

# **Masters Thesis Proposal**

## **Towards Teachers Quickly Creating Tutoring Systems**

Michael A. Macasek

Thesis Advisor:  
Professor Neil Heffernan

Thesis Reader:

---

Department of Computer Science  
Worcester Polytechnic Institute  
100 Institute Road  
Worcester, MA 01609

Saturday, April 16, 2005

### **Abstract**

Intelligent Tutoring Systems have historically been shown [5, 6] to be an effective means of educating an audience. While there is great benefit from such systems they are generally very costly to build and maintain. Currently it is estimated that 200 hours of time is required to produce one hour of Intelligent Tutoring Systems content [7, 8]. The Office of Naval Research is funding this work because they want to reduce the cost of construction for Intelligent Tutoring Systems. In order for Intelligent Tutoring Systems to be widely accepted and used in the classroom environment there needs to be a tool set that allows for even the most novice user to maintain and grow the system with minimal cost. The goal of this thesis is to create such a tool set targeted towards the Assistments Project [1]. One of the main goals of the Assistments Project is to provide a means for teachers to receive meaningful data from the system that they can take back to the classroom environment thus enabling a comprehensive learning solution. This tool set will be pivotal in the success of that goal as it is the primary means for teacher to interact with the system and receive such data. The effectiveness of the tool set will be examined by its ability to reduce the overall time it takes to package and distribute content on an Intelligent Tutoring System.

## 1.0 Introduction

The focus of this thesis work is aimed at “closing the loop” in the Assistments Project [1, 2]. The Assistments Project is an Intelligent Tutoring System whose main goal is providing tutoring for middle school level math. Currently the system is used in several Worcester Area middle schools in preparation for the MCAS [3] exams. The system provides a means for teachers to assign these tutors to their students which provide meaningful feedback and directed instructional learning based on the student’s actions. Closing the loop in the Assistments Project implies teaches to operate outside the intervention of system administrators. Teacher will be able to create and assign content as well as manage their classes and students.

For the Assistments Project to be successful the ability to self manage the system is a necessity. There is a need for a simple but powerful interface from which users from differing privilege levels are able to manage the users that fall under their control. Currently four levels of control in the system are needed. The required levels are to be *student* accounts who maintain their account, *teacher* accounts who maintain their accounts as well as administer their student’s accounts, *superintendent* accounts that maintain their account and administer their teacher’s account, and finally there are *system administrator* accounts who maintain all accounts.

## 2.0 Related Work

The key to having a successful Intelligent Tutoring System is for the content creation tools to be extensible. What this means is that a math expert, that is a person who is a math teacher or

in another math profession, should be able to easily create math content to be deployed via the Intelligent Tutoring System with little to no knowledge of what an Intelligent Tutoring System is or does.

While this thesis work will be primarily oriented towards integration with the Assistments Project many other online tutoring systems exist; many of which provide the same rich features as the Assistments Project. The Online Learning Initiative (OLI) [11] from Carnegie Mellon University provides an excellent collection of online tutors directed at many subject areas. While the OLI provides a wide range of online tutors, the tutors are lacking in extensibility resulting in a high cost for creating content. Cognitive Tutor Authoring Tools (CTAT) [12] from LearnLab also provides excellent online tutors in addition to being extensible. However, CTAT lacks the administrative tools being investigated by this thesis. The National Center for Research on Evaluation, Standards and Student Testing (CRESST) [10] not only provides an excellent online system but also has an extensive collection of tools to support the creation and distribution of content. However the CRESST system does not offer tutoring, instead it allows for open ended questions that are then evaluated by a human teacher. SlideTutor [9] is a domain specific web-savvy tutor system directed at the medical community. This system works well for its domain but is not very extensible beyond it

Within the Assistments Project there exists a small subset of the required features for the User Console; the tool that is to be created under this research.

Currently existing functionality includes the ability for all accounts to reset their passwords, teachers have the ability to create new classes for students to enroll, and teachers can reset student passwords.

Two additional tools that fall under the domain of the User Console are the Curriculum Creator and the Curriculum Assigner. These tools allow for the creation of a curriculum from an existing set of assistments and the subsequent assigning of the curriculums to teachers classes. The tools were designed to be functional and only work for a specific subset of assistments and fall short in ease of use.

The Assistment Builder [4] is an existing full-featured assistment creation tool. While this tool does not fall directly under the domain of the User Console it is a tool that primarily teachers will have access to and will be integrated into the User Console to provide a more intuitive group of related tools.

The created curriculums and assistments are administered to students via the eXtensible Tutor Architecture which is the underlying framework of The Assistments Project. [2]

### **3.0 Proposal**

Intelligent Tutoring Systems (ITS) have been proven to successfully aid student learning [5]. The current drawback of an ITS is that the creation and deployment of content requires extensive experience within the ITS discipline. Tools are needed to remove this steep learning curve allowing for an ITS to become common place in traditional school settings. Once teacher can quickly and effectively distribute content to their

students they can then utilize the feedback in the classroom setting to focus on under performing areas. Allowing teachers to take information from the ITS into the classroom will be the key for the future adoption of Intelligent Tutoring Systems and the creation a comprehensive learning solution. Such a solution hinges greatly on the successful development of tools that are usable by individuals from a limited ITS background.

To provide this solution I propose to investigate and create the User Console and determine its success based on usability tests. The main goal is to provide a set of tools that allow for teachers to integrate the use of Intelligent Tutoring Systems into their daily classroom routine as well as use the data gathered by the system to refine the areas of focus within the classroom.

It is worth noting that all tasks, outside the previously mentioned tools, are currently manually completed with regard to the Assistments Project. This process is both time consuming and a bottleneck, which inhibits full use of the system. The ultimate goal is to provide the means to accomplish all tasks without intervention from a project team member.

The User Console will consist of two core components; the User Manager and the Curriculum Manager. The User Manager will allow users to manage their accounts as well as their relationships to other accounts. The Curriculum Manager will be utilized mostly by teachers and is the means by which assistments are disbursed amongst students. There are currently three types of supported curriculums: Linear,

Random, and Experimental. Linear progresses through the included tutors in the order in which they appear in the curriculum. Random allows for the selection of a random a new tutor from the tutors included in the curriculum. Experimental has two sub-curriculums from which one is selected and then executed. The sub-curriculums can be any of the three supported curriculums.

The console essentially consists of three parts, a User Manager, a Curriculum Manager, and the Assistent Builder. As mentioned above the Assistent Builder is an existing tool that will be tied into the console for functional purposes.

The User Manager will function differently depending on the privileges of the users accessing it. It is important to note that users can and do belong to more than one group (i.e. a student can also be a teacher). It is because of this that there will be overlaps in the functionality thus an individual of a higher privilege class will have all the functionality of the groups below them as well as the functionality attributed to their group.

The *student* privilege class will have the most basic functionality. Students will be able to change their password, view their current classes, enroll in new classes, and drop currently enrolled classes. When a student wishes to drop a class they teacher of that class must approve it.

The *teacher* privilege class will have the same functionality as the students in addition to the following functionality. Teachers can view all their classes, rename their classes, add classes, and

remove classes. Teachers will also be able to reset the passwords for the students currently enrolled in their classes. Teachers can create students and enroll them in one of their classes as well as remove students from their classes and approve/reject class drop requests. Teachers can add/remove themselves to/from a superintendent's control. When a teacher wishes to be removed from a superintendents control the superintendent must approve it.

The *superintendent* privilege class will have the same functionality as the student and teacher class in addition to their functionality. Superintendents will have control over their teacher's accounts. They will be able to create teachers and place them under their control, reset existing teachers passwords, remove teachers from their control, and approve/reject teacher drop requests additionally superintendents can create schools and place newly created teachers in the newly created school.

The *system administrator* privilege class will have the same functionality as superintendents, teachers, and students as well as global system control. System administrators will be able to create/delete a user account as well as establish users associations. System administrators can create any group such as states/countries, districts, schools, and classes. System administrators will also be able to re-associate orphaned users (users not associated to any state/country, district, school, or class). Additionally system administrators will be able to reset a user's password.

The Curriculum Manager will only be accessible by the system administrators, superintendents, and teachers. The

Curriculum Manager will provide all functionality related to the distribution of assignments to students. This functionality includes the creation and assigning of curriculums.

The creation of curriculums will function the same for each privilege class. Each user will be able to create a curriculum from three main assignment pools. They have access to the assignments they created, publicly available repositories, and any items created by a user associated to them but below them in the user hierarchy. Upon creation curriculums will initially be marked as private, but access permissions will be able to be granted on school, district, state, and global levels. This will allow for creation of universal curriculums.

The assigning of curriculums has subtle differences depending on a user's privilege class and the functionality is stackable much in the same way as the User Manager. Users will be able to assign any curriculum they have made and/or any curriculum publicly available to them. Teachers will be able to assign curriculums to all their classes, to a particular class or subset of classes, and to a particular student or subset of students that are enrolled in their class(es). Teachers will be able to view currently assigned curriculums as well as whom they are assigned to and they can remove an assigned curriculum from a class(es) or user(s). Superintendents will be able to assign curriculums to the teachers under their control the result of which will be that all students associated to the teachers will be assigned the curriculums. System administrators will be able to assign any curriculum to any user or group of users.

In addition to the functionality described above the User Console will have background logging that will allow the project team to receive meaningful information about the use of the User Console. This information will help in tailoring the console as well as analysis of user habits and the degree to which a user is utilizing the system.

In order for this application to be useful in the future, consideration must be made for the extension of the privilege hierarchy as well as added functionality.

#### **4.0 Evaluation**

In order to determine the success of the User Manager the system will be presented to users who will then give feedback on their experiences. There will be two types of evaluation. The first phase will be a sit down with the users in which the user will be instructed on how to use the tools and encouraged explore the functionality. After the basic introduction the user will then be observed using the system. The second part of the evaluation will involve unsupervised usage of the tools were a user will be asked to complete several tasks with the tools and report on his/her experience with the tools.

I will evaluate the time it takes to package and distribute content less the time to construct the individual Assignments. The results for the evaluations will then be combined with log data to form a final evaluation of the success of the tool set and adjustments to the tools will be made to incorporate feedback.

It has been reported that on average it takes 20 hours to produce one hour of

content for an Intelligent Tutoring System [12]. As a measure of success I hope to cut this time in half to 10 hours of development time for one hour of Intelligent Tutoring System content.

## **5.0 Conclusion**

The purpose of this thesis is to “close the loop” in the Assistments Project by allowing self-management of all core system functionality. The tool set that is needed to achieve this goal is one that includes user and curriculum management that is accessible to even the most novice user. The success of this work will be determined by a usability study that will evaluate the ease of use and the ability of the tools to aid in the creation of a comprehensive learning solution.

## **6.0 Schedule**

**January – April 2005** – Initial Planning.

**May – July 2005** – Create the initial release of the User Manager.

**August 2005** – Complete final release of the User Manager

**August – September 2005** – Test the User Manager on system users, receive constant feedback, and incorporate feedback

**October – November 2005** – Prepare thesis presentation and defend thesis.

**December 2005** – Incorporate reader suggestions.

## 7.0 References

1. Razzaq, L, Feng, M., Nuzzo-Jones, G., Heffernan, N.T. et. al (2005). The Assistment Project: Blending Assessment and Assisting. To appear in the Proceedings of the 12th Annual Conference on Artificial Intelligence in Education 2005, Amsterdam.
2. Nuzzo-Jones, G., Walonoski, J.A., Heffernan, N.T., Livak, T. (2005). The eXtensible Tutor Architecture: A New Foundation for ITS. Poster, to appear in the Proceedings of the 12th Annual Conference on Artificial Intelligence in Education 2005, Amsterdam.
3. Massachusetts Comprehensive Assessment System,  
<http://www.doe.mass.edu/mcas/>
4. Turner, T.E., Macasek, M.A., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K. (2005). The Assistment Builder: A Rapid Development Tool for ITS. Poster, to appear in the Proceedings of the 12th Annual Conference on Artificial Intelligence in Education 2005, Amsterdam.
5. Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8, 30-43.
6. Morgan, P., & Ritter, S. (2002). An experimental study of the effects of Cognitive Tutor® Algebra I on student knowledge and attitude. (Available from Carnegie Learning, Inc., 1200 Penn Avenue, Suite 150, Pittsburgh, PA 15222)
7. Murray, T. (1999). Authoring Intelligent Tutoring Systems: An Anylysis of the State of the Art. *International Journal of Artificial Inelligence in Education*, 8, 30-43.
8. Anderson, J.R., (1993). *Rules of the Mind*. Hillsdale, NJ: Erlbaum.
9. Crowley RS, Legowski E, Medvedeva O, Tseytlin E, Roh E, Jukic DM. (2005). An ITS for medical classification problem-solving: Effects of tutoring and representations. Submitted to AIED 2005.
10. Vendlinski, T., Niemi, D., Wang, J., Monempour, S., Lee, J. (2005). Improving Formative Assessment Practice with Educational Information Technology. American Educational Research Association 2005 Annual Meeting.
11. Open Learning Initiative (OLI), <http://www.cmu.edu/oli/>

12. Koedinger, K. R., Alevan, V., Heffernan, T., McLaren, B. & Hockenberry, M. (2004) Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. Proceedings of 7<sup>th</sup> Annual Intelligent Tutoring Systems Conference, Maceio, Brazil. pg.162-173