Jeremic, Z., Jovanovic, J., & Gasevic, D. (2009). Evaluating an Intelligent Tutoring System for Design Patterns: the DEPTHS Experience. *Educational Technology & Society*, *12* (2), 111–130.

Evaluating an Intelligent Tutoring System for Design Patterns: the DEPTHS Experience

Zoran Jeremić[1], Jelena Jovanović[1] and Dragan Gašević[2]
[1]FON – School of Business Administration, University of Belgrade // jeremycod@yahoo.com // jeljov@gmail.com
[2]School of Computing and Information Systems, Athabasca University, Canada // dgasevic@acm.org

ABSTRACT

The evaluation of intelligent tutoring systems (ITSs) is an important though often neglected stage of ITS development. There are many evaluation methods available but literature does not provide clear guidelines for the selection of evaluation method(s) to be used in a particular context. This paper describes the evaluation study of DEPTHS, an intelligent tutoring system for learning software design patterns. The study which took place during the spring semester 2006 was aimed at assessing the system's effectiveness and the accuracy of the applied student model. It also targeted the evaluation of the students' subjective experiences with the system.

Keywords

Intelligent tutoring systems, design patterns, learning evaluation, student model assessment evaluation, adaptive presentation

Introduction

Software design patterns (DPs) have been recognized as very important and useful in real software development since they provide an elegant way of getting around problems that often occur. To be more precise, DPs are common conceptual structures that describe successful solutions to common and recurring software design problems. They can be applied over and over again when analyzing, designing, and developing software applications in diverse contexts (Harrer & Devedzic 2002). DPs are aimed at seamless reuse of software designs and architectures that have already proven their effectiveness in practice. In addition, their application guaranties high quality software solutions that are easy to maintain and extend.

To make use of the benefits that a DP-based software development offers, one has to master DPs both in theory and practice. Accordingly, it is very important to devise an effective approach for teaching DPs. Our experience in teaching object-oriented programming to software engineering students, suggests that there is no single best way to learn DPs. It depends on the students' knowledge of and experience in the field of software engineering in general and design patterns in particular. For example, beginners in this field are often confused by the number of available patterns, different forms for presenting design patterns and plenty of available sources of information (Raising 1998). It would be best for them to be provided with a step-by-step introduction into the field and strict instructions on how to conduct each step (i.e., what to learn, in which manner, and what resources to use). In this case, sources of information should be carefully selected in order to avoid the threat of using a source providing information that is too difficult for a beginner to understand. On the other hand, for experienced programmers this way of strictly guided learning is not suitable. Rather, exploratory learning would be more appropriate learning strategy to use. It assumes giving a student control over his (since gender-neutral language tends to be imprecise and/or cumbersome, throughout the paper we arbitrarily decided to use masculine pronoun for each generic reference to a student) learning process. In particular, the student chooses on his own what to learn (topics and their order), when to learn and how fast to proceed, at which level of difficulty to learn, and in what way to learn. Such a student has to have access to many different information sources to explore, and thus be enabled to develop his own understanding of the topics explored. Exploratory learners are always intrinsically motivated. It is not possible to force this learning style upon a learner (Holzkamp, 1995). To sum up, students come from different knowledge backgrounds and have different learning styles and preferences. To effectively meet these diversities, an e-learning system should be able to offer different learning experiences to different students.

We found that an intelligent tutoring system (ITS) for learning DPs can successfully address these issues, since it can provide different learning strategies for different types of students, as well as, adapt learning content according to the student's performance. ITSs use artificial intelligence's techniques and methods to provide a student with a content that is neither too easy (thus avoiding the drop of concentration), nor above his/her ability to understand (hence not

frustrating the student). ITSs make it possible to deploy a course suitable for students with widely varying needs and levels of motivation.

We have developed DEPTHS (Design Patterns Teaching Help System), an ITS for learning DPs, as a part of our research in the area of teaching DPs in higher education (Jeremic & Devedzic 2004). Our goal was to provide students with the benefits of one-on-one instruction in a cost effective manner. The system was primarily intended for teaching undergraduate students of Computer Science, but it can be equally successful in other education settings, as well. DEPTHS performs adaptation of the teaching material based on the student performance and the usage tracking data collected during the learning session. The quality of the adaptation provided by the system depends on many factors, such as the accuracy of the diagnostic tools that collect and process data, the student model that is used to store the data and the embedded teaching rules. The only way to verify the quality of the DEPTHS's adaptation functionalities is to evaluate the system in real conditions, with students who are learning the subject material. Evaluation is the central part of quality assurance, as it provides feedback on teaching and learning and thus helps to make the system better.

The evaluation of DEPTHS was conducted in the 2005/06 academic year, in the context of a course we teach at the Department of Computer Science of the Military Academy of Belgrade in Serbia. The aim of the evaluation was to determine how effective DEPTHS is for learning DPs. To this end, the following agenda was set to guide the research:

- Do a literature review in order to determine how the effectiveness of learning can be evaluated.
- Generate an evaluation instrument for determining the effectiveness of learning.
- Conduct an evaluation study to determine how effectively students learn with DEPTHS.

The paper is organized as follows: after giving literature review (Section 2), we introduce the DEPTHS system and give a general description of its functionalities (Section 3). Section 4 explains the DEPTHS architecture in details, whereas Section 5 gives an example usage scenario with DEPTHS. In section 6 we give a detailed description of the evaluation studies that were performed to evaluate the system's effectiveness and the accuracy of its student model. We give a comparison with some related work in section 7. Finally, we conclude with a discussion of the implications of this work for the next generation of intelligent tutoring systems.

Literature Review

Evaluation of an ITS is a difficult task. In particular, the main difficulty lies in the absence of a widespread agreement on how it should be performed.

The most well-known and used model for measuring the effectiveness of training programs is the model developed by Donald Kirkpatrick in the late 1950s (Kirkpatrick 1979). Since then, it has been adapted and modified by a number of researchers, but the basic structure has remained the same.

The Kirkpatrick's model defines four levels of evaluation (Kirkpatrick 1979):

- Evaluation of reactions – Reaction is the term that Kirkpatrick uses to refer to how much the students liked a particular training program. An evaluation of students' reactions consists of measuring their feelings, and does not include a measurement of what was actually learned. A typical instrument for gathering information regarding students' reactions is an open-ended questionnaire. This information is easy to collect, but does not tell enough about the training success.
- Evaluation of learning – This level of evaluation identifies how well the students understood the facts and techniques presented in the training material. This is much more difficult to measure than reactions. At this level, each student's learning should be measured by quantitative and objective means. Endres and Kleiner (1990) state that pretests and posttests are necessary when evaluating the amount of learning that has taken place.
- Evaluation of behavior (transfer of learning) – This level of evaluation identifies how well the students apply the acquired knowledge in their everyday practice. This kind of evaluation is more difficult than the abovementioned two and there are very few examples of studies in this area. Feedback from students, their supervisors, and peers as well as some other techniques can be used for collecting information at this level.

- Evaluation of results – The fourth level of evaluation refers to the training results or impact on the organization. Although measuring training programs in terms of results may be the best way to evaluate their effectiveness, this kind of measurement is very difficult to conduct. The major obstacles include the existence of many factors that are impossible to evaluate (e.g., social interaction between employees), and the relative lack of objective, valid tools to use. McEvoy and Buller (1990) question the relevancy of such evaluations. They claim that not all training is result oriented: it can also be used for purposes other than achieving a measurable impact on the performance of an individual employee.

Since Kirpatrick established his original model, other theorists, e.g., Jack Philips (1997), and indeed Kirkpatrick himself, have referred to a possible fifth level, namely ROI (Return of Investment) which is used to evaluate the efficiency of an investment or to compare the efficiency of a number of different investments.

Endres and Kleiner (Endres & Kleiner 1990) use Kirkpatrick's model as the foundation for their approach to evaluating the effectiveness of management training. They suggest setting initial performance objectives and monitoring accomplishment of those objectives after training. They offer an example in which participants are asked to write their personal and professional objectives at the end of the training experience. These objectives are sent to the participants approximately a week after the training. Two months later they are sent again, and the participants are asked to comment on their performance against these objectives:

- Reeves and Hedberg (2003) criticize Kirkpatrick's approach as overly simplistic for the following reasons:
- Control groups are rarely feasible in education or training contexts;
- Paper-and-pencil tests lack reliability and validity in measuring KSAs (Knowledge, Attitude, and Skills).
- 100% response rate is unrealistic
- Evaluation results are not the only input to decision-making within an organization

The experience reflected in the literature suggests that trainers often incorporate at least the first three levels of the Kirkpatrick's model in the design of training programs. In fact, many authors emphasize the importance of considering early in the training design process how each level of evaluation will be addressed. Kirkpatrick's fourth level of evaluation is still difficult to apply in practice (i.e., do the required measurements). The difficulty lies in the inability to separate training from the multitude of other variables that can impact long-term performance.

Functionalities of DEPTHS

DEPTHS (Fig. 1) is an ITS for teaching DPs. The course implemented in DEPTHS gradually introduces the concept of DP and presents students with the most frequently used kinds of patterns. In addition, DEPTHS provides learning materials adjusted to the student's performance (e.g., his background knowledge and performance in the current domain), and cognitive capacity (e.g., his working memory capacity). Even though it is a tutoring system, DEPTHS allows students to choose between the tutoring and the self-paced learning mode.

The topics of the DEPTHS course are organized in a dependency graph, with links representing the relationship between the topics, such as prerequisite and related topics relationships. A student is ready to learn a topic only if he has completed the prerequisites. Accordingly, the student must achieve sufficient score for the topic in order to qualify to proceed with the course.

DEPTHS performs both content-level (adaptive presentation) and link-level adaptation (adaptive navigation support) (Brusilovsky 1996)**.** Adaptive presentation means that students with different performance levels get different content for the same domain topic. During each learning session, DEPTHS observes the student's progress and adapts the course presentation accordingly. Likewise, the student's performance is used to adapt the visual representation of hyper-links to related topics. DEPTHS provides two kinds of navigational adaptation through the course material (Brusilovsky 1996):

- Direct guidance – The student is given only one option to continue the browsing activity, that is, just one button enables advancing to the next page. The destination of the "next" button (Fig. 1E) is dynamically determined by the system.

- Link removal – advanced students can choose the topics to learn by selecting appropriate links from the Content menu (Fig. 1A). However, the links that the system considers inappropriate are removed, i.e. they are no longer available.

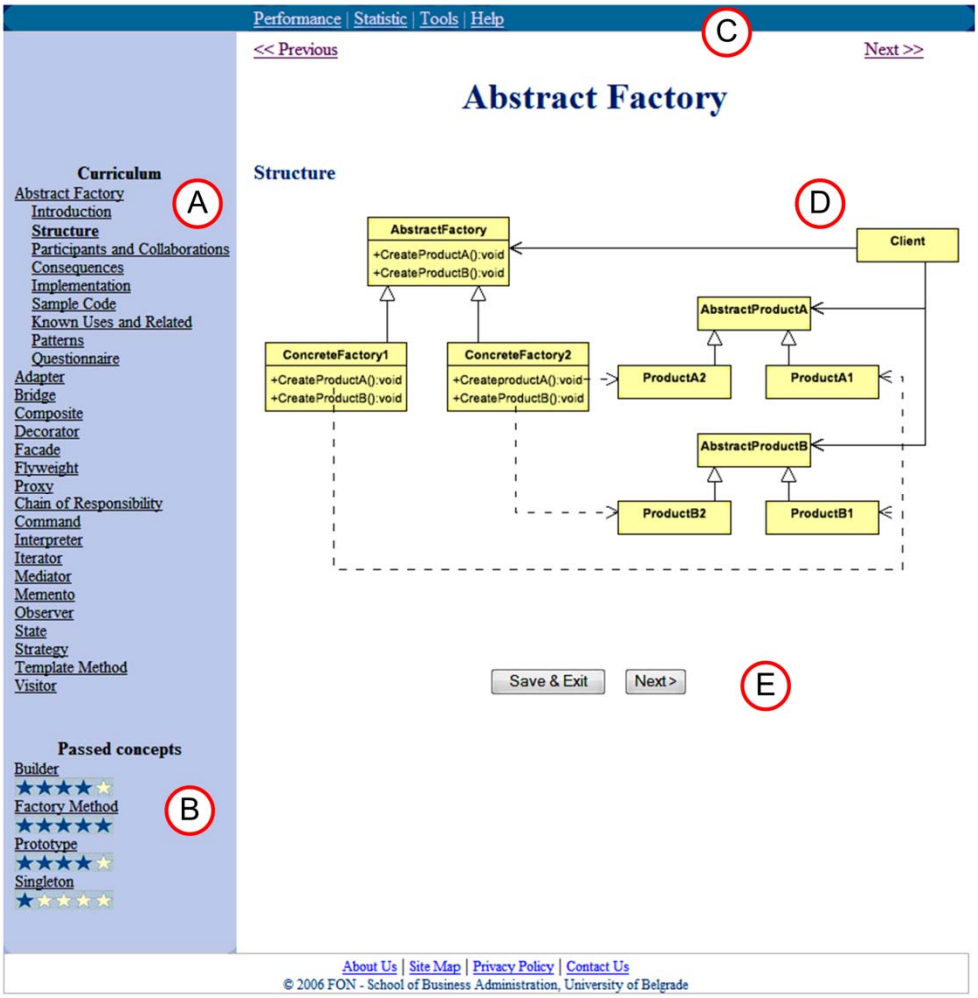


Figure 1. Web-based GUI of DEPTHS

DEPTHS continually monitors the student's behavior and performance and stores all the interaction data. This data is used to refine the instructional plan, as well as to generate recommendations for the further work (Prentzas et al. 2002).

DEPTHS architecture

The architecture of DEPTHS (Fig. 2) follows the main guidelines for ITS architecture. In particular, it is widely agreed that the major functional components of an ITS architecture should be: the domain knowledge model, the student model, the pedagogical module, and the user interface (Hartley & Sleeman 1973, Burton & Brown 1976, Wenger 1987). Accordingly the DEPTHS architecture consists of Pedagogical Module, Expert Module, Student Model, Domain Model, and Presentation module (Jeremic & Devedzic 2004b). The user interface component of the traditional architecture is adapted to the web-based environment and is dubbed Presentation Module in our model. In the following subsections we describe each of these components in turn.

DEPTHS is implemented as a classical client-server Web application with a Web browser on the client side and Tomcat 5.0 Web Server as JSP container on the server side (Jeremic et al. 2004).

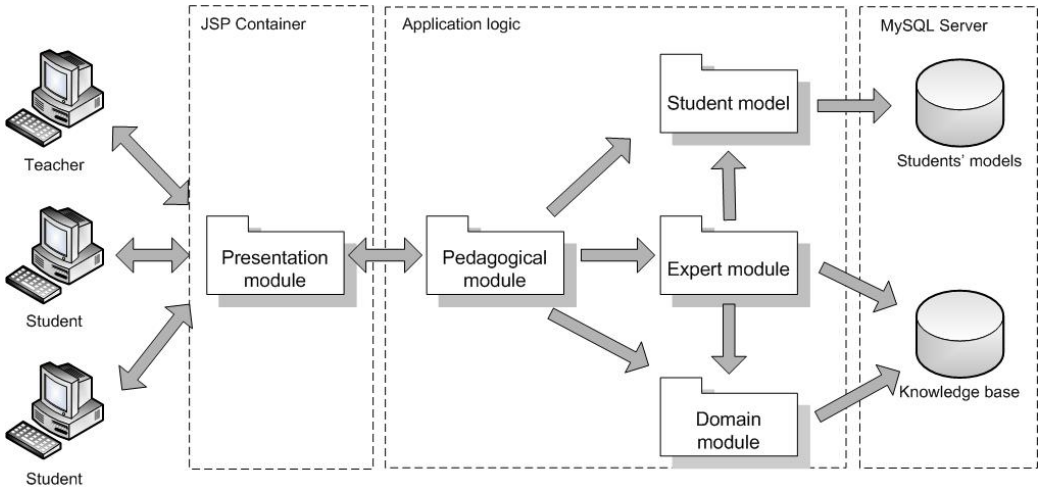


Figure 2. DEPTHS system architecture

Domain Model

Domain Model is designed as a network of concepts. What we refer to as a 'concept' is named differently in different research papers – attribute, topic, knowledge element, object, and learning outcome. We use this term to refer to a topic that describes a single design pattern. Concepts could be related to each other with prerequisite relations. One concept might be a prerequisite for another concept, i.e., the former should be learned before the latter can be presented. For each concept, there is a knowledge threshold (measured through the score obtained on assessment mechanisms) that must be reached by a student before the system can assume that the student has learned that concept. Each concept is decomposed into units – content elements that correspond to a particular web page to be presented to students. The number of units for a particular concept is defined by the teacher. DEPTHS uses the unit variants technique (Beaumont 1994, Kim 1995, Kay & Kummerfeld 1994) to achieve content-level adaptation. This means that the system keeps a set of alternative units for each concept and makes a selection based on the student's characteristics (e.g., based on the student's knowledge level: very low, low, good, very good, and excellent). Each unit has an arbitrary number of fragments – chunks of information that can be presented to the student. A unit may also contain fragment variants, i.e. fragments related by the "OR" relationship.

Pedagogical Module

Pedagogical Module provides the knowledge infrastructure necessary to tailor the teaching material to the student's learning needs as captured in his student model (see the next section for a detailed explanation of the *Student model*). This module is composed of three functional components (Jeremic et al. 2004):

- Instructional Planner generates an instructional plan as a sequence of instructional plan items (e.g. concepts and lessons) appropriate for the student's knowledge of the domain.
- Feedback Mechanism is responsible for providing a student with advice and recommendations for further work during a learning session.
- Assessment Component is in charge of collecting and processing data about a student's learning process and storing it in the student model. The accuracy of the diagnostic tools of this component is one of the main factors that affect the quality of the adaptation process.

All the above mentioned components use Expert Module for making decisions, such as selecting appropriate lesson to teach or questions to ask the student, and selecting advice about the lesson that is appropriate for a student to learn next.

Expert Module

Expert Module's main role is to make decisions that are used by Pedagogical Module for generating teaching plans and adaptive presentation of the teaching material. In many traditional ITS systems these roles are in the charge of the pedagogical module, but we decided to separate them in two distinct modules in order to increase the flexibility of the system.

One of the main tasks performed by this module is selection of instruction plan items that best suit to a particular student's knowledge level (Fig. 3). This is a three-step process which starts with the creation of a concepts plan. To create such a plan, Expert Module uses a set of facts about the student (from the Student model), the set of available domain concepts (from the Domain model), and a rule for concepts selection. Expert Module selects concepts having minimal (i.e. entry) knowledge level lower or equal to the student's current knowledge level and having all of the prerequisites satisfied. In other word, Expert Module will not choose those concepts that it believes the student is not ready to learn yet, neither will it select a concept whose prerequisite (some other concept) the student has not passed yet. This process repeats each time the student knowledge level is changed. The created concepts plan is displayed to the student in the form of adaptive Content menu (Fig. 1A).

When the student or the system selects the concept to be learnt next, Expert Module proceeds to create a detailed lesson plan. In the final step a test plan is created. DEPTHS does not create a detail lesson and test plan for each concept because there is a chance that some of them will not be needed (depending on the student's performance). These two steps will not be described in detail here since that issue is out of the scope of this paper (Jeremic, 2005).

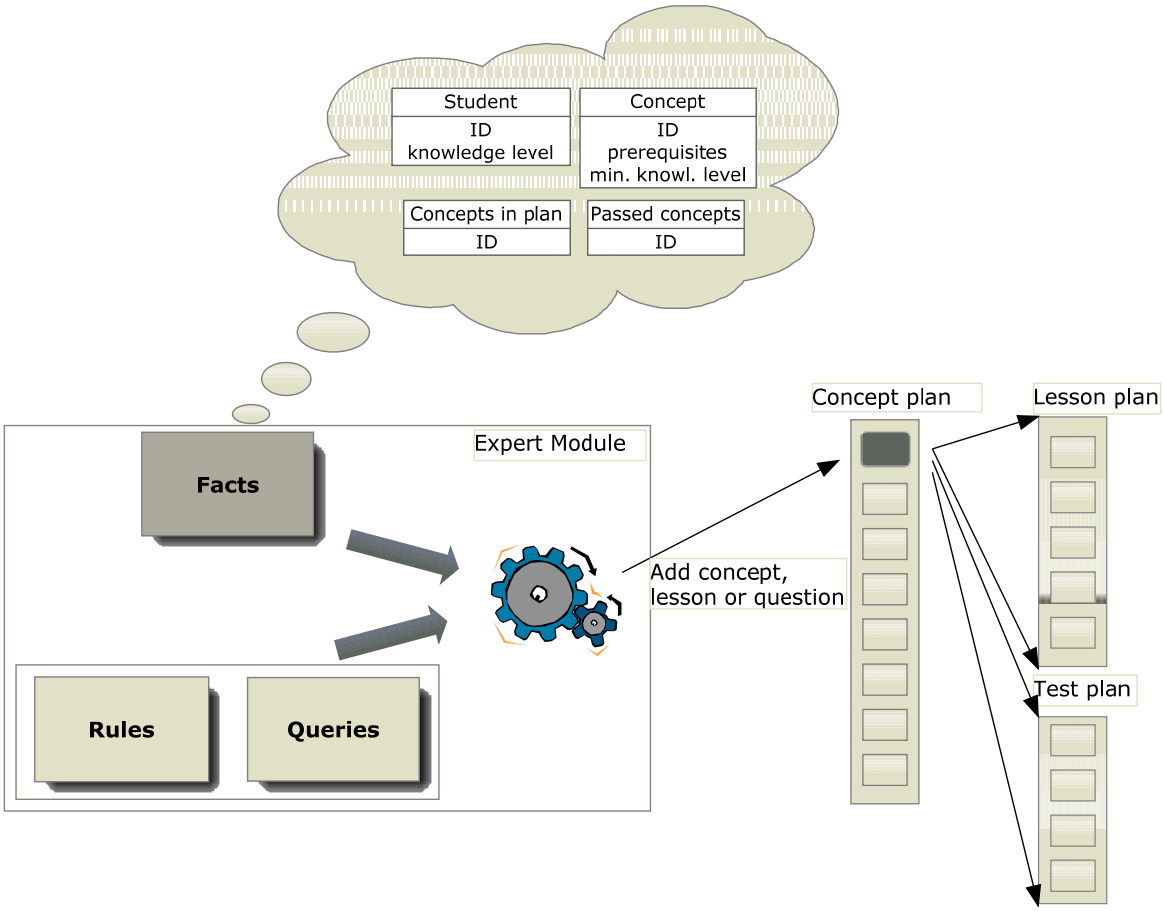


Figure 3. The process of creating Instructional plan in Expert module

Student Model in DEPTHS

The student model stores and updates data about the student's performance in a specific subject domain. It is essential for the system's operations that adapt instructional material to the student's characteristics (Devedzic 2001) and comprises both the model of the student and the mechanisms for creating the model (Prentzas et al. 2002).

The student model may keep any number of students' characteristics, depending on the system requirements. In DEPTHS, three basic categories of the students' characteristics are used:

1. Personal data – personal characteristics of a student (e.g., name, ID, and e-mail). This information represents the static part of the student model, and is collected during the student's first learning session with the system, through a questionnaire.
2. Performance data and individual preferences – cognitive and individual characteristics of a student. This part of the student model represents a mixture of static and dynamic data. Static data, such as the desired detail level, the experience level or the preferred programming language, is collected during the registration procedure (through a questionnaire). Dynamic data is derived from the learning sessions and is changed as the student progresses through the course material. Examples of dynamic data include: actual skill level, learning style, reasoning ability, etc. The performance data, both static and dynamic, is quintessential for one of the system's primary functions – adaptive presentation of the teaching material. This data represents the system's 'believes' of the student's traits. The quality of these believes directly affects the quality of the content-adaptation process.
3. Teaching history – data related to the student's interactions with the system during learning sessions. This part of the student model keeps track about everything that the student has done during the learning process. In particular, it keeps data about each student's session with the system, such as the time spent on solving tests and the student's success on a particular test. This data is less important for adaptive presentation than performance data, but it is very important for reflective learning which often plays considerable role in a learning process. The system uses this data in order to provide the student with feedback about what he has done well and where he failed, what should be revised and how to make the learning process more successful.

When registering a new student, the system creates his student model and populates it with default values. Miskelly uses a similar approach in Interactive Student Modeling System (ISMS) (Miskelly 1998). In particular, based on the student's self-assessment, the system classifies the student into one of the following categories: beginner, intermediate or advanced (expert), i.e., it assigns him one of the predefined stereotypes. A learning session then proceeds in compliance with the assigned stereotype until the completion of the first concept, when the student is tested for the first time. Based on the test results, the *Pedagogical Module* updates the values of the attributes in the *Student Model* concerning the student knowledge level (actual knowledge level). This data belongs to the performance data, and is updated based on the system's conclusions about the students' knowledge level.

DEPTHS uses tests to assess students' knowledge of each domain topic. A test is created as a set of questions and exercises dynamically selected depending on the student previous knowledge level. The *Pedagogical module* (Fig. 2), i.e. its *Assessment Component* to be more precise, is responsible for diagnosing student knowledge level based on the test results. It uses a set of pedagogical rules and domain knowledge in order to assess test results and diagnose the student's knowledge level based on these results. The result of this diagnosis depends on the three main variables: 1) the test difficulty, which is based on the difficulty of each question as specified by the question creator; 2) the time spent to solve the test, which is compared with the sum of average time for each question as specified by the question creator; and 3) the test result (success). The computation of the student's knowledge level is based on fuzzy sets (test difficulty, test duration and success) and is calculated as a fuzzy value (ranging from 0 to 6). Thus, computed knowledge level represents a quantitative measure of the system's belief about the student's level of understanding of the respective concept. Based on each concept's assessment, the system creates an overall estimation of the student's understanding of the domain. This value is calculated as the average value of all concepts degrees of mastery. If assessment on a particular concept is performed more than once, the best score is taken.

DEPTHS in Use – An Example Usage Scenario

When a student runs a session with DEPTHS for the first time, he has to give some basic information about himself, as well as to describe his own experiences in software development as a beginner, intermediate or advanced. After that, he is requested to solve a questionnaire composed of very general questions concerning software development.

Based on the questionnaire results and the student's self-evaluation, the system categorizes the student into an appropriate stereotype (student) model. Thereafter, the system switches to the teaching mode and develops an initial plan of instructional actions.

In the teaching mode, the student is presented with a concepts plan available as a navigation menu on the left side of the screen (Fig.1A), as well as the main menu and command buttons available on the top and bottom of the screen (Fig. 1C and 1E). Those links that the system considers inappropriate for the given student are hidden. The student can choose between two options available in teaching mode:

- following the system's advice (recommended for beginners);
- choosing the concepts to be learned next on his own (suitable for advanced learners).

If the first option is chosen, the command buttons are used for navigation; otherwise the navigation is done through the navigation menu. Each time the student presses the "Next" button or chooses a link from the navigation menu, a new lesson is presented in the main window area. Which lesson is going to be presented depends on the lesson plan created before learning of a specific concept has started. As DEPTHS uses "lesson variants" for adaptation on lesson's level, it will choose one of the available variants of the lesson based on the student's previous results. However, the student can choose or change the desired detail level (low, normal, high) any time from the main menu. For example, low detail level will contain only basic description of design patterns without source code examples, diagrams, etc. Student can also choose which programming language will be default for presenting source code examples (Java, C# or C++), or which design pattern form will be used (Gang of Four, Coplien or Alexandrian).

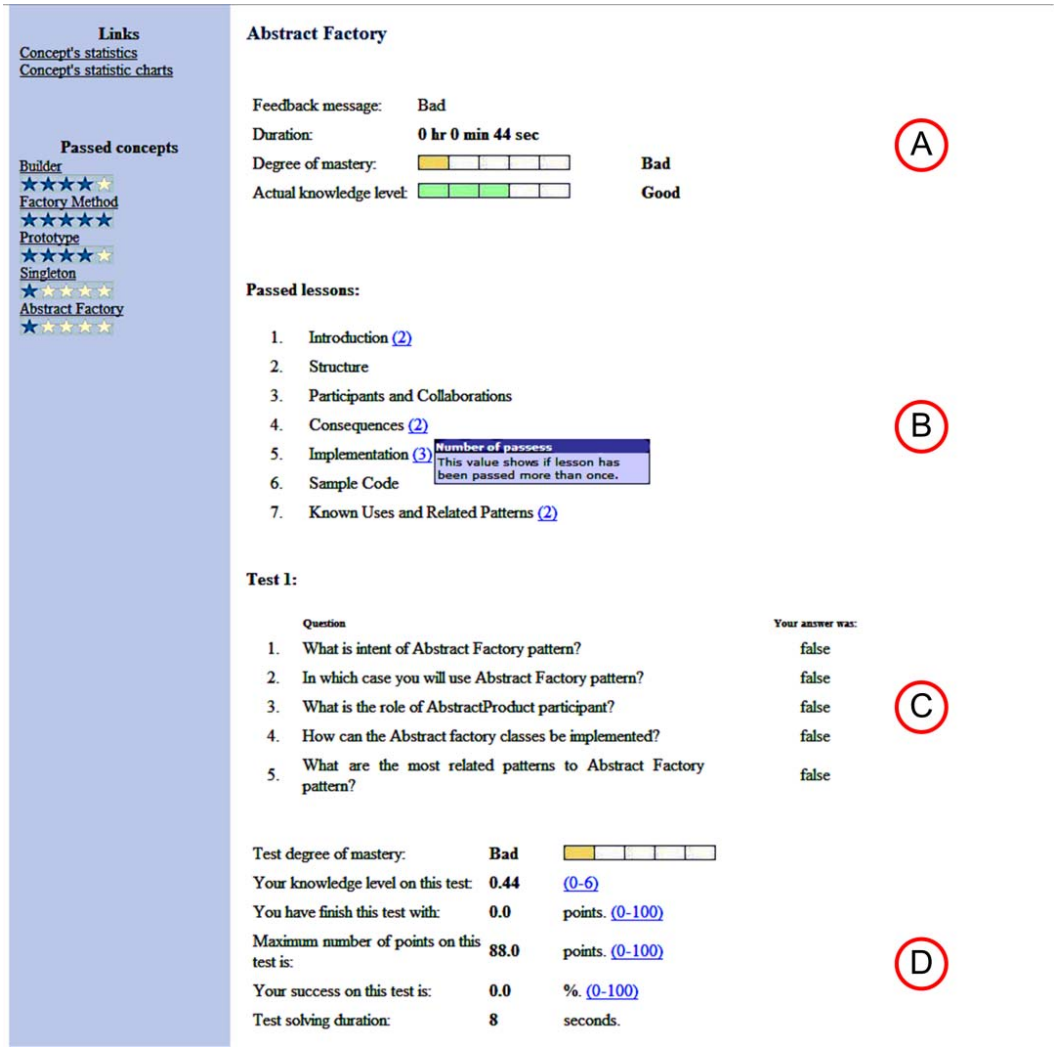


Figure 4. The student can see his performance record, which stimulates reflective learning

After the presentation of one domain concept is completed, the system switches to the evaluation mode. In this mode, the system provides the student with exercises and tests in order to assess how much the student has learned about the concept. If the student demonstrates insufficient knowledge of the concept, the system suggests better exploration of the current concept and provides the student with an appropriate learning path for that concept. As each question is connected with specific lesson(s), the system can easily provide links to the lessons which has to be relearned. The student can choose either to follow the advice or to move on. After the repeated exploration of the suggested lesson(s), the student is presented with another test addressing the lessons that have been relearned.

After each test, the system displays a page with student-directed feedback, including (but not limited to) the following:

- the student's performance on each question,
- the correct answer for each question,
- the lessons that should be learned again,
- the student's overall knowledge of the tested domain concept,
- the correlation between this test result and his previous results,
- many other statistical information, such as the time spent to solve the test, the difficulty level of each question, and points number on each question.

Besides this real-time feedback, the system makes available on-demand statistical feedback. This feedback takes the form of graphical representation of the student's progress through the course materials, using chart diagrams (e.g., line chart, column chart, and pie chart). In addition, it comprises pages with relevant information concerning the student's learning sessions with the system (Fig. 4). Besides the abovementioned information about tests (Fig. 4C and 4D), it provides information about each concept the student has passed (Fig. 4A) as well as the lessons passed for each concept (Fig. 4B).

After the completion of each concept, the system has to check the student's performance, and change the instructional plan accordingly. This change is reflected in the Content menu (Fig. 1A). The system also provides an annotated list of learnt concepts in a form of a navigation menu (Fig.1B). The annotations are used to indicate how well the student has performed on the passed concepts and enabling the student to reflect on his performance. The student can use this navigation menu to quickly access the concept(s) he wants to learn about again.

Evaluation studies

According to Kirkpatrick's model (described in Section 2), evaluation of Results/Organizational impact needs at least a two year evaluation period. In this study it was not possible to delay evaluation for such a long period as in two years time students would have already left the Academy. Regarding that fact, we decided to use the first two levels (reaction and learning) of the Kirkpatrick's model to evaluate the preliminary effectiveness of the DEPTHS system and the accuracy of its assumptions about the student.

The method that we applied to evaluate the system's effectiveness consisted of:

- analyzing the students' reactions to the training program, and
- conducting an experiment with an experimental and two control groups and comparing pre-test and post-test results of these groups.

The method we used to test the accuracy of the student model consisted of comparing the system's assumptions about students' performance level to their results on an external test. In particular, we monitored the progress of each individual student and assessed the quality of tutoring by comparing the student's records in his student model with his results on the external test. We have used the extent of overlapping of these values as the measure of the accuracy of the Student Model.

DEPTHS was evaluated with a group of 42 students who took part in our course on software development in Spring semester 2006 (from February to July 2006). The study was conducted at the Department of Computer Science of the Military Academy in Belgrade in Serbia, with the computer science students of the third academic year. It was organized during the class hours of the Software development course that offered face-to-face mode of learning. The

students were told that their performance observed during this study will influence their marks for the course. In particular, their performance with DEPTHS participated in the final mark with 40%. The students already had some elementary knowledge in the domain of software design patterns from their previous education.

Reaction evaluation

The *participant reaction* is a measure of "customer satisfaction" indicating the level of effectiveness and usefulness of a training program at the time the participants are experiencing it (Antheil & Casper 1986). It can be very useful for determining the quality of the learning process. In particular, if the participants' reaction to the training program is positive, learning is more likely to take place than in case when their reaction is negative. Every teacher is usually anxious to know how students react to his/her teaching efforts and what they have learned. Because the participants are at one place, the required information can be obtained quickly and easily and subsequently used to improve the program. We found that the evaluation of participants' reaction could be very useful for the overall evaluation of DEPTHS for various reasons:

- It helps us know immediately what worked well and what should be improved (or changed all-together).
- It provides information for improving the current system and designing future versions.
- It shows to the participants and stakeholders that we are interested in improving their satisfaction with the course and that we value their input.

The participants' reaction can provide us with valuable information, but we have to be very careful with conclusions, and use this information in combination with the information gathered through other evaluation methods. The reason is that in most cases, there is no definite relationship between what people feel about they have learned and what they actually did learn (Dixon 1990, Le Rouzic & Cusic 1998).

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. We defined the following rules to ensure accurate measurement:

- the interview was designed in a manner that the collected data can be easily tabulated and manipulated by statistical means. Most of the questions were close-ended questions, based on the Likert scale with five responses ranging from 'Very much' (5) to 'Not at all' (1);
- the interview was anonymous;
- we used a few open-ended questions in order to encourage the participants to make additional comments not elicited by the given questions.

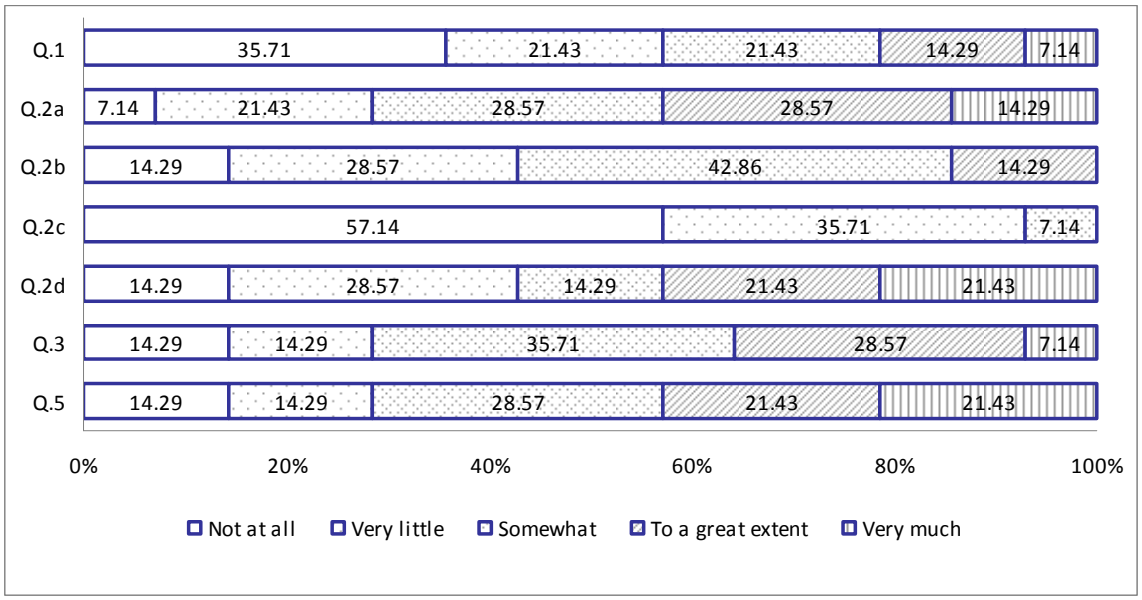
The questions were divided into three sections based on the type of information we were interested in. The first section gathered questions regarding computer-assisted learning. These questions were designed to gather information about the students' previous experience with and opinion about computer-assisted education. For example, we asked the students how often they use computers for learning, and for what purposes: to search the Web for relevant learning materials, to access a Web-based learning application, etc. In addition, we were interested in the students' perception of the relevancy of computer-assisted learning and whether it could (or even should) replace the traditional classroom-based learning.

The questions of the second section were related to the DEPTHS system. We think that a learning system must be easy to use, and must provide students with all tools necessary for successful learning. Students must not be frustrated during the use of the system, because of technical or other problems. We wanted to know to which extent we achieved these goals with DEPTHS and the students' responses to the second section of the questionnaire were supposed to give us the answer. In particular, we wanted to know what they thought about the system, did they experience some technical problems, did they enjoy (and to what extent) using the system, and if they found it difficult to use. We believed that these answers can help us to improve the design and functionalities of the system in the future.

The third section of the questionnaire was aimed at evaluating the course on design patterns offered by the DEPTHS system. The purpose of this section was to measure the students' satisfaction with the content of the course. We wanted to know what they thought about the organization of the course content, and if they found the content too

difficult for them. We also asked students about usefulness of the examples that were used throughout the course, and how the course could be improved.

The interview revealed that, in general, the students did not have much experience using similar systems. For example, the majority of them (92.85%) have either no or very little experience with learning management systems (Fig.5. – Q.2c)(all the questions that are mentioned (Q.x, x=1,..21) in this section are given in the Appendix). We found that most of them are not familiar with electronic educational tools, and do not use the Internet and web-based learning programs for education (Fig.5. – Q1, Q.2a, b, d). However, the students expressed positive attitude toward computer-assisted learning. More than 70% of them think that computer-assisted learning should be made available to supplement traditional lectures and exercises (Fig.5. – Q.3). However, only few (28,58%) students would not mind using learning programs that are in English (Fig.5. – Q.5). Note that the course offered by DEPTHS is in English while all other lectures are offered in Serbian. The students were supposed to be able to follow lectures in English since they had a mandatory 4 semester long English course as a prerequisite for the course in which DEPTHS was used.



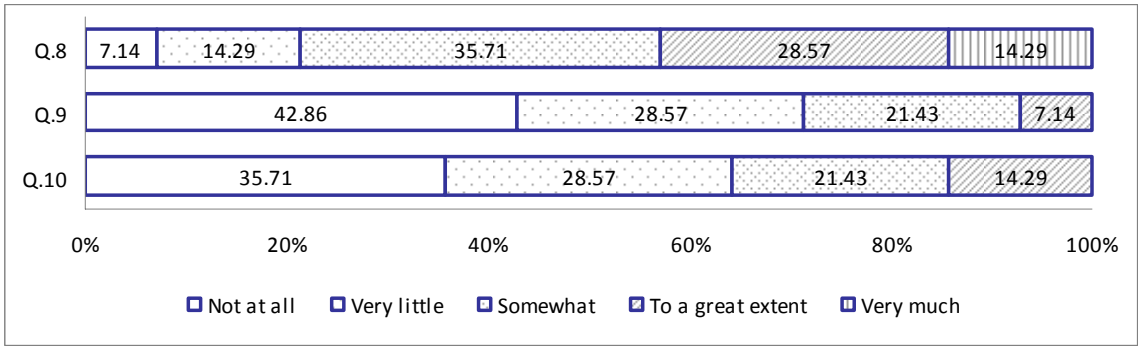
Q.1	How familiar are you with the electronic educational tools?
Q.2	How often do you use a computer for learning? To...
a)	... search the Internet for relevant web pages.
b)	... download papers or similar items (with a known Internet address)
c)	... access a learning management system offering a course of interest
d)	... use a web-based learning application
Q.3	Computer-based learning can replace lectures and exercises
Q.5	I have difficulties or I just don't like working with learning programs which are not in my mother-tongue.

Figure 5. The opinions expressed by the interviewed students about computer-assisted learning in general

The DEPTHS system received high marks from the students. The majority of the students (78.57%) felt that the DEPTHS system is very easy to use (Fig.6. – Q.8), only 7.14% found navigational tools confusing (Fig.6. – Q.9) and most of them (85.71%) did not experience either any technical problems (64.28%) or at least any significant problems (21.43%) while using the system (Fig.6. – Q.10).

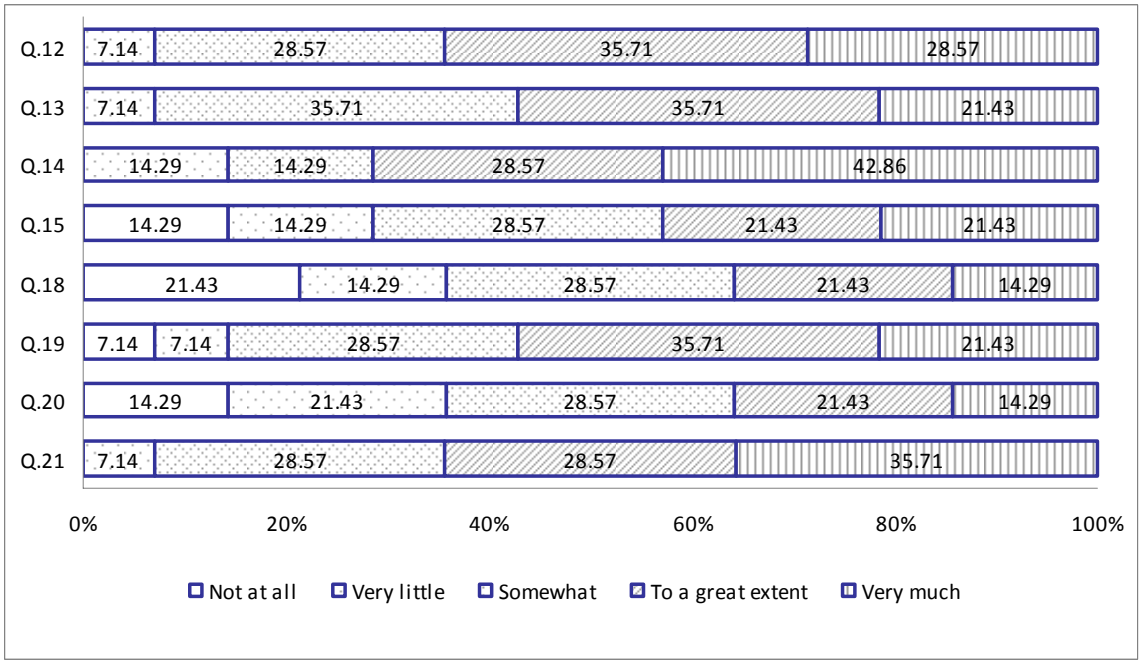
More than 90% of the students thought that the learning goal, outcomes, objectives and expectations for the course taught by the DEPTHS system were clearly stated and that the offered course met them (Fig.7. – Q.12 and Q.13). The majority of the students (85.71%) found the course content well organized (Fig.7. – Q.14), but some of them (28.58 %) thought that the amount of content was not suitable for the length of the course (Fig.7. – Q.15). Nearly two

thirds of the interviewed students (64.28%) reported that they were encouraged to take responsibility for their own learning (Fig.7. – Q.18). During the course, most of the students (84.72 %) often had a clear idea of where they were going and what was expected from them (Fig.7. – Q.19). Only 35.72% of the students reported that they were not given enough choices over how they are going to learn in the course (Fig.7. – Q.20). The majority of the students acknowledged that they had learned a lot about design patterns from the DEPTHS system (92.86 %) (Fig.7. – Q.21).



Q.8	Do you find the system easy to use?
Q.9	Do you find navigational tools confusing?
Q.10	Did you experience any technical problems while using the system?

Figure 6. The students' responses to the questions concerning the DEPTHS system



Q.12	The learning goal, outcome, objective, and/or expectations for the course were clearly stated.
Q.13	The offered course met the stated learning goal, outcome, and/or objective
Q.14	The course content was well organized.
Q.15	The amount of content was suitable for the length of the course.
Q18	I was encouraged to take responsibility for my learning.
Q.19	During the course, you usually had a clear idea of where you're going and what was expected from you.
Q.20	Students have a great deal of choice over how they are going to learn in this course.
Q.21	To what extent do you feel you have learned from the program?

Figure 7. Chart diagrams presenting the students' answers regarding the course taught by DEPTHS

The students were encouraged to report other advantages and disadvantages of learning with DEPTHS through the given open-ended questions. Having analyzed their answers to these questions, we realized that the majority of the students perceived good organization of the course content and convenient access to the relevant online resources and multimedia materials as the primary advantages of learning with DEPTHS. In addition, two students reported a great satisfaction with the feedback they were receiving during learning sessions. Three students found tests provided at the end of each concept as a great advantage of the system.

Students also noted some negative sides of learning with DEPTHS, which were primarily related to the technical problems that some of them had experienced, and the lack of support for collaborative learning. For example, one student reported that the problems with local area network obstructed his learning with DEPTHS. Four students reported that they would appreciate if they were enabled to communicate with each other or with the teacher during the session. Two students reported the lack of students' forum to support learning.

To sum up, the main advantages of the DEPTHS system, as perceived by the interviewed students, are a good organization of the course material, simplicity of the use and its ability to let students work at their own pace instead of learning with a tutor led session. Students also found that the feedback messages, which they were receiving during learning sessions, and advice about what should be learned next, were very useful. On the other hand, the major weakness of DEPTHS is its incapacity to support student-student and student-teacher interactions. In addition, the students mentioned that the system could be improved by adding more functionality for knowledge assessment.

Learning evaluation

For this evaluation study we decided to use 'nonequivalent comparison-group design' (Marczyk et al. 2005), one of the most commonly used quasi-experimental designs for determining system effectiveness. In order to comply with the requirement of this design, we attempted to select groups of students that are as similar as possible. We were constrained by the number of students and teachers in the Military academy. As this experiment was organized as a part of the regular classes of the Software development course, we were bound to engage all 42 students of the third academic year who were eligible to attend these classes. Since computer laboratories at the Military academy have 15 workstations for students, we had to limit the group size to the maximum number of 15 students. In such a limited environment an inadequate selection of participants (i.e. students) for the experiment is highly probable. In order to prevent this selection threat (an incorrect selection would obscure the experiment results), we decided to use two control groups, rather than the usual one.

Thus, one experimental-group and two control-groups, with 14 students each, were involved. The groups were created based on the students' final grades in the courses related to software engineering. Our presumption was that these grades indicate the students' general knowledge of the software engineering domain (column 'School grade' in Table 1). In addition, all of the students were tested at the outset of the study (in the so called pre-test phase) to evaluate their knowledge of the subject domain (i.e., software design patterns). The pre-test allowed us to measure differences between groups before their exposure to the intervention. This way we substantially reduced the threat of selection bias, since the pre-test revealed whether and to which extent the groups differed on the dependent variable (i.e., knowledge level of the subject domain) prior to the intervention.

The pre-test consisted of 16 multi-choice and free form questions. The questions were designed to test the students' elementary knowledge of software engineering in general and design patterns in particular. In addition, this test was aimed at identifying the students' expectations from the course and its delivery mode as well as their previous experience in the subject matter.

Table 1. A comparative overview of the students' school grades and their pre-test and post-test results

Group	School grade (5-10)	Pre-test (1-100)	Post-test (1-100)	Pretest/Posttest Difference	Statistical Significance
experimental group A	7.88	57.21	76.57	22.93	P<0.01
control group B	7.86	55.86	70.21	16.14	P<0.01
control group C	7.88	58.21	68.14	10.36	P<0.01

Throughout the semester, students in the experimental group (group A) were learning exclusively with the DEPTHS system, whereas students in the control groups (groups B and C) were learning in the traditional way. Three teachers were involved in this experiment, one for each group. Teachers in the control groups employed traditional methods (including multimedia resources), while the teacher of the experimental group was helping students in using the DEPTHS system and by providing assistance when needed.

The course consisted of 10 concepts (i.e., software design patterns) which were gradually introduced, one after the other. During the semester, the system was creating a test for each domain concept (i.e., design pattern) and assessing the knowledge of the students from the experimental group based on the test results (as explained in Section 4.4).

At the end of the semester, the students from each group had to take part in a test created by a human teacher. This type of test (often referred to as post-test) is used to appraise the learning effectiveness after the learning process took place. It consists of a number of questions aimed at a detailed evaluation of a number of issues related to the quality of the learning material and the delivery mode.

In order to analyze the tests' results, we decided to use a paired t-test to compare differences between pretest and post-test scores for each group (O'Rourke et al. 2005). The obtained results are presented in Table 1. The table reveals that in group A (students who learned with DEPTHS), the pretest score was 57.21, the post-test score was 76.57, and the scores showed significant improvement ($t=-9.19$, $P<0.01$). In group B, the pretest score was 55.86, the post-test score was 70.21, and the scores also showed significant improvement ($t=-5.88$, $P<0.01$), but in the lesser extent than in group A. Similar results were observed in the other control group (group C): the pretest score was 58.21, the post-test score was 68.14, and even though the scores showed significant improvement ($t=-5.92$, $P<0.01$), the improvement was not that great as in group A. In a nutshell, the statistical results showed that the difference between the pretest and post-test scores was greater in group A than in groups B and C. The test score improvements in groups B and C were similar. These findings indicate that DEPTHS has a beneficial effect on the students' learning and brings in improvements in performance over time.

"One-way analysis of variance (One-way ANOVA)" statistical model (Davies 2000) was used to identify differences among the three groups involved in the experiment. This model was chosen, since it proved as the most appropriate method of statistical analysis of experiments involving more than two groups. It can be used to test the hypothesis that the means among two or more groups are equal, under the assumption that the sampled populations are normally distributed. Our null hypothesis was stating the research question as "no difference" between the experimental group and the control groups.

Table 2. ANOVA table shows statistical analysis of the post-test scores among three groups

Groups	Count	Sum	Average	Variance
Group A	14	1072	76.57143	192.7253
Group B	14	983	70.21429	258.1813
Group C	14	954	68.14286	233.0549

Source of Variation	Sum of Squares	Degrees of Freedom (df)	Mean Square	F-value	P-value	F-critical
Between Groups	540.1429	2	270.0714	1.18459	0.316636	3.238096
Within Groups	8891.5	39	227.9872			
Total	9431.643	41				

f (F-value)	-	measurement of distance between individual distributions
df (degree of freedom)	-	an equivalent of currency in statistics – you earn a degree of freedom for every data point you collect, and you spend a degree of freedom for each parameter you estimate
P (P-value)	-	the probability of obtaining a result at least as extreme as a given data point, assuming the data point was the result of chance alone

Having applied the One-way ANOVA model on the pre-test results, we found that these scores did not differ much between the three groups (f=0.108, df=2, P= 0.89). Table 2 presents the results of applying the One-way ANOVA model on the post-test results. Post-test scores among three groups showed no significant differences (f=1.18, df=2, P=0.32), as well (Brown et al 2006). As F-value is lower than the F-critical value, we concluded that statistically there is no significant difference, and the score differences could be explained the best as being present by chance. The outcome of our test was that we accepted the null hypothesis that the students in the experimental group will perform in the similar manner as students who learned in the traditional way (groups B and C).

Even though the results of One-way ANOVA can bring conclusion that DEPTHS does not lead to significant improvements over the traditional learning, we are encouraged with the fact that our system even in its early stages has better results than traditional learning. In particular, the results of t-test performed on the experimental group have shown significant progress of those students who learned with DEPTHS. Moreover, the qualitative results however indicate that the students prefer this kind of learning over the traditional learning of design patterns. Our experiences from this evaluation will lead to many improvements of the system which will make it far more successful in the future.

Student model assessment evaluation

To evaluate the accuracy of the DEPTHS's student knowledge assessment we compared the students' knowledge of the domain as captured in their student models with their results on an external test, performed after the course was completed. For each student, the DEPTHS system keeps track of all the tests he has solved throughout the course (i.e., after having learned each domain concept) and evaluates his overall knowledge of the domain from the tests' results. The external test was designed by the group of teachers, experts in the domain of software engineering. It was given to the students at the end of the course to assess their knowledge of the whole course content.

The results of the 14 participants from the experimental group were analyzed. The questions of the external test were divided into 10 groups, each group covering one domain concept, so that, based on the test results, we could assess the students' knowledge of each concept individually. Comparing the results of the external test with the student model data (Fig. 8), we found that the external test shows slightly lower knowledge level (15.43%). That difference was expected, because the system (i.e. its student modeling component) assesses student knowledge on each concept immediately after the student has learned it, whereas the external test assesses student knowledge at the end of the whole course when a number of other variables affect the final results. For example, students could not use DEPTHS out of the classroom, but we could not control if they are using some other sources of information (literature or the Internet). Moreover, when estimating the student's knowledge of a concept, the system takes the best score over all the test attempts related to that concept. However, if instead of taking the best result we toke the average test result for a concept (from system's logs on each concept), we get results that are 11.37% lower than those on the external test. As the result of the student model assessment evaluation performed on DEPTHS, we finally concluded that the comparison between the external test and student model could not be realistic indicator of the student model accuracy, as it is performed during the long period of time (one semester), while system assesses student knowledge immediately after the learning of each concept. Therefore, we decided to analyze the system's log data in order to get some more information about the student model used.

By analyzing the system's log data we found that at the beginning of the course students tend to spent more time on each lesson, give more attempts at quizzing and were often revisiting previously learned concepts. The majority of the students (87%) attempted to solve the tests more than once during the first quarter of the course, and 34% gave three or more attempts. This trend was diminishing as the course proceeded. At the last quarter, 49% of the students made two or more attempts and only 11% tried to do the test more than twice. We think that this decrease in the number of test reatempts is due to the fact that as the end of the semester was approaching, the students were having increasingly more obligations on other courses and less time to dedicate to learning with DEPTHS. This is a well known phenomenon, present at the Military Academy for years.

We found an additional interesting issue by analyzing the system's log data. Some questions, annotated by their creators as very easy, were not performed well by students. Contrary, some other questions described as very difficult were surprisingly well done. Having analyzed this issue, we concluded that it originates from the fact that the Diagnosis module uses data about a question's difficulty level and time necessary to solve it provided by a

human teacher or question developer. However, a teacher can make a mistake when estimating the difficulty level of and the required time for a specific question. Such a mistake would lead to the inappropriate adaptation of test, for example, if a teacher describes a difficult question as very easy, the system will use it to test student with low level of knowledge. Based on this conclusion we have improved our Diagnosis module. It now performs analysis of the system's logs data, compares it with question's difficulty level and time necessary to solve the question, and changes it if a variance exists. We strongly believe that this model could significantly improve older solution. However, we have not evaluated this model yet.

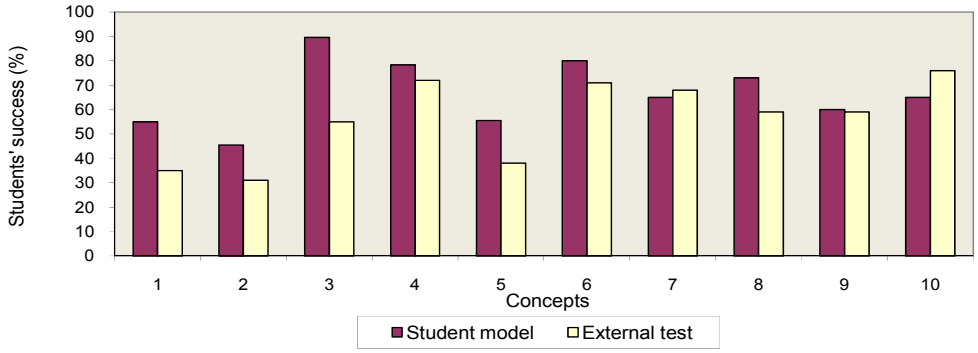


Figure 8. Graphically presented comparison of the student's knowledge as estimated by the DEPTHS's student model and the results of the external test

Related Work

The main problem in evaluation of intelligent tutoring systems is that the available literature does not suggest any clear guidelines for an evaluator which methods to use in particular context. In ITS community, the common practice is to perform experiments with a particular set of users. However, many different approaches to performing these experiments have been proposed by different authors.

Comparing to our work, we found a similar research presented in (Mitrovic et al. 2002). In this work the authors presents the results of three evaluation studies performed on SQL-Tutor, an ITS for the domain of SQL database query language. These studies were primarily focused on the assessment of the Constraint-Based student model that SQL-Tutor uses (i.e. how well this model supports student learning). In addition, the studies were aimed at collecting students' opinion of the system's usefulness. Overall, the results of these studies were positive. On the regular classroom examination that followed the experiment, students who had been learning with SQL-Tutor significantly outperformed students who learned in the traditional manner. However, this experiment was not controlled, and because some students kept on using the system after the class hours, the results can not be considered as a definitive proof of the system's quality. Contrary to this, students who worked with DEPTHS could not use the system out of the classroom. However, since our experiment lasted for a long period of time, other variables could have affected our results.

In (Weibelzahl 2003) authors describe an approach for the evaluation of adaptive learning systems in general, which assumes evaluation on four levels: evaluation of the input data, evaluation of the inference mechanism, evaluation of the adaptation decisions, and evaluation of the interaction. They have applied this model to an adaptive learning system, called the HTML-Tutor, to demonstrate the usefulness of the suggested approach. Similarly to the DEPTHS, they have performed the evaluation of the accuracy of student model. They suggested two methods to perform this: by comparing its assumptions to an external test and by comparing its assumptions to the actually displayed behavior of the student. Moreover, they have focused also on the evaluation of the inference mechanism. Their results show that the evaluation of the accuracy of the system's assumptions about the student could point to possible improvements of the inference mechanism. However, they did not present any results about the overall system effectiveness.

Another similar evaluation was reported in (Miller & Butz 2004). This study has evaluated the usability and effectiveness of Interactive Multimedia Intelligent System (IMITS), a system designed to tutor second year electrical engineering undergraduates. IMITS was evaluated from two perspectives, usability of the software and effectiveness of the software. The Usability Questionnaire was used to gather information concerning students' reactions to the software. Also, usability data was obtained from the system's log files. Similar technique as in DEPTHS was used to examine the impact of IMITS on student learning. The authors have used a quasi-experimental design, with a control and an experimental group of students. Analysis of variance (ANOVA) was used to test the hypothesis that the students who were using IMITS learned more than their counterparts under control conditions. Overall, students' responses were favorable. IMITS improved performance on at least one classroom achievement measure. Regression analyses revealed that the more students used IMITS to learn some engineering concept (with usage defined as percentage of the questions encountered on a particular engineering concept), the better they learned that concept. Comparing to IMITS, we used the similar evaluation design in DEPTHS. We have reported on the systems usability and its effectiveness. Both evaluations have shown that the students' responses were favorable. However, we go step further in evaluation of the system effectiveness. Contrary to IMITS evaluation of DEPTHS introduce a comparison of pre-test and post-test results in experimental group that lacks in IMITS.

Our approach to evaluation of DEPTHS was influenced by the work reported in (Barker et al. 2002). They have applied a qualitative approach when evaluating their adaptive multimedia tutoring system. The authors have pointed out that an evaluation in real-world settings presents so many uncontrollable variables that traditional quantitative methods can give misleading results. They have suggested combining qualitative and quantitative analysis and have shown the advantages of this combined approach in real-world contexts. Their quantitative ANOVA results have shown that certain differences in mean values are unlikely to stem from random variations among students. Such strong rejection of the random-variations hypothesis would be impossible with a purely qualitative analysis. While agreeing with this approach, we have also adopted Kirkpatrick's model of evaluation as a foundation of our evaluation design in DEPTHS.

We found an interesting approach for evaluating the accuracy of the student model in (Millan & Perez de la Cruz 2002). The approach is based on the usage of simulated students since the cognitive state of a simulated student, unlike that of a real student, can be precisely determined, so the evaluation results are more accurate. However, as the authors pointed out, simulations could not provide the same degree of validity as experiments with real human students. This work is a good example of separating the evaluation of the accuracy of a student model from the evaluation of the adaptations efficiency based on the student model. By evaluating an intelligent tutoring system modularly with simulated students, the effects of each component can be separated from one another and the weakest component(s) could be easily identified and improved in order to increase the effectiveness of an adaptive system. In the future, we are planning to apply the similar approach with DEPTHS, in order to evaluate its modules, namely student model, pedagogical module, as well as the accuracy of student model assessment and adaptation effectiveness. We believe that this approach could give us valuable results on design and efficiency of each DEPTHS's module.

Conclusions

This paper presents the evaluation of DEPTHS, an intelligent tutoring system (ITS) for teaching/learning software design patterns. The adopted evaluation approach is quite general, and can be equally well applied for evaluation of other ITSs. In particular, we used first two levels (reaction and learning) from the well-known Kirkpatrick's model (Kirkpatrick 1979). The conducted evaluation studies targeted primarily the effectiveness of the DEPTHS system as well as the accuracy of its assumptions about the students' knowledge level. The semester-long evaluation study has provided us with insights into strengths and weaknesses of the DEPTHS system. It has also made clear directions for future actions.

We reported several advantages of the DEPTHS system over the traditional approach to teaching/learning design patterns. Students who learned with DEPTHS found that the system helped them to learn a lot about design patterns. They were especially pleased with the system's ability to provide them with many useful information, feedback messages and advice for further work. Students' responses indicated the need for regular communication with teachers and other students as the underpinning priorities for successful completion of online learning.

To test the learning effectiveness of DEPTHS, we used t-test and compared the pre-test and post-test results of an experimental and two control groups. The statistical analysis showed that the change from the pre-test to the post-test results was greater in the experimental than in the control groups. These findings indicate that students who learned with DEPTHS performed better than students who learned in the traditional way and that learning with DEPTHS brings in improvements in performance over time. We have also applied the One-way ANOVA statistical model to test for differences among the three study groups. The outcome of this test was that we accepted our null hypothesis that the students in the experimental group will perform as well as students who learn in the traditional way. This finding is obviously inconsistent with the result of the t-test, and another confirmation of the difficulty of accurately measuring the effectiveness of a certain educational tool on the students' performance. However, we are encouraged with the fact that this system even in its early stages has better results than traditional learning.

Finally, we compared the students' knowledge of the domain as captured in their student models with their results on post-test in order to evaluate DEPTHS's ability to accurately assess student knowledge of the subject domain. We found that the proposed student model does reflect the students' knowledge of the subject matter, regardless the observed slight difference between the end-test results and the student model.

Regarding our future work, there are several directions to be explored. Our first goal concerning further improvements of the DEPTHS system includes enabling the Student Model to store not only student's knowledge, but also students' cognitive and behavioral characteristics, such as memory capacity and motivation. This data will facilitate even more effective adaptation of the learning material. We are also planning to support interactions among students and the tutor in a group discussion environment, as well as to enable integration of additional supporting software tools, such as ArgoUML (http://argouml.tigris.org/). Regarding that, we are currently developing a Semantic web-based framework that integrates a set of existing tools in order to enhance students learning experience when collaboratively learning about Software patterns on the Web (Jeremic et al. 2008).

References

Antheil, J.H., & Casper, I.G. (1986). Comprehensive evaluation model: A tool for the evaluation of nontraditional educational programs. *Innovative Higher Education, 11* (1), 55-64.

Barker, T., Jones, S., Britton, C., & Messer, D. (2002). The Use of a Co-operative Student Model of Learner Characteristics to Configure a Multimedia Application. *User Modeling and User-Adapted Interaction, 12* (2-3), 207-241.

Beaumont, I. (1994). User modeling in the interactive anatomy tutoring system ANATOMTUTOR. *User Models and User Adapted Interaction 4* (1), 21-45.

Brown, E., Stewart, C.D., Tim, J.B. (2006). Adapting for Visual and Verbal Learning Styles in AEH. *Proceedings of the 6th IEEE ICALT Conference*, CA: IEEE Computer Society, 1145-1146.

Brusilovsky, P. (1996). Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction, 6*, 87-129.

Burton, R.R., & Brown, J.S. (1976). A tutoring and student modeling paradigm for gaming environments. *ACM SIGCSE Bulletin, 8* (1), 236-246.

Davies, P.L. (2000). *The one-way analysis of variance*, retrieved March 8, 2009 from http://www.stat-math.uni-essen.de/~davies/one-way.ps.gz.

Devedzić, V. (2001). Knowledge Modeling – State of the Art. *Integrated Computer-Aided Engineering, 8* (3), 257-281.

Dixon, N.M. (1990). The Relationship Between Trainee Responses on Participant Reaction Forms and Posttest Scores. *Human Resource Development Quarterly, 1* (2), 129-137.

Endres, G.J., & Kleiner, B.H. (1990). How to measure management training and development effectiveness. *Journal of European Industrial Training, 14* (9), 3-7.

Harrer, A., & Devedzić, V. (2002). Design and analysis pattern in ITS architecture. *Proceedings of the ICCE 2002 Conference*, CA: IEEE Computer Society, 523-527.

Hartley, J.R., & Sleeman, D.H. (1973). Towards intelligent teaching systems. *International Journal of Man-Machine Studies, 5*, 215-236.

Holzkamp, K. (1995). *Lernen: Subjektwissenschaftliche Grundlegung*, Frankfurt, Germany: Campus Verlag.

Jeremić, Z. (2005). *An Intelligent Tutoring System for learning Software Patterns* (in Serbian), Master Thesis, Belgrade, Serbia.

Jeremić, Z., & Devedžić, V. (2004). Design Pattern ITS: Student Model Implementation. Proceedings of the IEEE ICALT 2004 Conference, CA: IEEE Computer Society, 864-865.

Jeremić, Z., & Devedžić, V. (2004). Student Modeling in Design Pattern ITS. *Paper presented at the KES 2004 Conference*, September 22-24, 2004, Wellington, New Zealand.

Jeremić, Z., Devedžić, V., & Gašević, D. (2004). *An Intelligent Tutoring System for Learning Design Patterns*, retrieved March 20, 2009, from http://www.ii.uam.es/~rcarro/AHCW04/Jeremic.pdf.

Jeremić, Z., Jovanović, J., & Gašević, D. (2008). A Semantic-rich Framework for Learning Software Patterns. *Paper presented at the IEEE ICALT 2008 Conference*, July 1-5, 2009, Santander, Spain.

Kay, J., & Kummerfeld, R.J. (1994). *An Individualised Course for the C Programming Language*, retrieved March 20, 2009, from http://www.cs.su.oz.au/~bob/kay-kummerfeld.html.

Kim, D.W. (1995). WING-MIT: Das auf einer multimedialen und intelligented Benutzerschnittstelle basierende tutorielle Hilfesystem für das Werkstoffinformationssystem WING-M2. *Paper presented at the Workshop Adaptivität und Benutzermodellierung in interactiven Systemen*, October 11-13, 1995, München, Germany.

Kirkpatrick, D.L. (1979). Techniques for evaluating training programs. *Training and Development Journal, 33* (6), 78-92.

Kirkpatrick, D. (1994). *Evaluating Training Programs*, San Francisco, CA: Berrett-Koehler.

Le Rouzic, V., & Cusick, M.C. (1998). Immediate Evaluation of Training Events at the Economic Development Institute of the World Bank: Measuring Reaction, Self-Efficacy and Learning in a Worldwide Context. *Paper presented at annual American Evaluation Association Meeting*, November 4-7, 1998, Chicago, USA.

Marczyk, G., DeMatteo, D., & Festinger, D. (2005). *Essentials of Research Design and Methodology*, NJ: John Wiley.

McEvoy, G.M., & Buller, P.F. (1990). Five uneasy pieces in the training evaluation puzzle. *Training and Development Journal, 44* (8), 39-42.

Millan, E., & Perez-de-la-Cruz, J.L. (2002). A Bayesian Diagnostic Algorithm for Student Modeling and its Evaluation. *User Modeling and User-Adapted Interaction, 12* (2-3), 281-330.

Miller, S., & Butz, B. (2004). *Evaluation of an Interactive Multimedia Intelligent Tutoring System*, retrieved March 20, 2009, from http://www.temple.edu/imits/shock/Evaluation2.pdf.

Miskelly, T. (1998). Interactive Student Modeling. Paper presented at the 36th Annual ACM Southeast Regional Conference, April 1-3, 1998, Marietta, GA, USA.

Mitrovic, A., Marting, B., & Mayo, M. (2002). Using Evaluation to Shape ITS Design: Results and Experiences with SQL-Tutor. *User Modelling and User Adapted Interaction, 12* (2-3), 243-279.

O'Rourke, N., Hatcher, L., & Stepanski, E.J. (2005). *A Step-by-Step Approach to Using SAS® for Univariate & Multivariate Statistics* (2nd Ed.), Cary, NC: SAS Publishing.

Phillips, J. J. (1997). A rational approach to evaluating training programs including calculating ROI. *Journal of Lending and Credit Risk Management, 79* (11), 43-50.

Prentzas, J., Hatzilygeroudis, I., & Garofalakis, J. (2002). A Web-Based Intelligent Tutoring System Using Hybrid Rules as Its Representational Basis. *Lecture Notes In Computer Science, 2363*, 119-128.

Rising, L. (1998). *The Patterns Handbook: Techniques, Strategies and Applications*, New York: Cambridge University Press.

Reeves, T.C., & Hedberg, J.G. (2003). *Interactive learning systems evaluation*, Englewood Cliffs, NJ: Educational Technology.

Weibelzahl, S., Weber, G. (2003). Evaluation the Inference Mechanism of Adaptive Learning Systems. *Lecture Notes in Artificial Intelligence, 2702*, 154-168.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*, Los Altos, CA: MorganKaufmann.

APPENDIX A

Computer Science and Software engineering,
Military academy in Belgrade, Department of Defense
Serbia 2006

COURSE EXPERIENCE QUESTIONNAIRE

This questionnaire is intended at collecting the information which will help us to evaluate the DEPTHS learning system from the perspective of its technical characteristics and performance as well as from the perspective of the quality and adequacy of the course it provides.

The information you will provide in this questionnaire will be kept strictly confidential. Neither your name nor any details that might be used to identify you will be published. We guarantee that the data will be used exclusively for requirements and statistical analysis.

1=Never/Not at all/Not good at all 5=Always/Very much/Extremely good

	SECTION: General, e-learning related questions					
1.	How familiar are you with the electronic educational tools?	1	2	3	4	5
2.	How often do you use a computer for learning? To...					
	a)...search the Internet for relevant web pages.	1	2	3	4	5
	b)...download papers or similar items (with a known Internet address)	1	2	3	4	5
	c)...access a learning management system offering a course of interest	1	2	3	4	5
	d)... use a web-based learning application	1	2	3	4	5
3.	Computer-based learning can replace lectures and exercises	1	2	3	4	5
4.	Computer-based learning should be used only to supplement classroom lectures and exercises.	1	2	3	4	5
5.	I have difficulties or I just don't like working with learning programs which are not in my mother-tongue.	1	2	3	4	5
6.	Web-based learning should be nothing more than the distribution of notes over the Internet.	1	2	3	4	5
	SECTION: Evaluation questions regarding the system					
7.	Did you enjoy learning with DEPTHS?	1	2	3	4	5
8.	Do you find the system easy to use?	1	2	3	4	5
9.	Do you find navigational tools confusing?	1	2	3	4	5
10.	Did you experience any technical problems while using the system?	1	2	3	4	5
11.	If your answer to the previous question was 5 or 4 please describe the kind of problems you had.	_____				
	SECTION: Evaluation questions regarding the course					
12.	The learning goal, outcome, objective, and/or expectations for the course were clearly stated.	1	2	3	4	5
13.	The offered course met the stated learning goal, outcome, and/or objective	1	2	3	4	5
14.	The course content was well organized.	1	2	3	4	5
15.	The amount of content was suitable for the length of the course.	1	2	3	4	5
16.	The content of this course was too difficult.	1	2	3	4	5
17.	Helpful examples were used.	1	2	3	4	5
18.	I was encouraged to take responsibility for my learning.	1	2	3	4	5
19.	During the course, you usually had a clear idea of where you're going and what was expected from you.	1	2	3	4	5
20.	Students have a great deal of choice over how they are going to learn in this course.	1	2	3	4	5
21.	To what extent do you feel you have learned from the program?	1	2	3	4	5
22.	This course can be improved by:	_____				
23.	Please feel free to make any additional comments about this course.	_____				