning paradigm [2] is to foster deep understanding of subject matter by

A. Bernstein et al. (Eds.): ISWC 2009, LNCS 5823, pp. 860–875, 2009.

engaging students in learning activities, not letting them be passive recipients of knowledge. Moreover, the students are involved in the knowledge construction and sharing through social interactions in the given learning context.

Following this paradigm, we identified several requirements that a learning environment for software DPs needs to meet [3]:

1. Enable students to learn at the pace and in a place that best suits them.
2. Integrate software development tools that would enable students to experience patterns-based software development in the context of real-world problems.
3. Include collaborative tools such as discussion forums, chat, and tools for software artifacts exchange.
4. Enable seamless access to online repositories of software DPs and communities of practice that will provide students with right-in-time access to the relevant online resources, that is, to the resources on software DPs 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 [4]. 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 describe an integrated learning environment for software DPs called DEPTHS (Design Patterns Teaching Help System) [3]. DEPTHS integrates an existing Learning Management System (LMS), a software modeling tool, diverse collaboration tools and relevant online repositories of software DPs. The integration of these different learning systems and tools into a common learning environment is achieved by using the Semantic Web technologies, ontologies in particular. Specifically, ontologies enabled us to formally represent and merge data about students interactions with the systems and tools integrated in DEPTHS. On top of that data, we have built context-aware educational services that are available throughout the DEPTHS environment. These services enrich and foster learning processes in DEPTHS in two main ways:

- recommendation of appropriate learning content (i.e., Web page(s), lessons or discussion forum threads describing software DP). We can recommend fine-grained learning resources, and make the recommendations aware of the recognized learning needs.
- fostering informal learning activities by bringing together students and experts that are dealing with the same software problem or have experience in solving similar problems.

2 Scenario of Use

Effective learning of software DPs requires a constructive approach to be applied in the teaching process. It is very important that students experience software development and the application of DPs on real-world examples, in order to develop a deep understanding of the basic principles behind them and to learn how to apply them in different situations. Having this in mind, we have explored a number of theories and research fields in the area of project-based and computer supported collaborative learning. Based on the guidelines for teaching software engineering to students [5] [6] and our own teaching experience, we have identified the following three as the most important for teaching/learning software DPs: Learning through Design, Project-based learning (PBL) and Engagement theory. More details on pedagogical aspects of this work could be found at [7]. In what follows, we present a usage scenario which illustrates how these theories are applied for learning software DPs in DEPTHS.

A typical scenario for learning software DPs with DEPTHS assumes a project-based learning approach with collaborative learning support (Fig. 1). In particular, a teacher defines a specific software design problem that has to be solved in a workshop-like manner by performing several predefined tasks: brainstorming, creating and submitting solutions, evaluating solutions etc.

First, a student is asked to present his ideas about possible ways for solving the problem under study and to discuss and rate his peers' ideas. In order to get the required information for performing this task, he searches online repositories about software DPs and the related course content. DEPTHS makes this search more effective by providing semantically-enabled context-aware learning services for finding related online and internally produced resources (Fig. 1B). Moreover, to get some initial directions on the performing task, the student uses semantically-enabled peers finding service (Fig. 1A) to find people who have shared interests and are engaged in similar problems. As we explain in the following section, both kinds of services are enabled by leveraging formally represented semantics of the learning context and learning resources (both online resources and those internally produced). Afterwards, the student has to find associations between the gained knowledge and the problem that has to be solved and to propose potential solution strategies. Later, consultations are directed at confirming the idea and refining it to accommodate criticisms.

Previous works on similar problems could be useful for students as they give them opportunities to learn from positive examples; and provide them with new facts, information, and an idea how to apply the same approach (design patterns) in a similar situation. Moreover, exploring previous works provokes critical thinking as it helps the student think about alternatives along with their advantages and disadvantages. DEPTHS context-aware learning services for discovery of relevant learning resources (both external and internal) greatly facilitate this task. These services are powered by semantic annotations of learning resources: ontologies enable capturing and formal representation of the semantics of the content of those resources, as well as the context of their creation and usage (see Section 3).

Having acquired the required knowledge, students should complete the deliverable using the software modeling tool. This learning activity requires students to externalize their knowledge, to analyze possible solutions and to provide a design rationale.

After completing the project, students are asked to evaluate their own project, as well as to perform evaluation of each other's work. Students reflect critically on their own and others' contributions, and acquire knowledge about other possible solutions. This helps them recognize possible improvements in their own solutions. DEPTHS uses ontologies to capture the semantic of the students' evaluations, so that they can be used for recommendations as well as feedback provisioning.

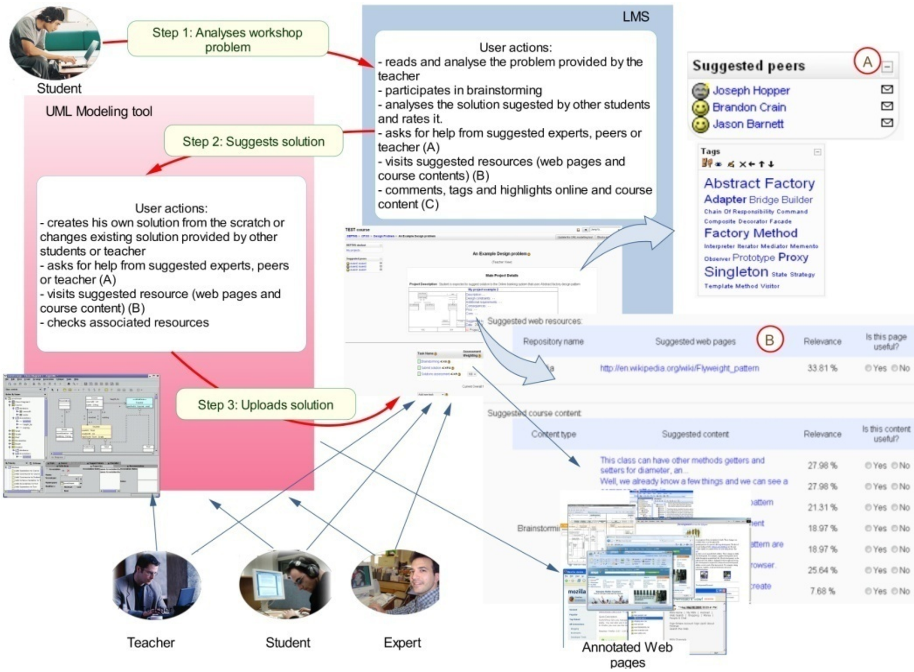

Fig. 1. An example learning scenario with DEPTHS: problem-based learning with collaborative learning support

All students' projects are published and publicly available; they are stored together with contextual semantic-rich metadata which facilitates their discovery and reuse. Students may be anxious that their work will be so visible, but it does seem to push them along to polish their projects. Moreover, students can learn from each other as portions of their projects became available before the final due date.

3 The DEPTHS Learning Environment

DEPTHS integrates existing, proven learning systems, tools and services in order to provide an effective collaborative environment for teaching and learning software DPs (Fig. 2). In particular, DEPTHS currently integrates an LMS (Fig. 2A), a software modeling tool (Fig. 2B), a feedback provisioning tool for teachers (Fig. 2C), a collaborative annotation tool (Fig. 2D), and online repositories of software patterns

(Fig. 2E). This integration is achieved through a flexible underlying ontology-based framework called LOCO (Fig. 2F).

3.1 The Ontological Foundation of DEPTHS

LOCO (Learning Object Context Ontologies) [4] is a generic framework capable of formally representing all particularities of the given learning context: the learning activity, the learning content that was used or produced, and the student(s) involved. Accordingly, the framework integrates a number of learning-related ontologies, such as learning context ontology, a user model ontology, and domain ontologies. These ontologies allow one to formally represent all the details of any given learning context, thus preserving its semantics in machine interpretable format and allowing for development of context-aware learning services. The LOCO ontologies are developed by following the Linked Data best practices (http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/). In particular, linkages were established with well-known Web ontologies such as the Dublin Core vocabulary, FOAF (Friend-Of-A-Friend, http://xmlns.com/foaf/0.1), and SIOC (Semantically Interlinked Online Communities, http://sioc-project.org). All the ontologies of this framework are publicly accessible (http://iis.fon.rs/LOCO-Analyst/loco.html).

DEPTHS currently makes use of two ontologies of this framework Learning Context ontology and domain ontology, that is used for representing the domain of software patterns. The Learning Context 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 resources on software DPs from online repositories and its own repository of software artifacts (which may also contain artifacts produced by other students and shared by the community). It can also identify and draw students' attention to the related threads in discussion forums, as well as identify peers that could help in the given learning situation (i.e., learning context). This ontology was extended to allow for an unambiguous representation of learning contexts specific to the systems, tools and services that DEPTHS integrates (the ontologies are available at the project's website: www.learningdesignpatterns.org).

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 Learning Context ontology, some of these activities are recognized and grouped as three basic types of activities: reading, doing an assessment and collaborating. However, some specific types of activities and events typically occurring within software modeling tools were not supported by the formalisms of the LOCO's Learning Context ontology. Accordingly, we extended this ontology with a set of classes and properties that enable collecting data about user's activities in software development tools, namely Brainstorming, Submitting and Assessing.

Another important concept determining any particular learning context is content item, which represents any kind of content available in a learning environment (e.g. a lesson, course, question, discussion forum post). We have extended the *ContentItem* class with three concepts relevant for learning software DPs, namely, *DesignProblem*, *Task* and *Diagram* (Fig. 3). The *DesignProblem* class identifies a software problem defined by a teacher and presents the basis for project-based learning in the DEPTHS environment. Students are expected to suggest their own solutions of this problem by producing a UML diagram(s) as a graphical representation of the solution. To allow for formally representing students software models, we have introduced the Diagram class as a subclass of *ContentItem*. The *Task* class represents a task that a teacher can define in the context of solving a specific software design problem. We have modeled them as subclasses of the *Task* class (Brainstorm, Submission and Assessment).

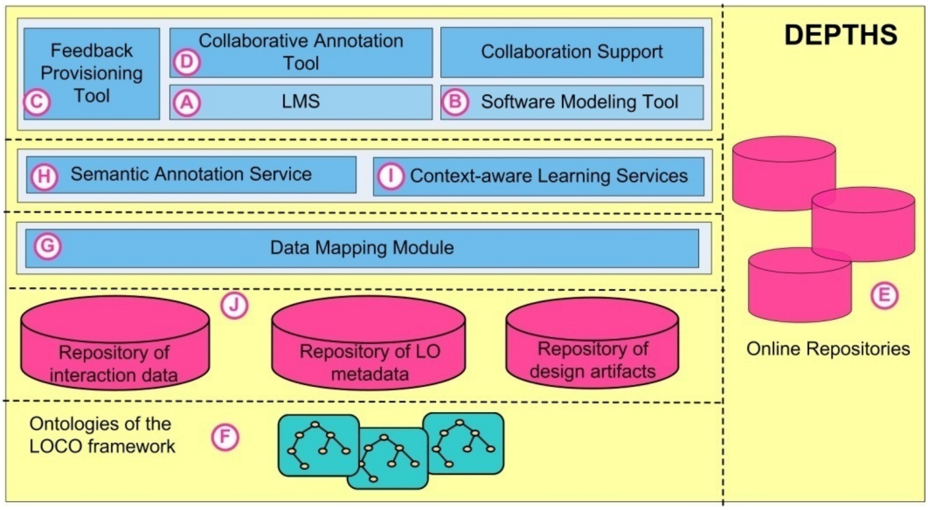

Fig. 2. DEPTHS architecture

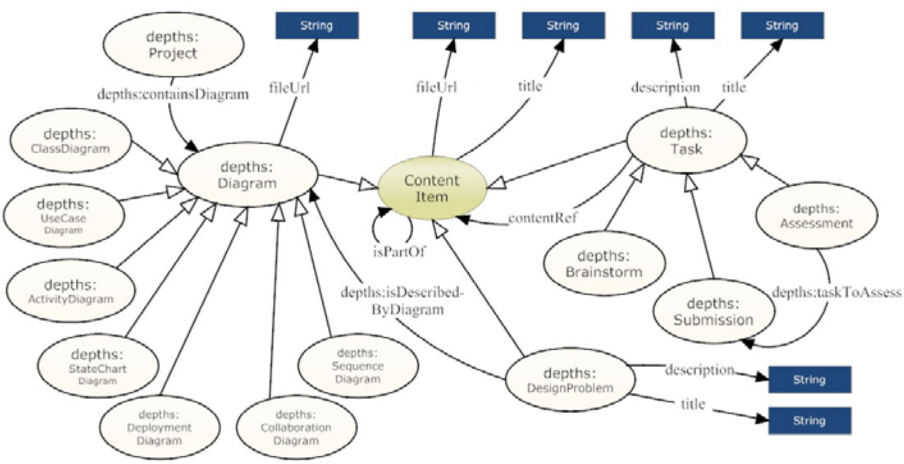

Fig. 3. The extension of the Learning Context ontology for the DEPTHS environment

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 semantically relevant online resources and extract metadata that is subsequently used for finding resources appropriate for a student's current learning context. It also annotates semantically the products of learning activities, such as chat messages or discussion posts. By leveraging the semantic annotations, DEPTHS can easily connect the products of learning activities with online learning resources, and use this information 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 for the design patterns domain [8][9][10][11], we have chosen the set of ontologies suggested in [11] 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 DPs and patterns collections, and to offer more information on usability knowledge and the contextual factors that impact this knowledge domain. This approach to pattern representation has the ability to federate distributed pattern collections. These ontologies include a set of pattern forms (e.g., Coplien Form [12], Gang of Four Form [1]) arranged in an inheritance hierarchy which can be easy extended with additional pattern forms (Fig. 4).

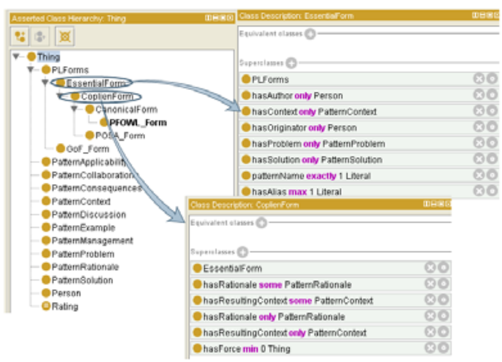

Fig. 4. Domain ontology for describing software design patterns

3.2 The Architecture of DEPTHS

The LOCO ontologies are used as the basis for the storage and exchange of data among DEPTHS components. In particular, these ontologies underlie two DEPTHS repositories (Fig. 2J):

Repository of interaction data stores data about students' interaction with learning content as well as their mutual interactions in the DEPTHS learning environment. The interaction data are stored in the form of RDF triples compliant with the extended Learning Context ontology of the LOCO framework (e.g., {ContentItem} loco:hasUserEvaluation {UserNote}).

Repository of LO metadata stores semantic metadata about all kinds of learning objects (LO) used in the courses under study. This metadata formally define the semantics of the learning content the metadata is attached to. They are stored as RDF triples compliant with the extended Learning Context ontology and the domain ontology of software DPs (e.g., {ContentItem} loco:hasDomainTopic {dp:DesignPattern}).

DEPTHS also includes the *Repository of design artifacts* which uses open standard formats to store software artifacts created by students. The students' artifacts are stored in two formats: XML Metadata Interchange (XMI) and Scalable Vector Graphic (SVG). The former facilitates storing UML diagrams in the format suitable for later reuse in any software modeling tool, whereas the latter is suitable for content presentation in a Web browser.

Since different learning systems, tools and services use different formats for representing and storing interaction data, DEPTHS integrates *Data Mapping Module* (Fig. 2G) which performs the mapping of those native data formats (e.g., the exchanged chat messages stored within the LMS's database, using a proprietary database schema) into RDF triples compliant with the LOCO's Learning Context ontology (e.g., {ChatMessage} loco:sentBy {User}). The resulting (RDF) data is stored in the *Repository of interaction data*. Data mapping is performed as a two step process: the initial mapping, that is performed during the initialization of the system; and the real-time mapping that is performed throughout each learning session in order to keep the semantic repository updated (with data about the interactions occurring during that session). The initial mapping is performed using D2RQ (http://www4.wiwiss.fu-berlin.de/bizer/D2RQ/spec/) – an open source platform that facilitates the mapping of relational databases into ontological models. This way a lot of valuable data that resided in the LMS' database prior to the DEPTHS initialization are made accessible to DEPTHS in the form of RDF statements. Real-time mapping is based on events in the DEPTHS environment (e.g., posting a message in a discussion forum thread, reading chat message). DEPTHS uses Sesame API (http://www.openrdf.org/) to create RDF statements (compliant with the LOCO's Learning Context ontology) for each event and stores them in the *Repository of interaction data*.

3.3 Educational Services in DEPTHS

The next layer in the DEPTHS architecture consists of learning support services, namely Semantic Annotation and Indexing Service and Context-aware Learning Services.

Semantic Annotation and Indexing Service (Fig. 2H) is used for the semantic annotation and indexing of documents from publicly accessible repositories of DPs, as well as internal content created by students and teachers (e.g., exchanged messages, and assignments). Since all these resources are used by Context-aware Learning Services to help students in specific learning contexts, DEPTHS needs additional information about them in order to use them in an appropriate way. Specifically, the most important information for DEPTHS regarding these resources is what each specific resource is about and how relevant it is for a specific DP. However, this information is not directly available. Having analyzed many online repositories of DPs (e.g., Portland Pattern Repository, http://c2.com/ppr/; Hilside patterns, http://hillside.net/patterns/; etc.), we found that the majority of them contains mass of documents, but only a few of those documents describe a specific DP. In addition, most of relevant documents describe in detail exactly one DP (we refer to it as *dominant pattern* throughout the section), but also mention many other DPs (e.g., in the form of links to other documents and related DPs). Moreover, one particular DP is often discussed in many different documents. In order to make this crowd of documents useful for learning purposes, DEPTHS has to analyze how relevant each document is for learning a specific DP, that is, it has to perform semantic annotation and indexing of these documents.

Having tested many of the available tools for semantic annotation, we decided to use the KIM framework (http://www.ontotext.com/kim/index.html), that provides APIs for automatic semantic annotation of documents. Semantic annotation in KIM is based on the PROTON (http://proton.semanticweb.org/) ontology which we have extended with the ontology of software patterns (see Section 3.1) in order to make KIM aware of the concepts from the domain of software DPs. In that way, we were able to use KIM annotation facilities to automatically annotate documents from an online repository. To make documents from an online repository available to KIM for processing, DEPTHS uses a Web crawler that traverses through the structure of the online repository, collects URLs of all its documents and sends them to the KIM annotation facilities.

Semantic annotation of the internally produced content (e.g., chat messages, forum messages and ideas) is performed in the similar way, immediately after the user creates a content unit (i.e., following the event of submitting a new content unit to the system). Additionally, each recognized term (DP label) is changed to the hyperlink, used for launching web-recourse and internal content finding services.

Based on the semantic (meta)data generated using KIM, DEPTHS is aware of the DPs mentioned in each document. However, in order to find out what the most relevant DP for a given document is, and what the most relevant document is for a specific DP in the whole corpus of documents, additional indexing is performed.

The index data for each document contains two values: the dominant DP (i.e., concept from the DP ontology) and its accompanying relevance value. To find the dominant software DP in each document (i.e., what the document is about), DEPTHS uses the term frequency-inverse document frequency (TF-IDF) [13], which is a proven statistical measure used in information retrieval to evaluate how important a word is to a document in a collection. DEPTHS uses a document's semantic annotations to create a collection of all software DPs mentioned in that document. Subsequently, it checks how relevant the document is for each DP found in it in order to find the dominant DP. If the TF-IDF value for the dominant DP is lower than the predefined threshold, DEPTHS eliminates this document as not enough relevant for any specific DP. As the result of this process, we got a set of documents, each of which describes one DP. Additionally, for each of the indexed documents (i.e., those whose dominant DP pass the threshold), we also capture the relevance value. The relevance value is computed as a cosine similarity between the TF-IDF value of a document's dominant DP (concept) and the vector of the TF-IDF values of the DPs (concepts) discovered while the document was semantically annotated. Thus, we can use this relevance value in the process of ranking. That is, the data about what each document is about (i.e., concept of the dominant pattern) and how relevant it is for a specific DP (i.e., relevance value), are stored in the *Repository of LO metadata*, and used later by the Web resource finding service for the recommendation of Web resources.

Context-aware learning services. (Fig. 2I) are accessible to all systems and tools integrated in the DEPTHS environment and are exposed to end users (students) as context-aware learning features. They include (but are not limited to):

Web resource finding. Based on the metadata provided by the Semantic Annotation and Indexing Service, this service generates a list of recommended Web resources from publicly accessible repositories of software DPs. The service gets activated when a student selects a hyperlink to a domain topic (i.e., a topic related to software DPs) available within different kinds of learning content (e.g., lessons, chat messages and project descriptions). These hyperlinks result from the semantic annotation of the learning content done by the Semantic Annotation and Indexing Service. Afterward, the service sends a query to the *Repository of LO metadata* to extracts information about all Web resources that are relevant for the desired domain topic (i.e., DP), as well as their relevance values. Additionally, this service makes use of students estimations of the resource's relevance for the current DP (each time a student visits a suggested Web resource he is asked to rate its relevance for the given DP). Students' positive and negative ratings affect the resource's overall rating according to the influence factor (value between 0 and 1) defined by the teacher or the system administrator.

Discovery of relevant internally produced resources. This service suggests internally created resources (e.g., discussion threads, brainstorming notes, and project description) that could be useful for a student to solve a problem at hand in the given learning context. The computation of relevance is done in a similar manner to the one above described for external, Web resources.

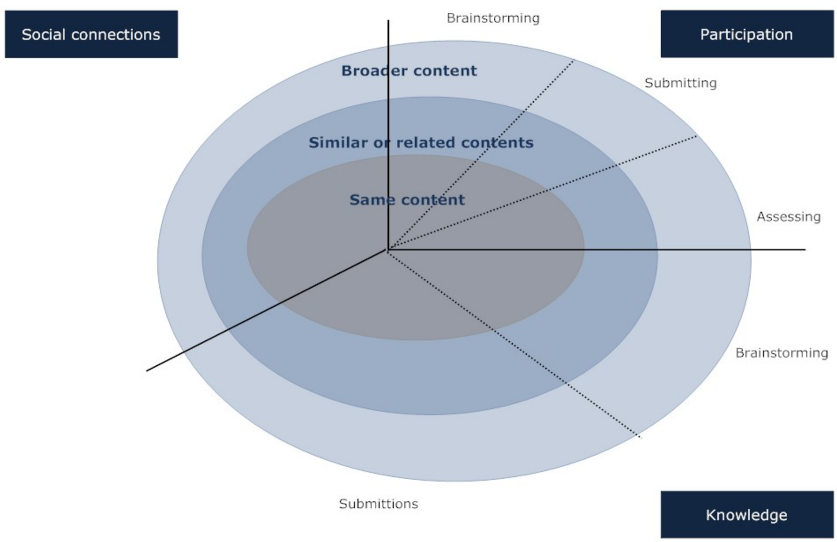

Fig. 5. Factors affecting estimation of potential collaborators' competences in DEPTHS

Experts, teachers and peers discovery. Based on the current learning context, this service suggests other students or experts as possible collaborators. Collaborators are selected and sorted using an algorithm which considers their competences on three different levels (Fig. 5): same content (i.e., current software problem), similar or related learning content (i.e., similar software problem) and broader content (i.e., software problem in the same course). Estimation of a peer's competence (C) on each level is performed through assessing three types of competence indicators:

- participation (C_p) in learning activities (e.g., brainstorming, submitting or assessing peers' works). Each activity has different impact factor on the system's estimation of the student's competences. This factor is defined in the system itself, but could be changed by the teacher.
- knowledge level (C_k) estimated by the teacher and obtained through the peers' evaluations, including projects evaluations and idea ratings. However, not all ratings have the same influence on the knowledge level estimation. For example, a high mark given by a student with high competences on the given topic has more impact on the final knowledge level appraisal than a high mark given by a student with average or low competences.
- social connections (C_{sc}) with the peer asking for help - the stronger the social connection with a specific person, the more suitable that person is for the help provision [14]. We believe that an already appointed social connection could be much more successful and effective than new connections with people one does not know. Social connections among students are mined from their interaction data (represented in accordance with the Learning Context ontology and stored in the DEPTHS's *Repository of interaction data*). For newcomers we are using their FOAF profiles, if available. Otherwise, this indicator is not considered.

All this data is collected through the queries performed on both DEPTHS' semantic repositories. Each type of competence indicator has different influence on the overall competence which depends on the predefined weight factor (WF) assigned to it (default systems' values could be changed by the teacher). To compute the overall competence of a peer, we use the following formula:

$$C_{total} = \frac{C_p \cdot WF_p + C_k \cdot WF_k + C_{sc} \cdot WF_{sc}}{3}$$

where $WF_p + WF_k + WF_{sc} = 1$, and $0 \leq WF_p, WF_k, WF_{sc} \leq 1$.

Context-based semantic relatedness. This service is used by all other services, as it allows for: i) computing context-based semantic relatedness between tags that students define and/or use in the given learning contexts [15]; ii) connecting students' tags with appropriate concepts of the domain ontology (i.e. disambiguation of the tags with the domain concepts); iii) resolution of students' queries containing both, tags and domain concepts relevant for the given learning context. This service connects tags with the concepts of the domain ontology as well as resources annotated with these concepts and stored in the DEPTHS semantic repositories (Repository of LO metadata and Repository of interaction data). That way the system selects only appropriate tags to show in the given moment based on the current learning context, and connects these tags with appropriate domain concepts related to it. This service is currently in its development stage and uses context-based semantic relatedness measure described in our previous work [15].

3.4 DEPTHS Implementation

The DEPTHS framework proposed in this work is a generic one. It is not dependent on any specific learning system/tool. However, for the purpose of our research we have implemented DEPTHS by leveraging open-source solutions and extending them with Semantic web technologies. Specifically, we have integrated Moodle LMS (http://moodle.org), ArgoUML software modeling tool (http://argouml.tigris.org), OATS (Open Annotation and Tagging System) tool (http://ihelp.usask.ca/OATS) for collaborative tagging and highlighting and LOCO-Analyst tool to provide teachers with feedback regarding students' activities [15]. Moreover, we use semantic annotation services of the KIM framework (http://www.ontotext.com/kim) and the Sesame server (http://www.openrdf.org) for semantic repositories. In order to provide students with context-aware educational services of the DEPTHS framework, we have extended both Moodle and ArgoUML so that they can make use of these services. Moreover, we have developed a Moodle module that supports project-based collaborative learning, that is, it supports and integrates in DEPTHS several kinds of collaborative activities such as brainstorming, submitting and assessing projects. Coupled with ArgoUML and educational services in DEPTHS it provides effective learning of software DPs, as described in Section 3 (screenshots used in Fig 1 are taken from this implementation).

4 Evaluation

The evaluation of DEPTHS was conducted in February 2009, in the context of a course that the first author of this paper taught at the Department of Computer Science of Military Academy in Belgrade, Serbia. DEPTHS was evaluated with a group of 13 students of the fifth year of the computer science program who took part in our course on software development. The students already had some elementary knowledge in the domain of software DPs, but they were not familiar with the particular software DPs used in this experiment (Facade, Adapter, Strategy, Composite, Visitor and Decorator).

The students were divided into 4 groups (3 groups with 3 students and 1 group with 4 students), based on the teacher's subjective opinion about their knowledge in the domain of software development and their previous results in the courses related to software engineering. The size of the groups is based on our belief and teaching experience that work in small size groups (3 or 4 students) is a necessity for effective education of software engineers.

The aim of the evaluation was to determine how effective DEPTHS is for learning DPs. Specifically, we wanted to evaluate the perceived usefulness of the use of engagement theory in software engineering education. Moreover, we wanted to check if active students' involvement in real world problems and the employment of context-aware educational services could ensure a more effective way of imparting knowledge in the domain of software development.

Before the experiment started, a demonstration of DEPTHS functionalities along with a training using a task similar to the one used in the experiment, were performed with students. Each group was assigned a different task (i.e., a software design problem). Students were asked to suggest solutions and evaluate each others' solutions within one week period of time. Actually, project organization used in the experiment was based on the learning workflow described in Section 3.

We used an interview to collect data about the students' satisfaction with and attitudes towards learning with the DEPTHS system. The interview was also supposed to reveal the students' perceptions regarding the effectiveness of learning with DEPTHS.

The questions were divided into three sections based on the type of information we were interested in. The first section (14 questions) gathered data regarding the students' previous experience with computer-assisted learning. The questions of the second section (15 questions) were related to the DEPTHS system and the third section (11 questions) was aimed at evaluating the learning program on software DPs offered by DEPTHS system. Most of the questions (33) were multiple-choice questions with 5 possible answers, ranging from 1 (most negative) to 5 (most positive). There were 6 open-ended questions and 1 combined (multiple choice and open-ended).

We used three methodologies to analyze the results gained in this experiment. First, we analyzed the results of the overall corpus of the students using standard descriptive statistic instruments such as frequency, mean, median, and average. The second kind of analysis consisted of comparing the groups of students that were derived by splitting the results data based on the students' answers on the questions from the first section of the interview. Finally, we used Pearson's chi-square test to find if there is significant association between different variables. We used SPSS tool (www.spss.com) to process data and analyze the results.

Having analyzed the results, we found that the majority of students (84.62%) have experience in using Internet to find relevant information, collaborate with colleagues on solving common tasks (53.85%) and use tools for message exchange and discussion (84.62%). However, they have far less experience with online learning tools (only 23.07% are familiar with e-learning tools) and using the Internet to find peers for solving problems (only 38.46% answered positively).

The DEPTHS system received high marks from the students. Majority of them (53.85%) reported they have learned as effectively as in traditional way, and 30.77% reported that they have learned more than in traditional way. The students reported it was intuitive and very easy to use (76.92%), but they also have reported some technical issues. These issues were caused by the software bug that caused problems in uploading UML diagrams to the repository, and have been resolved one day after the beginning of the evaluation. We believe that this issue could have affected students' confidence in the system. The students felt educational services provided in DEPTHS are very helpful: Web resource recommending service - 92,30%; course content recommending service – 84,61%; and peers recommending service – 76,92%. They also thought that the activities provided within the tasks considerably contribute to the learning process (brainstorming – 76.92%, and evaluating each other's works – 100%).

Having analyzed the trends of the different groups of students based on their answers on the first group of questions, we found that all students that have taken a course offered through an LMS consider educational services provided in DEPTHS as very useful. Students that have used computers to find relevant collaborator for domain that they are currently working on, have much more positive attitude for brainstorming tool in DEPTHS and learning from other students ideas. There is no significant difference between those students who are familiar with e-learning and those who are not with respect to their experience of learning in DEPTHS environment. However, it is interesting to notice that all students who are not familiar with e-learning gave the highest mark for educational service for Web pages recommending.

We have identified 27 variables' pairs that have significant association through the use of Pearson's chi-square (we used standard level of significance $p<0.05$). For example, there is a significant association between the students' answers on the question if they used tools for sending messages and discussions, and the question if they think that the other students' ideas were useful. Many useful conclusions could be drawn from these associations about how close various variable impacts might be on student achievement. For example, we found that students' satisfaction with the educational services for recommending relevant Web pages and course content affected their latter satisfaction with their learning results using this program. We believe that this association is strongly related with the students' level of adoption of these services as very important and useful part of a learning system. However, the deeper analysis of these results is out of scope of this paper.

5 Related Work

The framework proposed in this paper is related to two favored research fields: collaborative learning in the domain of software engineering and context-aware learning. 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 that support knowledge creation and sharing through the collaborative learning process based on the active learning principles.

In [16], the authors suggested an 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 the integration of existing tools, it still lacks context-aware educational support (e.g., recommending online resources relevant for the given learning context) that is available in our framework. Besides our framework promises additional support for collaborative learning as it offers social tagging support.

One of the main objectives of the EU project APOSDLE is to develop a system that would be able to provide knowledge workers with learning resources relevant for their present work context [17]. In particular, based on the immediate work context of a user, the system should identify his/her missing competencies and learning needs and suggest appropriate learning resources. These learning resources are created on-the-fly from a variety of resources (documents, videos, expert profiles, and so on) already stored in the workplace and may be in the form of learning material or suggestions to contact experts and /or colleagues. The system's functionality is based primarily on its knowledge base that stores an integrated representation of various kinds of knowledge (e.g., domain, task, and instructional). Knowledge integration and advanced search and retrieval capabilities (associative information retrieval) are enabled by the Semantic Web technologies. Obviously, this approach has a lot of commonalities with the one we suggested in this paper. Nonetheless, having grounding our approach in pedagogical theories and best practices of collaborative learning, we can expect to provide students with better learning experience.

An e-learning framework proposed in [18] supports a peers' recommendation based on the student's context. Student's context is defined as the result of the interaction of three key elements: the knowledge potential, the social proximity and the technical access. Comparing to DEPTHS approach of peers' recommendation, this approach is advantageous as it considers technical context that includes factors that may influence e-learning such as technical media or time proximity. However, unlike the approach proposed in DEPTHS, this approach does not consider the influence of student's participation in the learning activities on his competences to help other students. Another approach [19] suggests use of ontologies to support online professional communities. By constructing a semantic model of the content, the interactions and the structure of the community, the activities of an online professional community can be supported. However, our work goes a step further and uses ontologies in concrete learning contexts.

Another related research work is going on within the recently started ENSEMBLE project (www.ensemble.ac.uk). The project aims at exploring the potential of the emerging Semantic Web to support teaching in complex and rapidly evolving fields where case-based learning is the pedagogical approach of choice. The main idea is to combine key elements of digital repositories, semantic web technologies, and features of 'social software' in order to allow for reuse through reconfiguration, adaptation, and collective action. However, this project is still in its early research stage.

6 Conclusions

Collaborative learning through project-based work helps students reflect on their learning experiences in ways that promote abstraction from experience, explanation of results, and understanding of conditions of DPs applicability in real world situations; it also provides the experience of working in software development teams. Following this paradigm, we have developed a learning environment for software DPs which leverages semantic technologies to integrate several existing learning systems and tools, and provide context-aware educational services that together allow for effective learning of software DPs. Our present implementation and first evaluation results convince us that this environment could significantly contribute to effective teaching and learning of DPs. The use of Semantic Web technologies greatly facilitated the development process. In particular, by using a common RDF-based data model as an exchange mechanism among content and (interaction) data providers (i.e., learning systems/tools integrated within DEPTHS), we enabled seamless integration of heterogeneous data and content formats different providers rely upon. Moreover, by leveraging the integrated data and the semantics of the interaction data model (i.e., the Learning Context ontology) we developed educational services that enable fast and effective search for relevant resources and possible peers.

We are encouraged with the results of the initial evaluation study that show very positive students' attitude toward learning in DEPTHS. Students' perception of system's usefulness is valuable and encouraging for our further research. However, the results we got still do not have a statistical power, as the participants' sample was too small. Further research is required that would include sufficient participants to ensure the general applicability of the findings. In addition, in our future work we intend to do a more precise evaluation of each specific educational service as well as a quantitative evaluation of the students' learning effectiveness. Moreover, beside the support for working with text-based and UML based learning objects, our intention for future work is to extend the system to support other non text-based learning objects (e.g. Flash animations, videos, graphics). We hope this will make the DEPTHS framework useful in broad range of domains.

References

[1] Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995)

[2] Warren, I.: Migrating to a Teaching Style that Facilitates Active Learning. CiLTHE Stage 1 Dissertation, Lancaster University (2002)

[3] Jeremić, Z., Jovanović, J., Gasević, D.: Towards a Semantic-rich Collaborative Environment for Learning Software Patterns. In: Proc. of the 3rd European Conference on Technology Enhanced Learning, pp. 155–166 (2008)

[4] Jovanović, J., et al.: Using Semantic Web Technologies for the Analysis of Learning Cotent. IEEE Internet Computing 11(5) (2007)

[5] Jazayeri, M.: The Education of a Software Engineer. In: Proceedings of the 19th IEEE International Conference on Automated Software Engineering, pp. xviii–xxvii (2004)

[6] Bagert, D., Hilbum, T., Hislop, G., Lutz, M., McCracken, M., Mengel, S.: Guidelines for Software Engineering Education, Version 1.0. Technical Report CMU/SEI-99-TR-032, Software Engineering Institute, Carnegie Mellon University, Pittsburgh PA (1999)

[7] Jeremić, Z., Jovanović, J., Gašević, D.: Project-based Collaborative Learning Environment with Context-aware Educational Services. In: Proceedings of 4th European Conference on Technology Enhanced Learning (ECTEL 2009), Nice, France (accepted, 2009)

[8] Dietrich, J., Elgar, C.: A Formal Description of Design Patterns using OWL. In: Proceedings of ASWEC. IEEE Comp. Soc., Los Alamitos (2005)

[9] Montero, S., Diaz, P., Aedo, I.: Formalization of web design patterns using ontologies. In: Proceedings of the 1st International Atlantic Web Intelligence. Conf. (AWIC), Spain, pp. 179–188 (2003)

[10] 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)

[11] Henninger, S.: A Framework for Flexible and Executable Usability Patterns Standards. In: 31st IEEE Software Engineering Workshop (SEW-31), USA, pp. 23–34 (2007)

[12] Coplien, J.O.: Software Patterns. SIGS Books, New York (1996)

[13] Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)

[14] Bull, S., Greer, J.: Peer Help for Problem-Based Learning. In: Proceedings of ICCE/ICAI, Taiwan, pp. 1007–1015 (2000)

[15] Torniai, C., Jovanovic, J., Gasevic, D., Batemen, S., Hatala, M.: E-Learning Meets the Social Semantic Web. In: Proc. of 8th IEEE Int'l Conf. on Advanced Learning Technologies, pp. 389–393 (2008)

[16] Jovanović, J., Rao, S., Gašević, D., Devedžić, V., Hatala, M.: An Ontological Framework for Educational Feedback. In: Proceedings of the 5th International Workshop on Ontologies and Semantic Web for Intelligent Distributed Educational Systems, USA, pp. 54–64 (2007)

[17] Ghidini, C., Pammer, V., Scheir, P., Serafini, L., Lindstaedt, S.: APOSDLE: Learn@work with semantic web technology. In: I-Know 2007, Graz, Austria (2007)

[18] Yanlin, Z., Yoneo, Y.: A Framework of Context Awareness support for peer recommendation in the e-learning context. British Journal of Educational Technology 38(2), 197–210 (2007)

[19] Ankolekar, A., Sycara, K., Herbsleb, J., Kraut, R., Welty, C.: Supporting online problem-solving communities with the semantic web. In: Proc. of the 15th International WWW Conference, Scotland, pp. 575–584 (2006)