

User Modeling in a Distributed E-Learning Architecture¹

Peter Brusilovsky, Sergey Sosnovsky, Olena Shcherbinina
School of Information Sciences
University of Pittsburgh, Pittsburgh PA 15260, USA
peterb@pitt.edu, sas15@pitt.edu, ols1@pitt.edu

Abstract: This paper is focused on user modeling and adaptation in distributed E-Learning systems. We describe here CUMULATE, a generic student modeling server developed for a distributed E-Learning architecture, KnowledgeTree. We also introduce a specific, topic-based knowledge modeling approach which has been implemented as an inference agent in CUMULATE and used in QuizGuide, an adaptive system that helps students select the most relevant self-assessment quizzes. We also discuss our attempts to evaluate this multi-level student modeling.

1 Introduction

A number of researchers working on adaptive E-Learning technologies argue that the way to E-Learning classroom for these technologies goes through a distributed, component-based architecture for adaptive E-Learning [2; 5; 6]. The problem is to develop an architecture that will allow independent teams to develop user-adaptive components that could interact in parallel with the same user while integrating collected information, resulting in better adaptations for the user. At the moment, there are two main competing approaches to user modeling in a distributed, component-based architecture: centralized [1; 7] and decentralized [9] user modeling. Decentralized (or distributed) user modeling had its roots in agent-based architectures, while centralized user modeling had its start in user modeling shells and is currently represented by user modeling servers such as CUMULATE and Personis [8]. In our past projects we have explored both centralized [1] and distributed [3] modeling and we think that the former approach currently provides a more reliable and practical solution. We continue to explore the centralized approach in the context of KnowledgeTree, a distributed architecture for adaptive E-Learning based on reusable intelligent learning activities [2]. This paper focuses on student modeling in the context of the KnowledgeTree architecture. We introduce the student modeling server CUMULATE, which was developed to support centralized student modeling. We provide an example of its support of user modeling in QuizPACK - an adaptive hypermedia

¹ This material is based upon work supported by the National Science Foundation under Grant No. 0310576.

service for delivering self-assessment quizzes. We also discuss the issue of the evaluation of CUMULATE-like servers and present our attempt to evaluate the performance of CUMULATE in the context of the QuizPACK service.

2 CUMULATE, a User Modeling Server

The KnowledgeTree architecture is based on a centralized approach. This architecture assumes the presence of four kinds of components in the distributed E-Learning system: student portals, activity servers, value-added services and student modeling servers. A *learning portal* supports the course-authoring interface and maintains a runtime interface for the student. The content of the course is formed by interactive reusable learning activities that are delivered by *activity servers*. A *value-added service* is able to "pass through" the "raw" content adding some valuable functionality to it - such as adaptive sequencing, annotation, visualization, or content integration. All components that interact directly with a student are expected to send information about each important student action to the student modeling server. The server processes this information into a student model and provides student information by request to any interactive component that wants to adapt to the student. The architecture allows the presence of multiple servers of the same kind - including portals and student modeling servers, however, it is assumed that each user works with one portal and one student-modeling server in the context of each course. Any server that complies with the set of inter-component communication protocols could be immediately integrated into the architecture. A student-modeling server should comply with two protocols - accepting student events and replying to requests about the student sent by other components. The internal organization of the server, including the organization of the user modeling is not dictated. Space is provided for competition between different server organizations and user modeling approaches.

The CUMULATE server (Centralized User Modeling for User and Learner-AdapTive Environments) was developed as a generic student-modeling server for the KnowledgeTree project. Following earlier approaches [1; 7] CUMULATE represents information about a student on two levels (Fig. 1): the event storage and an inferred User Model. All student actions that each interactive component is able to trace are sent to CUMULATE, using a standard http-based event-reporting protocol. The structure of this protocol allows an interactive component to report the kind of event (i.e., a specific learning activity or a specific step within more complex activities), the progress (for example, success or failure) and any additional component-specific information). CUMULATE adds a timestamp to each reported event and stores it permanently in the event storage. This is different from traditional student modeling approach in which these events are immediately processed and discarded. The event storage is open to a variety of inference agents that process this data in different ways and convert it into a more familiar form of name-value pairs that altogether form the inferred UM. Various agents can attempt to infer a multitude of user parameters using different methods. For example,

some agents could be focused on inferring the student’s knowledge from different aspects, others could store the student’s interests, even others could monitor the student’s level of motivation. Different agents could attempt to infer the same parameters from the same event storage using different methods. The architecture anticipates the use of internal (i.e., in-server) and external inference agents. The latter kind requests information from event storage and updates inferred UM using dedicated protocols. Moreover, an application could access event storage directly and process the events in a specific way.

The current architecture could be considered unnecessarily general and flexible, but this was a design decision. We need a student modeling architecture which would allow us to explore a range of different student modeling approaches which cover the same event storage. At the moment, we have implemented several internal inference agents that are used by different activity servers and services within KnowledgeTree. To provide a complete example of CUMULATE user modeling and its use in KnowledgeTree, we will focus the rest of the paper on the specific inference agent that performs the topic-based modeling of student knowledge.

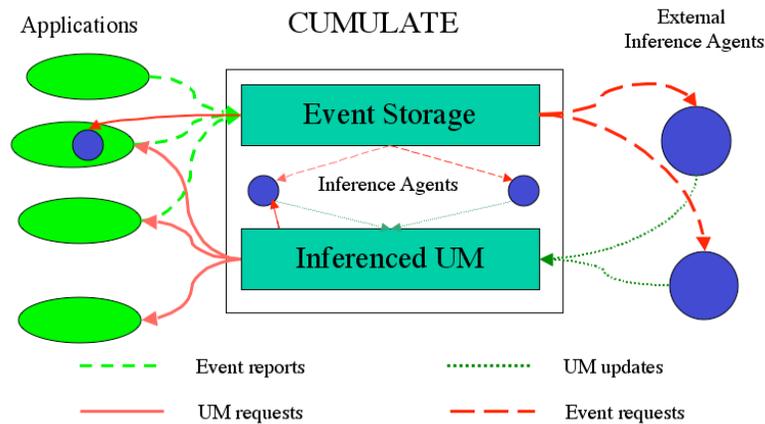


Fig. 1. The structure of the CUMULATE server

3 Topic-Based Knowledge Modeling and its Implementation

Topic-based knowledge modeling is a simplification of the *concept-based knowledge modeling* used in a number of adaptive systems. The state of student knowledge is represented as a weighted overlay covering a set of coarse-grain elements called topics. Each educational activity can contribute to only one topic (whereas in the concept-based approach, it can contribute to multiple concepts). A typical course-level domain model includes just 40-60 topics (in contrast to the several hundred concepts used in concept-based modeling).

Due to its simplicity, topic-based modeling was the first approach implemented within CUMULATE. Our implementation of topic-based modeling is transparent for course authoring: under each topic, the author identifies several educational activities. Student progress within these activities defines the user's understanding of a topic. CUMULATE provides a form-based authoring interface for topics and activities and a *topic-based inference agent*. The interface supports definitions of topics and their relationships with activities. For each activity associated with a topic (and, if necessary, for each substep within an activity step) the author can specify its contribution to the topic. The agent uses the authored relationships between topics and activities to transform events that were reported during the student's work with the topic-related activities into a student knowledge level for that topic (0 to 1). The agent uses a relatively simple method of knowledge calculation. For all activities belonging to that topic, it calculates the weighted average score of reported results.

The first system to explore topic-based knowledge modeling was QuizGuide - a value-added service that provides personalized access to self-assessment quizzes for the C programming language. The quizzes are generated by an activity server called QuizPACK [4]. The "added value" provided by QuizGuide is adaptive annotations. QuizGuide groups all QuizPACK quizzes into topics and adaptively annotates a link to each topic with a target-arrow icon. This icon helps the student choose the topic to work on by indicating which topics are most important and which need additional work. QuizGuide employs topic-based modeling to visualize the current level of knowledge for each topic as the number of arrows in the topic's target, ranging from little or no knowledge (no arrows) to very good knowledge (three arrows).

Since QuizPACK was an original component of KnowledgeTree (i.e., the student's answer to every question of every quiz were already traced in CUMULATE's event storage), and since the topic-based inference agent was already developed, the implementation of QuizGuide was quite straightforward. QuizPACK quizzes were grouped into 22 topics; these topics were defined with the new topic-based authoring interface, forming unique *C-Programming topic scope*; and all quizzes belonging to each topic were added as contributing activities for this topic, with the percentage of contribution depending upon quiz difficulty (see [4] for details). The user interface of QuizGuide was then implemented as a CGI application that requests the current student knowledge level of all topics (inferred by the topic-based agent) through the CUMULATE query interface and then generates navigation frame with adaptive annotations.

5 The Evaluation of QuizGuide

We have evaluated QuizGuide using the traditional "with or without" approach. Some evaluation details are provided in [4]. Despite the relatively simple user modeling and adaptation techniques used in QuizGuide, the system has achieved a remarkable impact on student learning and performance. Guided by adaptive annotations, the students explored more questions, worked with questions more persistently, and accessed a larger variety of

questions. The increase in participation resulted in the increase of their knowledge at the end of the course. However, what can really be evaluated in a "with or without" study of a system driven by a universal user modeling server, which has an inference engine based on authored rules? Are we evaluating the server itself, the topic-based student modeling approach implemented by one of its inference agents, or just the quality of the job done by the author in defining topics and connecting them with activities? The user modeling literature provides no guidance on how to evaluate user modeling servers or approaches. In our opinion, the "proof" of the success of our architecture is its very ability to implement a new student modeling approach and to author the student modeling part of a new adaptive application. The fact that the present application has been successful is, however, an additional argument in favor of topic-based student modeling.

References

1. Brusilovsky P (1994) Student model centered architecture for intelligent learning environment. In Fourth International Conference on User Modeling, Hyannis, MA, 15-19 August 1994, pp. 31-36, also available at <http://www2.sis.pitt.edu/~peterb/papers/UM94.html>
2. Brusilovsky P (2004) KnowledgeTree: A distributed architecture for adaptive e-learning. In The Thirteenth International World Wide Web Conference, WWW 2004 (Alternate track papers and posters), New York, NY, 17-22 May, 2004, pp. 104-113
3. Brusilovsky P, Ritter S, and Schwarz E (1997) Distributed intelligent tutoring on the Web. In du Boulay B and Mizoguchi R (eds) In AI-ED'97, 8th World Conference on Artificial Intelligence in Education, 18-22 August 1997. IOS, Amsterdam, pp. 482-489
4. Brusilovsky P, Sosnovsky S, and Shcherbinina O (2004) QuizGuide: Increasing the Educational Value of Individualized Self-Assessment Quizzes with Adaptive Navigation Support. In World Conference on E-Learning, E-Learn 2004, Washington, DC, USA, November 1-5, 2004, pp. 1806-1813
5. Carmona C and Conejo R (2004) A learner model in a distributed environment. In De Bra P and Nejdl W (eds) Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004), Eindhoven, the Netherlands, August 23-26, 2004. Lecture Notes in Computer Science 3137, Springer-Verlag, Berlin, pp. 353-359
6. Conlan O, Wade V, Gargan M, Hockemeyer C, and Albert D (2002) An architecture for integrating adaptive hypermedia services with open learning environments. In World Conference on Educational Multimedia, Hypermedia and Telecommunications, EDMEDIA'2002. Denver, CO, June 24-29, 2002, pp. 344-350
7. Kay J (1995) The UM toolkit for cooperative user models. *User Modeling and User-Adapted Interaction* 4: 149-196
8. Kay J, Kummerfeld B, and Lauder P (2002) Personis: A server for user modeling. In De Bra P, Brusilovsky P and Conejo R (eds) Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002), Málaga, Spain, May 29-31, 2002, pp. 201-212
9. Vassileva J, McCalla G, and Greer J (2003) Multi-Agent Multi-User Modeling. *User Modeling and User-Adapted Interaction* 13: 179-210.