Cooperative Shared Learning Objects in an Intelligent Web-Based Tutoring System Environment

Sylvia Encheva[1] and Sharil Tumin[2]

[1] Stord/Haugesund University College, Bjørnsonsg. 45, 5528 Haugesund, Norway
sbe@hsh.no
[2] University of Bergen, IT-Dept., P. O. Box 7800, 5020 Bergen, Norway
edpst@it.uib.no

Abstract. This paper focuses on an intelligent Web-based tutorial system composed of reusable learning objects (LOs). Reusable LOs lay down the foundation for a large scale collaboration among educational organizations. LOs developed at different organizations can be used by other organizations thus providing high quality courses to all students. Expert instructors at each organization maintain a repository of LOs. For organizations owning LOs an audit trail for each LO can be easily gathered for billing purposes.

1 Introduction

An intelligent tutoring system (ITS) is an educational software system containing machine intelligence components. An ITS adapts instruction to a student's individual learning needs by collecting information on that particular student's performance. Such a system makes inferences about strengths and weaknesses, and suggests additional work. One of the most common solutions for student diagnosis is testing [6].

A critical aspect in the development of an ITS is how related knowledge is represented and how reasoning for problem solving is accomplished [7]. Some single knowledge representation schemes are used in [5] and [16], and hybrid representations in [8] and [13]. Learners tend to learn better and more deeply if they are motivated by an internal interest and desire to master the material, as opposed to extrinsic rewards and punishments such as grades [9].

In this paper we discuss an intelligent Web-based tutorial system (IWBTS) composed of reusable LOs. Reusable LOs lay down the foundation for a large scale collaboration among educational organizations. LOs developed at different organizations can be used by other organizations, thus providing high-quality courses to every student. Expert instructors at each organization maintain a repository of LOs. Such a flexible framework is not trivial. It is in fact difficult to design and implement. However, by carefully defining LOs and relations among them, together with careful analysis of integration among supporting subsystems, the framework can be realized.

Y. Luo (Ed.): CDVE 2005, LNCS 3675, pp. 227–234, 2005.

IWBTS provides a blueprint for solving each particular problem while introducing a new term or concept. The support for subsequent problems is limited by giving Socratic hints. The number of hints required for the student to solve each problem serves as a measure of learning efficiency. IWBTS also provides different levels of presentation with respect to problem solving. Each level has tutorial material generated for it, since it is important to target tutorial tasks at the student's ability. The tutorial model in IWBTS involves Socratic dialogs that foster critical thinking on course content through a series of questions and answers that enable students to explore and develop understanding. Corrective and elaborative aspects are used for tutoring in IWBTS.

The assessment part of IWBTS contains a polytomous Item Response Theory (IRT) based model [6] for handling data from several response options following each question in a test. IRT models for polytomous items not restricted to two response alternatives are applied by IWBTS to both multiple choice items that possess several response alternatives and to free-response items.

2 Related Work

A survey of current approaches in the area of technologies for electronic documents that are used for finding, reusing and adapting documents for teaching or learning purposes is presented in [17].

LOs are defined as any digital resource that can be reused to support learning [18] and any digital, reproducible, and addressable resource used to perform learning activities or learning support activities, made available for others to use [10]. These definitions do not discuss relationships among LOs in a reuse environment. Our model is based on independent LOs, where a LO is an atomic entity, i.e. one that cannot be decomposed to lower LOs.

A personalized intelligent computer-assisted training system is presented in [14]. A system mapping classroom lectures into Web-based educational lessons is described in [4]. A Category-based Self-improving Planning Module for a tutor agent that utilizes the knowledge learned from automatically-derived student categories to support efficient on-line self-improvement is presented in [11].

ITSs use a tutoring model to guide students' learning with Socratic hints. A Socratic hint is a cognitive and motivational strategy aimed at evoking the student's curiosity and requiring further thought from him [12]. The corrective behavior [15] of the software rejects a wrong value and conveys that it was incorrect. If there is a second consecutive mistake, then the elaborative behavior [15] comes into action, suggesting an appropriate relationship by which the value can be derived. The suggestion depends on what the student has done so far.

IRT, also called latent trait theory, is the study of test and item scores based on assumptions concerning the mathematical relationship between abilities (or other hypothesized traits) and item responses [2]. Polytomous IRT based models are capable of handling data from several response options while dichotomous IRT based models consider only two possible scored responses such as true/false, correct/incorrect, endorsed/not endorsed, etc. [1].

A framework for sharing protected Web resources among independent organizations is presented in [3]. Very important components of the model are a collaborative, distributive management of user membership in a group in one organization, resource management by the service provider, and security enhancement.

3 System Framework

IWBTS is based on a flexible framework that provides the instructors with possibilities in designing courses using shared, reusable LOs and provides the students with an ITS environment.

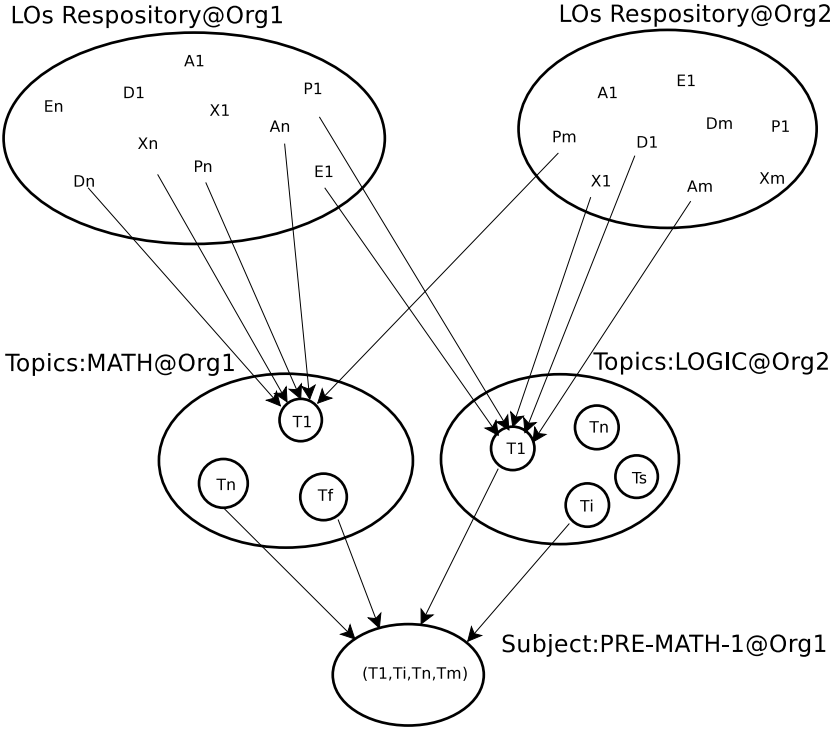

Fig. 1. Learning objects

Course design costs a significant part of a WEB-based learning project budget. A big cost reduction is possible if as many LOs as possible are reusable and are shared among collaborative organizations. Instructors can just define which LOs should be included in their courses. Using drills and students responses to questionnaires in a WEB-based course can provide students with an ITS environment. This will make the course building more flexible and reduce costs for both students and organizations.

A subject $\mathcal{S}_i = (\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n)$ is defined as a strictly sequential list of topics $\mathcal{T}_j$ by a course builder (Fig. 1). Each $\mathcal{S}_i$ definition is saved in the database in XML. The subject's definition contains all the metadata needed to uniquely describe the subject in question. These metadata include server, domain, title, topics list, and other operational parameters.

A topic $\mathcal{T}_j = \{\mathcal{P}_\alpha, \mathcal{E}_\beta, \mathcal{X}_\gamma, \mathcal{D}_\delta, \mathcal{A}_\epsilon\}$ is composed of theoretical parts $\mathcal{P}_\alpha$, exercises $\mathcal{E}_\beta$, examples $\mathcal{X}_\gamma$, drills $\mathcal{D}_\delta$, and assessments $\mathcal{A}_\epsilon$, where some of them can be empty. Any $\mathcal{P}_\alpha, \mathcal{E}_\beta, \mathcal{X}_\gamma, \mathcal{D}_\delta, \mathcal{A}_\epsilon$ is a LO designed by expert instructors. These LOs are WEB documents (HTML, PDF, Flash, Java-applet, WEB-form, etc.).

Each $\mathcal{T}_j$ is defined by the course builder in XML and saved in the database. The topic's definition contains all the metadata needed to describe the topic content, drill, and assessment policies. These documents can be used to provide learners with dynamic HTML pages for each invocation of a topic depending on its LO's set.

Topics are defined dynamically by diagnostic components of the system. The drills' design contains inference rules analyzing students' answers to carefully prepared tests. This provides students with an environment supporting an ITS. Students' test scores and wrong responses are used as input parameters to a diagnostic component in the $\mathcal{D}_\delta$, which provides recommendations to alternative personal topic trails containing relevant LOs in order to solve the current topic's drill.

Assessments $\mathcal{A}_\epsilon$ are used for both formative and summative evaluations. Exercises $\mathcal{E}_\beta$ give a list of unsolved problems for students to practice on, while examples $\mathcal{X}_\gamma$ can be a list of solved problems, Flash, or Java-applets for students to work with. Since each LO is self-contained, they can be developed independently at different organizations. Expert instructors at each organization maintain a repository of LOs containing $\mathcal{P}_\alpha, \mathcal{E}_\beta, \mathcal{X}_\gamma, \mathcal{D}_\delta$, and $\mathcal{A}_\epsilon$. For students at different collaborative organizations, they will have the advantage that all LOs are accessible to them. For organizations owning the LOs, an audit trail for each LO can easily be gathered for billing purposes.

3.1 System Model

RPC based agents provide system integration and specialized services (Fig. 2). They communicate with each other by a request-respond mechanism in which remote procedure calls are done among different agents providing users with a dynamic and personalized learning environment.

XML subject and topic definition documents retrieved from the database are converted into an internal datastructure by an agent before the final HTML documents are presented to students. By using the user's response data and the user's current status, the system invokes appropriate agents to provide each learner with a corresponding HTML page.

Agents providing the system with specialized functions are developed and maintained separately. Different communication protocols such as SOAP and XML-RPC can be used. Agents can be implemented and can run on a multi-host and multi-OS depending on implementation architecture.

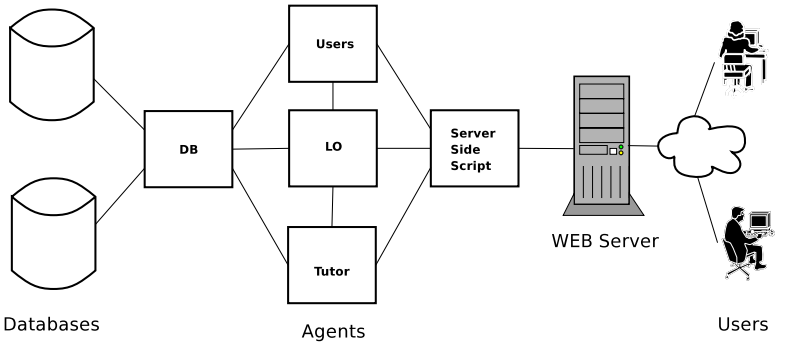

Fig. 2. Agents

3.2 System Architecture

The system uses an Apache WEB server and a PostgreSQL relational DBMS that provide the back-end database. The Python script programming language is used for all the agents. These three system software components are high quality, well proven, reliable, and free open source software.

XML-RPC implemented in Python provides the necessary tools for implementing multi-platform agents. Python encourages modular software engineering methodology. Many core modules are freely available and continually maintained by Python's Web communities.

Functions provided by Python's xmlrpclib are used to export agents' data structures to XML and to import XML back to agent's data structures. These agents' data states are then saved in the database in XML and are used as extended memory for running agents in the system for a particular user at a specific time. Such saved data states are subject and topic definitions, user's profile and status, user's test scores, and topic trails.

Server-side scripts (SS) and these agents are core components in IWBTS providing the following functionalities: create dynamic HTML pages using template files, thus producing a consistent page style and a structured navigational menu for the users; process users' GET/POST variables from Web forms and manage Web cookies; call appropriate agents in responce to users' interaction with the system.

A User agent contains user and instructor modules. A user module contains, among other things, user registration, user administration, user authentication, and authorization. To use the system a user needs to register. A user needs to fulfill certain criteria to be able to subscribe to a subject. Users are authenticated by consulting a users' domain Directory server via Lightweight Directory Access Protocol (LDAP) on logon. Users are authorized to access a subject if subscribed. Instructor modules provide an interface for the instructors for defining topics and subjects, students' status reports, and messaging.

An LO agent is responsible for dispatching LOs, creating dynamic menu structure, and providing a user's current profile and status. It is a core sys-

tem component for providing a personalized learning environment for the users. It keeps track of users' navigational activities, guiding users through the course in responce to their previous actions, responses, and drill scores. SS calls the LO agent, and it in turn calls a DB agent. User profile and/or status data provided by the DB agent gives enough information for the LO agent to render dynamic HTML for the user via SS. This agent also collects users' audit trails.

A DB agent provides database connectivity for the other agents. This agent receives Structured Query Language (SQL) statements from other agents as call parameters. If the SQL statements are 'select' statements then the DB agent will return the calls with data arrays read from the database. This agent provides database service abstraction. Other agents need not consider the lower database client functions to do SQL queries. The DB agent acts as a proxy database client on behalf of others agents.

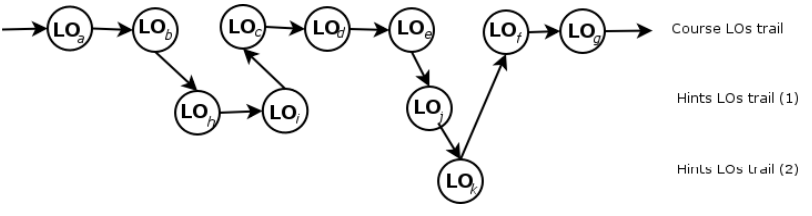

Fig. 3. Trails

A tutor agent responds to users' answers from tests and drills. A pedagogically crafted scheme with a set of questions and answers provides the agent with a programmed intelligence, in which wrong answers to a drill lead learners to appropriate hints and advice. The students can then subscribe to those LOs suggested by the hints or try the drill again and continue with the current course. Students can jump back to the current course trail at any time while following trails suggested by the tutor agent. The agent calculates scores, shows result status, and keeps track of assessments taken by each student. After each assessment the agent gives recommendations on how to proceed with the course. The tutor agent's instructor modules provide the instructors with an interface for students' progress reports and diagnostics.

4 Conclusion

A Web-based ITS should be able to respond to students' needs as they progress through the system. IWBTS provides a structured learning system environment, suitable for producing automatic links through simple logic-based reasoning mechanisms and guiding students in their navigation. Such an approach helps to overcome a major concern about a cognitive overload in Web enhanced courses. A mutually respectful environment and probing questions are essential elements of effective tutorial dialogs provided by IWBTS.

An instructor is presented with a complete report about every student's activities by IWBTS. The multi-agent architecture tracks student's responses, performs a qualitative reasoning on them, and delivers immediate diagnostics and advice. A qualitative reasoning engine in IWBTS evaluates each student separately, compares students' results, and then creates their learning profiles.

IWBTS is composed of reusable LOs. A large scale collaboration among educational organizations can provide high quality courses to their students by reusing LOs. For organizations owning LOs, an audit trail for each LO can easily be easily for billing purposes.

References

1. Dodd, B.G., DeAyala, R.J., Koch, W.R.: Computerized adaptive testing with polytomous items. Applied Psychological Measurement **19**(1) (1995) 5–22
2. Embretson, S., Reise, S., Reise, S.P.: Item Response Theory for Psychologists. Lawrence Earlboum associates, Inc., NJ (2000)
3. Encheva, S., Tumin, S.: Collaborative Role Management for Sharing Protected Web Resources. Lecture Notes in Computer Science, Vol. 3190. Springer-Verlag, Berlin Heidelberg New Jork (2004) 221–229
4. Friedland G., Knipping L., Rojas R. and Tapia E.: Web Based Education as a Result of AI Supported Classroom Teaching. Lecture Notes in Artificial Intelligence, Vol. 3190. Springer-Verlag, Berlin Heidelberg New Jork (2003) 290–297
5. Guin-Duclosson, N., Jean-Danbias, S., Norgy, S.: The AMBRE ILR: How to use Case-Based Reasoning to teach methods. In Cerri, S.A., Gouarderes, G., Paraguacu, F. (eds.). Lecture Notes in Computer Science, Vol. 2363. Springer-Verlag, Berlin Heidelberg New Jork (2002) 782-791
6. Guzmàn, E., Conejo, R.: A model for student knowledge diagnosis through adaptive testing. Lecture Notes in Computer Science, Vol. 3220. Springer-Verlag, Berlin Heidelberg New Jork (2004) 12–21
7. Hatzilygeroudis, I., Prentzas, J.: Knowledge representation requirements for intelligent tutoring system. Lecture Notes in Computer Science, Vol. 3220. Springer-Verlag, Berlin Heidelberg New Jork (2004) 87–97
8. Hatzilygeroudis, I., Prentzas, J.: Using a hybrid rule-based approach in developing an intelligent tutoring system with knowledge acquisition and update capabilities. Journal of Expert Systems with Applications **26** (2004) 477–492
9. Johnson, L.W., Rizzo, P.; Politeness in tutoring dialogs: "Run the factory, that's what I do". Lecture Notes in Computer Science, Vol. 3220. Springer-Verlag, Berlin Heidelberg New Jork (2004) 67–76
10. Koper, E.J.R.: Combining re-usable learning resources and services to pedagogical purposeful units of learning. In A. Littlejohn (Ed.), Reusing Online Resources: A Sustainable Approach to eLearning London Kogan Page (2003) 46–59
11. Legaspi, R., Sison, R. and Numao, M.: A Category-Based Self-Improving Module. Lecture Notes in Computer Science, Vol. 3220. Springer-Verlag, Berlin Heidelberg New Jork (2004) 554–563
12. Lepper, M.R., Woolverton, M., Mumme, D., Gurther, G.: Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In S.P. Lajoie, S.J. Derry (Eds.): Computers as cognitive tools. LEA, Hillsdale, NJ (1993) 75–105

13. Magoulas, G.D., Papanikolaou, K.A., Grigoriadou, M.: Neuro-fuzzy synergism for planning the content in a Web-based course. Informatica **25** (2001) 39–48
14. Pecheanu E., Segal C. and Stefanescu D.: Content modeling in Intelligent Instructional Environment. Lecture Notes in Artificial Intelligence, Vol. 2774. Springer-Verlag, Berlin Heidelberg New Jork (2003) 1229–1234
15. Self, J.: Student models: What use are they?. In P. Ercoli., and R. Lewis, Artificial intelligence tools in education. Amsterdam: North Holland. (1988) 73–86
16. Vanlehn, K., Zhendong, N.: Bayesian student modeling, user interfaces and feedback: a sensitivity analysis. International journal of AI in Education **12** (2001) 155–184
17. Vercoustre, A. and McLean, A.: Reusing educational material for teaching and learning: Current approaches and directions. International Journal on E-learning, a special issue on Technologies for Electronic Documents. **4**(1) (2005) 57–68
18. Wiley, D. (Ed.): The Instructional Use of Learning Objects. Agency for Instructional Technology and the Association for Educational Communications and Technology (2002)