

An Open Integrated Exploratorium for Database Courses

Peter Brusilovsky*
1(412)624-9404
peterb@pitt.edu

Sergey Sosnovsky*
1(412)624-9437
sosnovsky@gmail.com

Danielle H. Lee*
1(412)624-9437
suleehs@gmail.com

Michael V. Yudelson*
1(412)624-9437
mvy3@pitt.edu

Vladimir Zadorozhny*
1(412)624-9411
vladimir@sis.pitt.edu

Xin Zhou*
1(412)624-9437
reniorc@gmail.com

ABSTRACT

In this paper, we present an open architecture that combines different SQL learning tools in an integrated Exploratorium for database courses. The integrated Exploratorium provides a unique learning environment that allows database students to take complimentary advantages of multiple advanced learning tools.

Categories and Subject Descriptors

H.5.4 [Hypertext/Hypermedia]: Architectures, Navigation. H.2.3 [Languages]: Query languages.

General Terms

Design, Languages.

Keywords

E-Learning, Learning Environment, Problem-Solving, SQL, Integration.

1. INTRODUCTION

Structured Query Language (SQL) is the most widely-used multipurpose database language. Extensive knowledge and fluency in SQL is mandatory to master modern database technologies. The most reliable way to achieve the required level of SQL fluency is to gain considerable practical experience. Meanwhile, existing SQL teaching methodologies are focused on presenting concepts and selected examples with relatively few practical exercises. These examples and exercises are typically targeting a “classical” subset of SQL knowledge and ignore the considerable number of SQL features which make it a powerful tool for efficient and practical data management. As a result, database graduates require considerable advanced training once they enter the realm of industrial environments.

Several advanced systems were created to support teaching and

learning SQL. These systems can be roughly classified into two groups. One group supports student learning by providing interactive examples, which demonstrate basic concepts of SQL and illustrates the behavior of its components. [1; 2]. These examples are frequently created using multimedia. Another group supports learning by offering students SQL problems and evaluating their solutions [3; 4]. Both approaches have proven effective. For example, SQL-Tutor system demonstrated a significant improvement in problem solving performance on the post-test [4].

However, a single advanced tool, while being efficient and sophisticated, supports only a part of the course and only a single type of educational activities. By complementing each other’s strength, an integrated set of different tools can better support the needs of a particular course. However, the use of such different advanced tools in the context of one class is hardly possible. Simple educational content (such as HTML pages) can be easily integrated in one system by importing it as a part of a courseware package, or by providing links to it. In contrast, most of the rich interactive activities cannot be copied, or referenced, so easily. For example, SQL-Tutor problems have to be delivered and evaluated dynamically by a dedicated server. Before a student starts working with a problem, she has to be authenticated. After the problem evaluation is performed, the system should store the results of the student’s work to support learning continuity and proper adaptation. The teacher may want to monitor the student’s work. The students may want to observe the growth of their knowledge over the use of various learning activities. Other components of an integrated system may want to access this data to adapt to an individual student. The ability to access and use this information is one of the keys to the success of advanced content in modern E-learning.

To integrate several advanced interactive systems, one has to resolve a number of problems, such as: *How to allow the student to access activities in different systems without the need for multiple logins? How the student’s actions, observed by one system, can be stored in such a way that another system can utilize it? How the current knowledge state of a student can be derived from a log of his activities in several different systems?* In this paper, we report our initial progress towards resolving these problems in the framework of the Integrated Exploratorium for Database Courses. The Exploratorium has been designed to combine multiple types of advanced interactive content served by

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ITiCSE’08, June 30 - July 2, 2008, Madrid, Spain.
Copyright 2008 ACM 978-1-60558-115-6/08/06...\$5.00.

* School of Information Sciences, University of Pittsburgh, 135 N. Bellefield Ave. Pittsburgh, PA 15232 USA

independent content servers. Currently, the Exploratorium integrates three types of interactive learning activities (annotated examples, self-assessment questions and SQL labs) available on the course portal. Several adaptive hypermedia techniques are employed to provide personalized access to educational content. The architecture of the Exploratorium is open and the current version can be easily extended with additional activities and components. This paper presents the components of the Exploratorium as well as its architecture, which allows integration of these components. It also presents the results of the system evaluation in both graduate and undergraduate database classes.

2. SQL EXPLORATORIUM

2.1 Interactive Examples

The SQL Exploratorium offers a large set of interactive annotated SQL examples delivered using the WebEx system [5]. Each example consists of a SQL code fragment with important lines being explained. The WebEx interface allows students to interactively explore code explanations in arbitrary order (See Figure 1). By clicking on a checkbox, a student opens the explanation for the corresponding line of code. The click history is logged in the user model to produce an individual history-enriched environment for every student. Lines that have been explored by the student receive checkmarks. The color of checkboxes indicates how many students in the class accessed these lines of code. The deepening intensity of color indicates a higher number of students who have accessed the line. These adaptive visual cues implement social navigation support [6], helping students to explore relevant code fragments.



Figure 1: A WebEx example accessed through the Knowledge Tree portal.

There are 64 available SQL examples providing students with access to more than 300 annotated lines of code. Each example targets one or more SQL concepts. The conceptual model of every example is expressed formally as a set of corresponding concepts from the SQL ontology (see Section 3). This allows the user modeling component of the Exploratorium to track the progress of the user's knowledge and to adapt to that level of knowledge.

2.2 SQL Knowledge Tester

The SQL-KnoT (Knowledge Tester) offers students an opportunity to test and practice their problem-solving skills. It generates questions that require a student to write an SQL query for a sample database, evaluates the correctness of student's answer, and provides a student with feedback (see Fig. 2 for a

sample SQL-KnoT question). SQL-KnoT is similar to other web-based testing systems [3; 4], but uses a novel approach to question generation and answer evaluation. Every time a student accesses a SQL-KnoT question, the actual question text is generated by the corresponding template from the set of predefined databases. When SQL-KnoT evaluates a student's answer, it randomly generates several starting states of the question database. After that, SQL-KnoT compares the result produced by the student solution with the result produced by the pre-stored correct answer. For the needs of our courses, we have developed about 50 templates capable of generating over 400 actual questions. SQL-KnoT questions are indexed with concepts from the same SQL ontology. This ensures consistent tracking of student knowledge based on the work with relevant SQL-KnoT questions and WebEx examples.

Question:

Based on the tables below, write the required SQL expression.

Task:

Show all the information contained in table "category".

Table Name Schema & Sample Data (click +/- to show/hide sample data)

actor(+)	actor_id	first_name	last_name	last_update					
address(+)	address_id	address	address2	district	city_id	postal_code	phone	last_update	
category(-)	category_id	name	last_update						
	1	Action	2006-02-15 04:46:27.0						
	2	Animation	2006-02-15 04:46:27.0						
city(+)	city_id	city	country_id	last_update					
country(+)	country_id	country	last_update						
customer(+)	customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update

Figure 2: A problem presentation in SQL-KnoT.

2.3 Adaptive Navigation for SQL Questions

SQL-KnoT questions are available in two modes:

- in the direct mode (Figure 2) students access the questions in non-adaptive manner, through the corresponding lecture on the learning portal;
- in the adaptive mode (Figure 3) students receive navigation support from QuizGuide adaptive service [7], which helps to choose questions that are appropriate for this particular student.

QuizGuide provides adaptive topic-based access to questions. SQL-KnoT questions are grouped into 20 topics, which comprise the SQL part of the course. To guide the student to the most appropriate topic to practice, QuizGuide annotates the link to each topic with an adaptive "target-arrow" icon. The annotation is based on the two factors:

- the student's personal performance for the topic (the number of arrows in the target icon increases as the student demonstrates current progress with the topic's questions)
- the relevance of the topic to the current goal of the course. (as new topics are introduced in class, topic icons change the color to bright-blue – "current topic", light-blue – "prerequisite for a current topic", or gray – "not-relevant topic"; a crossed icon means that students are not ready for the topic yet).

The QuizGuide service has been successfully used before for helping students to study C programming [7]. Due to the positive experience with this adaptive service, we have decided to employ it as a part of the SQL Exploratorium to facilitate students working with SQL-KnoT problems.

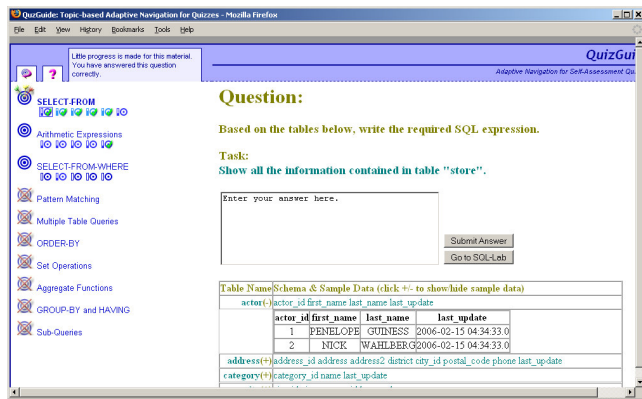


Figure 3: QuizGuide adaptation for SQL-KnoT questions.

2.4 SQL-Lab

SQL-Lab supports interactive exploration of SQL problem scenarios. SQL-Lab interface allows students to formulate queries, observe their results and test performance of SQL scripts. The scripts can be designed by the students themselves, or copied from the set of WebEx examples. When accessed from an SQL-KnoT question, SQL-Lab can be used in a *problem-solving mode*. In this mode, the student can see the question statement, work on the solution, and submit the tested solution to SQL-KnoT (Figure 4).

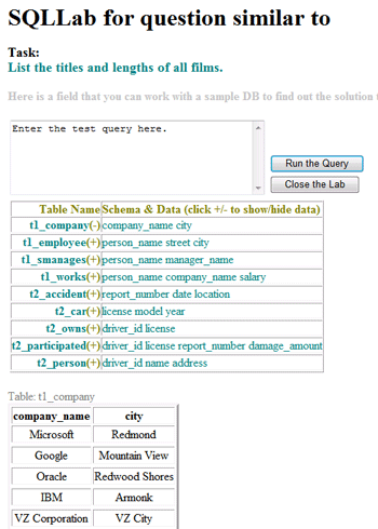


Figure 4. An example of a SQL-Lab screenshot.

2.5 Knowledge Tree Course Portal

The Knowledge Tree Course portal provides access to all learning content and plays the role of a Learning Management System. It allows a teacher to structure the learning content according to the needs of the course. Knowledge Tree is implemented using a common folder-document paradigm. Each course is perceived as a folder that may contain a sequence of folders (for example, lecture folders if the teacher chooses to structure material by lecture).

Lecture folders contain individual resources such as SQL-KnoT problems, WebEx examples or course materials, which the teacher chooses to provide for this lecture. The Knowledge Tree interface is shown in Figure 1. Here, the left frame presents a tree of folders, the main frame shows folder or resource content, and the top frame contains menu with available tools. Knowledge Tree integrates diverse resources together providing single sign-on functionality for its users and convenient views of course material.

3. INTEGRATION APPROACH

3.1 Ontology-based Content Integration

The conceptual integration of the diverse learning content offered by the Exploratorium is based on the central SQL Ontology. The ontology provides the components of the Exploratorium with a common ground for conceptual modeling of SQL domain. This common-ground approach has been previously implemented, for example, in Onto-AIMS [8]. The ultimate goal of the ontology employed for domain modeling in adaptive education is to provide the basis for overlay modeling of student's knowledge. This fact introduces several constraints that should be met by the ontology in order to successfully serve as a domain model. Such ontology must ensure the level of concept granularity necessary for precise assessment and modeling of significant knowledge elements in the learning domain. The ontology relations should follow the principles of formal knowledge representation models to allow automatic unambiguous knowledge propagation between relevant concepts (e.g. the knowledge of the concept *ReferentialIntegrity* can influence the knowledge of its super-concept *ForeignKey*).

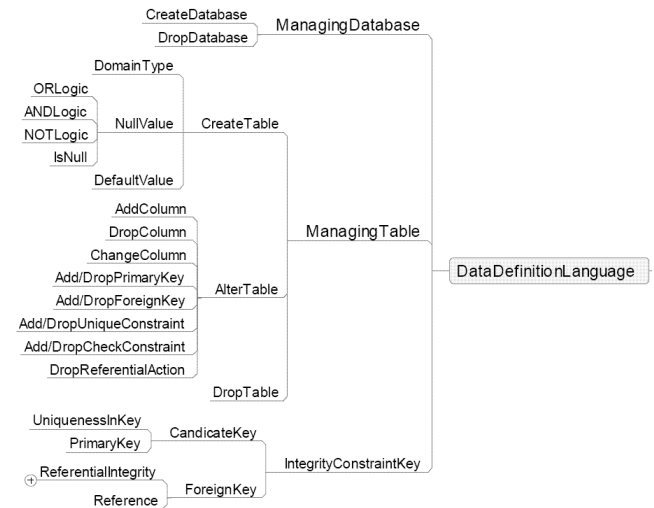


Figure 5. A fragment of SQL Ontology.

We followed these requirements while developing the SQL ontology (Figure 5 visualizes a fragment of this ontology). As we did not use any complex modeling constructs, such as axioms, we did not need the level of expressivity provided by advanced ontology representation languages like OWL. Our ontology is modeled as a RDFS ontology. It consists of 347 RDF triples and defines 166 *rdfs:Class*'s. The complete ontology can be accessed at <http://www.sis.pitt.edu/~paws/ont/sql.rdfs>.

3.2 Open Integration Architecture

The technical basis for the integration of the tools described in this paper is provided by ADAPT² (Advanced Distributed Architecture for Personalized Teaching and Training), which is an extension of KnowledgeTree architecture developed by our group earlier [9] (see Figure 6). The main purpose of ADAPT² is to integrate rich interactive content and provide adaptive functionality for content that lacks it.

The Knowledge Tree portal introduced above is one of the key components of ADAPT². From an architectural point of view, Knowledge Tree serves as an integrator of all educational resources for students. It provides a single sign-on and access point for all WebEx interactive examples, SQL-KnoT problems (whether directly or via QuizGuide), and SQL-Lab. Upon entering the portal, students can follow links to the integrated tools and interact with learning content. QuizGuide serves as intermediate aggregator of SQL-KnoT problems.

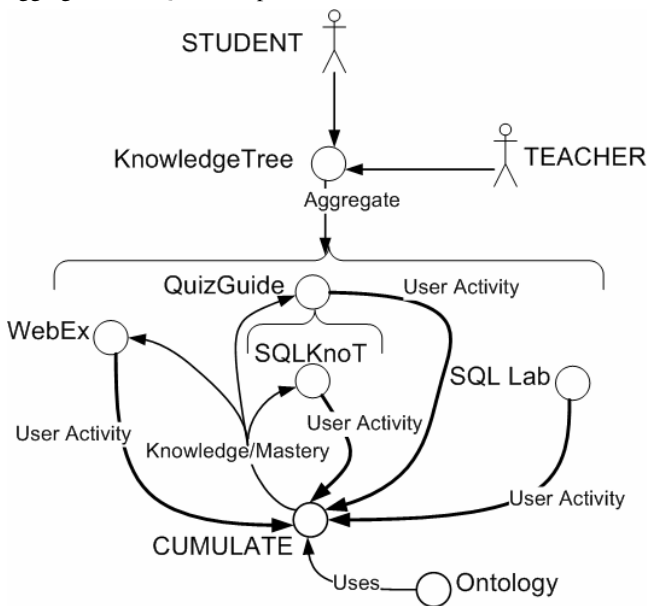


Figure 6. ADAPT2 architecture.

Another important component of ADAPT² is the user modeling server CUMULATE [10]. CUMULATE accepts reports about user activity from educational tools including navigational clicks (WebEx interactive examples), attempts to solve problems (SQL-KnoT/QuizGuide), and application of problem-related skills (SQL-Lab). Once these reports arrive, CUMULATE performs inferences of user knowledge with respect to domain ontology. The result of the inference is the individual and group overlay user models of student knowledge.

Inferences performed by CUMULATE are used by adaptive systems to perform personalization of the presentation. Interactive examples served by WebEx visualize individual and group click traffic. SQL-KnoT employs reports about students' knowledge to tailor the newly-generated problems. QuizGuide annotates links to SQL-KnoT problems with navigational cues. As students work more with the tools and their knowledge grows, the presentation of educational material incrementally changes. The two-way information exchange between interactive learning content and CUMULATE is performed using standard HTTP-based protocols.

4. CLASSROOM STUDIES

The Exploratorium has been introduced to two database courses (one undergraduate – 37 students and one graduate – 36 students) at the School of Information Sciences, University of Pittsburgh. Undergraduate students worked with it over the entire Fall 2007 semester. The graduate class started using the system in the middle of the semester. In both classes, students were encouraged, but not required, to work with the Exploratorium tools. At the end of the semester, students who used the systems sufficiently over the semester received a modest amount of extra credit. We have used following criteria to recognize sufficient usage of Exploratorium:

- at least 20 answers to at least 10 SQL-KnoT questions given over the period of no less than 2 weeks;
- at least 30 lines browsed in at least 10 WebEx examples over the period of no less than 2 weeks.

The total number of students satisfying these criteria for at least one system was 44 (26 for the undergraduate class and 18 for the graduate). The total number of students who tried the Exploratorium was 48. Hence, we have observed a high student retention ratio, which might indicate a high level of satisfaction on the part of the students. On average, 9 out of 10 students who tried the Exploratorium tools continued to work with them. Overall, students gave about 5500 answers to SQL-KnoT Problems and more than 7500 views of the WebEx annotations. Some students used only a single tool; however, the majority of them (about 70%) worked with all systems in a balanced manner. Even in a single session, they often switched between different systems to explore different forms of the content available for the topic of interest. The session analyses shows that out of almost 300 sessions less than a half were dedicated solely to a single system. The majority of sessions have had registered transactions (student interactions) from at least two of the main Exploratorium tools. In every fifth session, students used all three systems. Figure 7 schematically visualizes distribution of registered sessions among the Exploratorium tools. We believe that this data confirms the importance of integration of multiple interactive educational activities in a single system, which is the main idea of the Exploratorium.

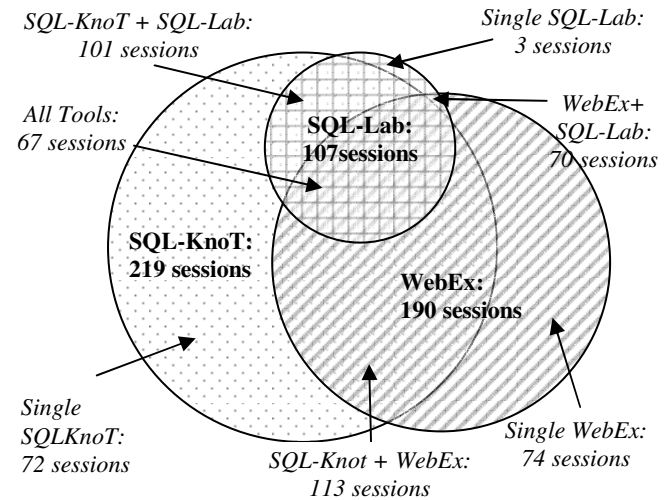


Figure 7. Session Distribution

At the end of the semester, we collected student evaluations of the Exploratorium tools from those undergraduate students who tried any of the Exploratorium components. The results of the questionnaire demonstrate a very positive attitude towards all Exploratorium tools. For instance, 94% of students agree or strongly agree that the use of annotated examples helped them during the course, 69% thought similarly about SQL-Lab, and 89% about the QuizGuide tool. Eighty-four percent of students believed or strongly believed that SQL-KnoT problems challenged them intellectually throughout the course. The adaptive features of the Exploratorium tools have been appreciated by the students as well; for example, 90% of the students agreed or strongly agreed that QuizGuide navigation helped them to choose appropriate questions. Figure 8 visualizes results of the subjective evaluation of the Exploratorium by students.

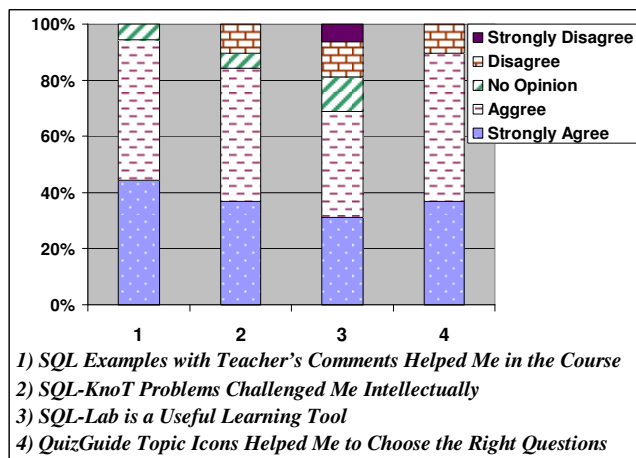


Figure 8. Results of Student Evaluation of the Exploratorium

5. SUMMARY AND FUTURE WORK

We presented an open Exploratorium for database courses. This system provides access to several types of advanced educational activities, each served by an independent Web-based system with a different purpose. The Exploratorium integrates the diverse interactive content, while allowing a teacher to structure the access to this content according to the needs of a specific course. The system also tracks a user's work with all of its components and builds a model of student knowledge, which is used by some of the components to adapt to an individual student. The Exploratorium was evaluated in a graduate and an undergraduate database class. The evaluation demonstrated that the students extensively used the system and its components. The students' feedback about the system and its most important features was highly positive.

Our plans include another round of system evaluation in an undergraduate database course. In addition, we want to extend the current work in two directions. First, in the spirit of the open nature of the Exploratorium, we are working with several partners to add other kinds of interactive content such as SQL Tutor problems [4] and SIETTE quizzes [11]. Second, we are attempting to capitalize on the presence of a student model in the system and provide more kinds of adaptive learning activities and

integration services. We welcome potential collaborators and invite users of the system to contact us.

6. REFERENCES

- [1] Pahl, C., Barrett, R., and Kenny, C. Supporting active database learning and training through interactive multimedia. In: Proc. of 9th annual SIGCSE conference on Innovation and technology in computer science education, ITiCSE '04, ACM, 27-31.
- [2] Guimaraes, M. The Kennesaw Database Courseware (KDC): strong points, weak points, and experience using it in a classroom environment. Journal of Computing Sciences in Colleges, 21, 3 (2006), 91-96.
- [3] Kenny, C. and Pahl, C. Automated tutoring for a database skills training environments. In: Proc. of 36th SIGCSE Technical Symposium on Computer Science Education, (St. Louis, MO, February 23-27, 2005), 58-62.
- [4] Mitrovic, A. An Intelligent SQL Tutor on the Web. International Journal of Artificial Intelligence in Education, 13, 2-4 (2003), 173-197.
- [5] Brusilovsky, P., Yudelson, M., and Sosnovsky, S. An adaptive E-learning service for accessing Interactive examples. In: Nall, J. and Robson, R. (eds.) Proc. of World Conference on E-Learning, E-Learn 2004, (Washington, DC, November 1-5, 2004), AACE, 2556-2561.
- [6] Dieberger, A., Dourish, P., Höök, K., Resnick, P., and Wexelblat, A. Social navigation: Techniques for building more usable systems interactions, 7, 6 (2000), 36-45.
- [7] Brusilovsky, P. and Sosnovsky, S. Engaging students to work with self-assessment questions: A study of two approaches. In: Proc. of 10th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE'2005, (Monte de Caparica, Portugal, June 27-29, 2005), ACM Press, 251-255, available online at <http://www2.sis.pitt.edu/~peterb/papers/ITiCSE05.pdf>.
- [8] Denaux, R., Dimitrova, V., and Aroyo, L. Integrating Open User Modeling and Learning Content Management for the Semantic Web. In: Ardissono, L., Brna, P. and Mitrovic, A. (eds.) Proc. of 10th International User Modeling Conference, (Edinburgh, Scotland, UK, July 24-29, 2005), Springer Verlag, 9-18.
- [9] Brusilovsky, P. KnowledgeTree: A distributed architecture for adaptive e-learning. In: Proc. of 13th International World Wide Web Conference, WWW 2004 (Alternate track papers and posters), (New York, NY, 17-22 May, 2004), ACM Press, 104-113.
- [10] Yudelson, M., Brusilovsky, P., and Zadorozhny, V. A User Modeling Server for Contemporary Adaptive Hypermedia: An Evaluation of Push Approach to Evidence Propagation. In: Conati, C., McCoy, K. and Paliouras, G. (eds.) Proc. of 11th International Conference on User Modeling, UM 2007, (Corfu, Greece, 25-29 June, 2007), Springer Verlag, 27-36.
- [11] Conejo, R., Guzman, E., and Millán, E. SIETTE: A Web-based tool for adaptive teaching. International Journal of Artificial Intelligence in Education, 14, 1 (2004), 29-61.