The eXtensible Tutor Architecture: A New Foundation for ITS

Goss NUZZO-JONES, Jason A. WALONOSKI, Neil T. HEFFERNAN, Tom LIVAK
Worcester Polytechnic Institute
100 Institute Rd, Worcester, MA 01609
(508) 831-5569
goss@wpi.edu, jwalon@wpi.edu, nth@wpi.edu, tomlivak@alum.wpi.edu

Abstract. The eXtensible Tutor Architecture (XTA) was designed as a platform for creating and deploying many types of Intelligent Tutoring Systems across many different platforms. The XTA presently has support for state graph pseudo-tutors and JESS model-tracing cognitive tutors, in both a client and server context. The XTA was designed with future development in mind, allowing easy specification of new tutor types, tutoring strategies, and interface layers. It has been used as the foundation of the Assistments Project, a wide scale web based ITS deployment. The Assistments Project is on track to provide ITS content to 100,000 students in the state of Massachusetts.

1. Introduction & Background

This research was conducted to develop a scalable, stable framework for deploying Intelligent Tutoring Systems (ITS) of many types to a variety of platforms, but was developed in particular based on the needs of the Assistments Project [3]. This project required that we be able to support a range of tutor types (constraint-based, cognitive-model, etc), provide stability and scalability, and deliver tutoring content to a host of clients – either rich client applications, or thin light-weight HTML clients. Additionally, we aimed to create an environment capable of supporting many tutoring strategies, operate as both a client and scaleable server application, provide logging capabilities for student analysis, and remain highly extensible for future development. To accomplish these goals, we employed component-based software engineering practices as well as judicious use of design patterns such as separation of logic and presentation. The results of this research were used as the deployment mechanism for the Assistments Project, a mathematics ITS project based at Worcester Polytechnic Institute and Carnegie Mellon University [3].

2. The eXtensible Tutor Architecture

The result of our research is a framework that we refer to as the eXtensible Tutor Architecture (XTA). This framework controls the interface and behaviors of our intelligent tutoring system via a collection of modular units. These units conceptually consist of a *curriculum* unit, a *problem* unit, a *strategy* unit, and a *logging* unit. Each conceptual unit has an abstract and extensible design allowing for evolving tutor types and content delivery methods. The current implementation has full functionality in a variety of useful contexts.

The *curriculum* unit represents a collection of educational content scheduled for tutoring. The *curriculum* is composed of one or more *sections*, with each *section* containing *problems* or other *sections*. This recursive structure allows for a rich hierarchy of different types of *sections* and *problems*.

The *section* component is an abstraction for a particular listing of problems. This abstraction has been extended to implement our current *section* types, and allows for future expansion of the *curriculum* unit. Currently existing *section* types include "Linear" (*problems* or sub-*sections* are presented in linear order), "Random" (*problems* or sub-*sections* are presented in a pseudo-random order), and “Experiment” (a single *problem* or sub-*section* is selected pseudo-randomly from a list, the others are ignored). Plans for future *sections* types include a “Directed” *section*, where *problem* selection is directed by the student’s knowledge model [1].

The *problem* unit represents a problem to be tutored, including questions, answers, and relevant knowledge-components required to solve the problem. For instance, a problem could be implemented as a hierarchy of questions connected by correct and incorrect answers, along with hint messages and other feedback. Each of the questions represented by a *problem* composed of two main pieces: an *interface* and a *behavior*.

The *interface* definition is interpreted by the runtime and displayed for viewing and interaction to the user. To handle multiple target GUIs, the XTA deals with collections of low-level widgets with simple behaviors (such as text labels, fields, buttons, etc) that are translated into high-level widgets with complex behaviors (such as spell-checking text fields or algebra parsing text fields). This separation allows for easier conversion of interfaces towards a target UI (HTML or Swing, for example). A *behavior* definition for each *interface* is what allows each *problem* to define the results of actions (such as clicking a button) on the *interface*. Careful design keeps tutoring content developers focused on high-level widgets, while low-level representations can be ignored.

The *strategy* unit allows for high-level control over *problems* and provides flow control between *problems.* The *strategy* unit consists of *tutor strategies* and the *agenda*. Different *tutor strategies* can make a single *problem* behave in different fashions. Some types of *tutor strategies* that have already developed include message strategies (for hints, feedback, and instruction), explain strategies (for problem explanation), and forced scaffolding strategies (forcing a student into a particular branch of a problem). Future *tutor strategies* could include dynamic behavior based on knowledge tracing of the student log data, for example, which would allow for continually evolving content selection without a predetermined sequence of *problems*.

The final conceptual unit of the XTA is the *logging* unit, which receives detailed information from all the other units relating to all user actions and component interactions at every level of the system. Judicious logging can record the data required to replay or rerun a user’s session to explore their misunderstandings of the content, reveal usage-patterns to aid in the detection of system gaming (superficially going through tutoring-content without actually trying to learn) [2], and provide useful reports to educators to enhance classroom instruction.

Each conceptual unit in the XTA is capable of being appropriately networked to leverage the benefits of distributing our framework over a network and across machines, to provide scalability. Also, based on memory footprint testing, thousands of copies of the XTA could run on a single machine. More importantly, the individual units described above are separated by network connections. This allows individual portions of the XTA to be deployed on different computers. Thus, in a server context, additional capacity can be added without software modification, and scalability is assured.

The XTA can also transform with little modification into a classic client-server architecture (such as Java WebStart) or a thin-client server configuration (HTML) depending on particular requirements. Both types of applications allow for pluggable client interfaces due to the *interface* definitions described earlier.

3. Methods and Results

The XTA has been deployed as the foundation of the Assistments Project [3]. This project provides mathematics tutors to Massachusetts students over the web and provides useful reports to teachers based on student performance and learning. The system has been in use for a year, and has had nearly 1000 total users. These users have resulted in over 1.3 million actions for analysis and student reports [2]. To date, we have had a live concurrency of approximately 50 users from Massachusetts schools. However, during load testing, the system was able to serve over 500 simulated clients from a single J2EE / database server combination. The primary server used in this test was a Pentium™ 4 with 1 GB of RAM running Gentoo Linux. Our objective is to support 3,000 users concurrently.

Tutors that have been deployed include scaffolding state diagram pseudo-tutors with a variety of strategies. We have also deployed a small number of JESS cognitive tutors for specialized applications. It should be noted that the tutors used in the scaling test described above were all pseudo-tutors, and it is estimated that a much smaller number of JESS tutors could be supported.

In summary, the launch of the XTA has been successful. The configuration being used in the Assistments project is a central server as described above, where each student uses a thin HTML client and data is logged centrally. Public school staff have enthusiastically reviewed the software. Since September 2004, the software has been in use at least three days a week over the web by a number of schools across central Massachusetts. This deployment is encouraging, as it demonstrates the stability and initial scalability of the XTA, and provides significant room to grow.

4. Conclusions

The larger objective of this research was to build a framework that could support 100,000 students using ITS software across the state of Massachusetts. We're encouraged by our initial results from the Assistments Project, which indicate that the XTA has graduated from conceptual framework into a usable platform (available at http://www.assistments.org). We have many planned improvements to the system including dynamic *curriculum* sections, *tutor strategies* that alter their behavior based on knowledge tracing of the student log data, and additional *interface* display applications for alternative GUI platforms. We believe the reconfigurable and customizable nature of the XTA could make it a valuable tool in the continued evolution of Intelligent Tutoring Systems.

References

[1] Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4 (2), 167-207.

[2] Feng, Mingyu, Heffernan, N.T. (2005). Informing Teachers Live about Student Learning: Reporting in the Assistment System. *Submitted as poster to the 12th Annual Conference on Artificial Intelligence in Education 2005, Amsterdam*

[3] Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K. R., Junker, B., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T.E., Upalekar. R, Walonoski, J.A., Macasek. M.A., Rasmussen, K.P. (2005) The Assistment Project: Blending Assessment and Assisting. *The 12th Annual Conference on Artificial Intelligence in Education 2005, Amsterdam*

This research was made possible by the US Dept of Education, Institute of Education Science, "Effective Mathematics Education Research" program grant #R305K03140, the Office of Naval Research grant # N00014-03-1-0221, NSF CAREER award to Neil Heffernan, and the Spencer Foundation.