Proceedings of the workshop "Intelligent Educational Systems on the World Wide Web",
8th World Conference of the AIED Society, Kobe, Japan, 18-22 August 1997

The Challenge of Constructing a Mega-tutor over the Web

Kurt Rowley
Advanced Training Technologies Branch
U.S. Air Force Research Laboratory (AL/HRCT)
7909 Lindbergh Dr., Brooks AFB, TX 78235-5352, USA
E-mail: RowleyK@cobra.brooks.af.mil

Abstract. The development of generic and reusable ITS components is a current problem studied by ITS researchers. The use of the WWW to collaboratively construct a mega-tutor is proposed as a challenge and future benchmark for ITS researchers.

1. Introduction

To help researchers develop and implement ITS or ILE environments with generic knowledge bases, generic student models, or other reusable components, it would be useful to have a common framework within which the developers could share the modules with other developers. Yum and Crawford have suggested that a WWW-based collaborative development system could be a framework for developing reusable tutoring systems, and expanding the ability of developers to implement ITS environments (Yum & Crawford, 1996). Preparing this type of a framework would require the sharing of code and technical information, and large-scale collaboration on the architecture of the ITS. One possible approach to supporting this type of framework would be to set a benchmark, a challenge or goal that would provide a specific task for developers and researchers. Such a project, the development of a framework for the reuse of components of an ITS, and the cooperation of developers and researchers on accomplishing a specific task, could be rapidly accomplished with the help of the WWW. By working on a large-scale common project like a large-scale distributed tutor, or a '*mega-tutor*', many developers could collaborate and eventually provide a great benefit to the entire educational WWW community.

2. A *Mega-tutor*

The main idea behind the construction of a mega-tutor would be that the web would be used as a delivery mechanism for a large and unlimited tutor, essentially a tutor that spans the web. The mega-tutor would eventually be composed of many thousands of agent-like objects, each of which could act as components of a tutoring or learning environment system, and be called-on by search engines according to instructional, training, or performance needs of a given student, then combined dynamically to meet those needs. The agent-type components of the mega-tutor would each be certified to perform a specific function, and the search engine for constructing the tutor would seek an effective combination of components based on the expected effectiveness of each object in accomplishing the purposes identified by the student. A tutor would then be dynamically constructed by the search engine and it would perform its function until completion, and then return control to the original calling system, or to the user.

3. The Benchmark

One of the major challenges for constructing this type of mega-tutor is to develop sufficiently generalizeable and generic tutoring objects for the requirement that the objects be combined in a dynamic situation. The test for this benchmark would be that of an end-user, or an intelligent search engine, could 'drag-and-drop' multiple modules together to create a new tutor. The modules might include knowledge bases, user interfaces (students, teachers, peers), inference engines, or other instructional resources. Some of these modules might contain components that could adapt to or control other components. The communication channels for the modules would intelligently self-configure according to the abilities of each module. These modules could be combined by instructors, or mentors who could measure the effectiveness of the tutor and trouble-shoot any problems that might arise in the tutor's self-configurations. The modules could also be combined automatically by intelligent search engines. Any combinations of modules would be possible, so tutors could be constructed on an ad-hoc basis for individual students, for small collaborative learning groups, for working teams, or for large groups of students (see Fig. 1).

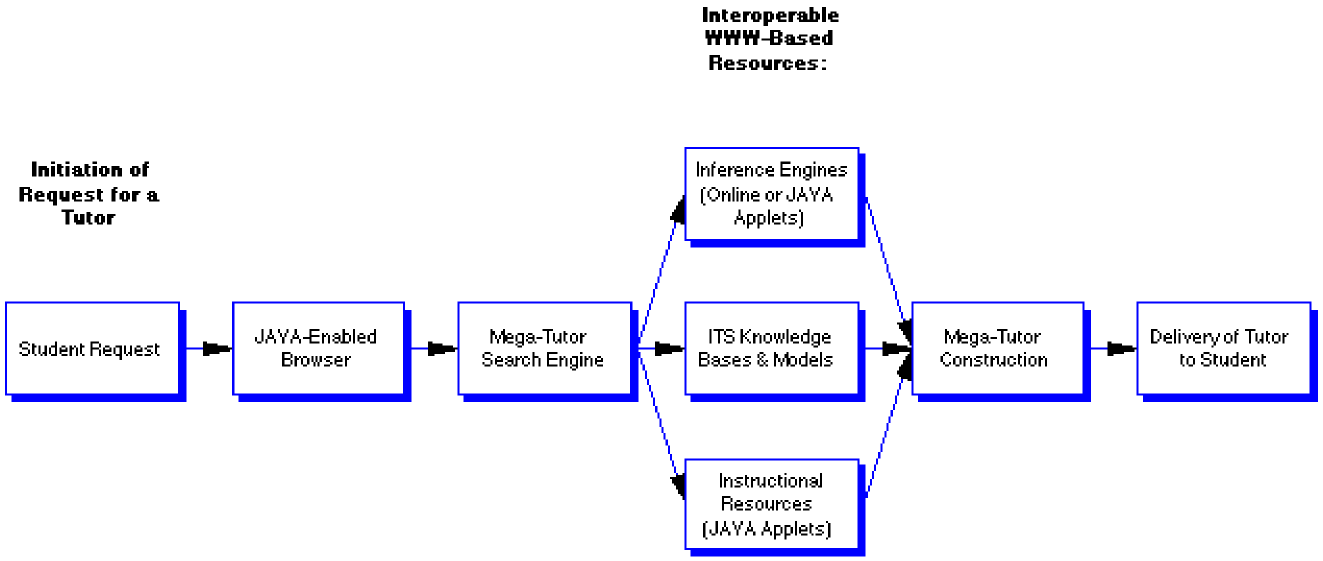
Interoperable WWW-Based Resources:
Initiation of Request for a Tutor
Student Request
JAVA-Enabled Browser
Mega-Tutor Search Engine
Inference Engines (Online or JAVA Applets)
ITS Knowledge Bases & Models
Instructional Resources (JAVA Applets)
Mega-Tutor Construction
Delivery of Tutor to Student

Figure 1: Mega-tutor Functional Model.

4. Application

A good example of this capability over the Web would be as follows: imagine that you are an accountant working on a spreadsheet and you have a complicated question about how property depreciation works. You are connected to the web, and using a *mega-tutor search engine* you construct a tutor to teach yourself about depreciation, including the financial formulas and applicable tax law. The tutor runs until you understand the concept of depreciation and are ready to return to your spreadsheet. Sounds great, but how would it work? A sophisticated WWW search engine would help narrow down the question, link to interoperable ITS engines, knowledge bases, or other instructional agents, and construct an intelligent tutor to help assist you. This type of mega-tutor system would exist across the web, and its design would be based on a merger of concepts, drawing from AI and ITS research, interoperability initiatives, embedded systems research, instructional science, and information retrieval research.

The use of the tutoring components would be largely invisible to the end-users. The users would only know that they were using a sophisticated search engine that was able to construct effective training to meet their individual needs. Of course the amount of content necessary for this type of mega-tutor is significant, but if the interoperability initiatives currently underway for ITS components are successful, the content could be developed across the world by thousands, or even millions of instructional developers, all of whom used knowledge engineering tools that adhered to interoperable data standards. By beginning work with a few simple domains, and pursuing the 'drag-and-drop' benchmark, a collaborative development system should be capable of this task. Perhaps the construction of a mega-tutor is an idealistic challenge, but it could provide a benchmark test environment for the use of interoperable ITS components over the Web, and it provides a vision of the future of ITS systems.

Reference

Yum, Kwok-Keung & Crawford, John R. (1996). On the feasibility of an interoperable tutorial machine to support the development and delivery of teaching. *ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs.*