

EarthTutor: A Multi-Layered Approach to ITS Authoring

Kristen PARTON, Aaron BELL, and Sowmya RAMACHANDRAN
Stottler Henke Associates, Inc., San Mateo, CA
{parton, bell, sowmya}@shai.com

Abstract. This paper presents the EarthTutor authoring tool, a multi-layered system designed to remove the technical hurdles preventing teachers from having full control over the structure of their lessons, without sacrificing the power and technical flexibility required for an effective ITS implementation. EarthTutor reduces the ITS design problem to a series of related but independent layers. Each layer in the process hides the implementation details of lower layers, while providing a foundation for higher levels. This approach allows a core group of advanced authors to do the more complex authoring tasks and export their work to a wider audience of novice authors, who complete the authoring process.

1. Introduction

One of the primary goals of creating an authoring tool for Intelligent Tutoring Systems (ITSs) is to allow subject matter experts (SMEs), usually non-programmers, to easily build tutoring content. However, in [4], Murray argues that ITS design is by nature a difficult task, and requires “an adequate understanding of both the representational structure and the design process [of the ITS].” Because designing ITSs involves creating multifaceted components and the complex interactions between them, authoring tools can end up being very complicated, difficult-to-learn applications. On the other hand, creating an easy-to-use authoring tool may unduly constrain the ITS design, resulting in shallow or overly simplistic ITSs with no room for customization by the author. The EarthTutor authoring tool attempts to be both powerful and user-friendly by splitting the authoring into multiple layers with different interfaces. Each layer can then be developed by a different group of authors, which shifts the burden of complexity onto a small, core group of advanced authors, while allowing a larger group of novice authors to access the power of the authoring tool through a simpler interface.

2 Authoring Tool Overview

The EarthTutor ITS was designed for NASA to teach remote sensing image processing, a domain in which students analyze satellite data using an image processing application. EarthTutor’s objective is to convince professors and teachers to switch from using pen-and-paper course materials in a computer lab to using an interactive, adaptive ITS integrated with software the students are already using. Full integration with the host application allows EarthTutor to closely monitor the student as he completes real-world tasks in the application, as well as add instructional annotations to the application. The student is presented with a series of cards, which contain interactive behaviors embedded in HTML pages. Behaviors consist of questions and real-world tasks the student must complete in the

host application. Authors specify the logic behind the tutoring behaviors, which includes monitoring the student's actions, presenting feedback, and updating the student model.

Figure 1 shows how components created in the EarthTutor authoring tool are used to build the student interface. The EarthTutor ITS core contains the general ITS components, such as student modeling, instructional planning, and the student GUI. Since EarthTutor works inside of a host application, there is also a plug-in layer that contains host-specific functionality (for instance, getting the active satellite image).

Both the ITS core and the plug-in layer export primitive functions called *actions* and *predicates* that allow interactive behaviors to access the ITS and host application. The advanced author interface contains a graphical interface for defining behaviors as flow charts (described in detail in [3]). The author builds flow charts using actions, predicates and other flow charts. Flow charts are parameterized and reusable, so flow charts can build upon other flow charts. Unlike a Finite State Machine, each behavior can also have local variables. Advanced constructs such as creating polymorphic behaviors, exception handling, and behavior scheduling parameters are also available for finer control, and the flow chart toolkit comes with an interactive debugger for rapid prototyping and development. Flow charts are exported to the novice interface as a *behavior library*.

In the novice interface, the author defines a hierarchical course structure, with a course containing labs, and labs containing cards. The author can set properties for the courses, labs and cards such as prerequisites and student modeling parameters. Each card contains HTML content and *behavior templates*. Behavior templates allow the novice authors to invoke the behaviors designed by the advanced authors by supplying the necessary parameters. Adding a behavior template to a card indicates 1) that the flow chart linked to the template should be executed when the card is displayed, and 2) that the student interface should replace the template with a UI component (defined by the advanced author in the flow chart). Behavior templates allow novice authors to tailor tutoring behaviors to their own pedagogical needs using parameters, but the interface is reduced to WYSIWYG HTML and simple forms.

The end result is an ITS that is much more interactive than a typical web-based ITS, and therefore, better at helping and assessing the students. For example, to teach a student how to calibrate an image, a less interactive ITS would explain the steps involved, ask the student to calibrate the image, then ask a follow-up question to check that the image was properly calibrated. In EarthTutor, a tutoring behavior would: 1) ask the student to open an image 2) wait for the student to open a file, then check that the file is correct 3) ask the student to use the calibration function and 4) wait for the student to calibrate, then check

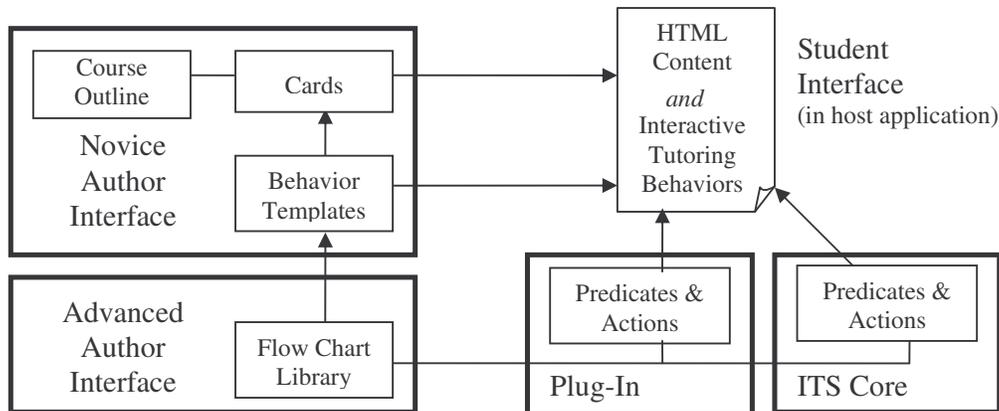


Figure 1. EarthTutor system architecture. The ITS core contains the student model, instructional planning, and displaying the student UI. The plug-in contains host-application specific functionality; e.g. allowing us to monitor the student's use of the image processing software.

the parameters of the calibration function. If the student uses the wrong parameters, the tutor will ask the student to undo the calibration and try again. Afterwards, the tutor can ask follow-up questions to make sure the student understood how the calibration works. Next time the student needs to calibrate, the tutor will simply ask him to calibrate an image, and expect him to remember the steps.

3 Discussion

The goal of the EarthTutor authoring tool was to provide content developers with control and flexibility, without sacrificing ease-of-use. Some authoring tools are designed to be easily used by classroom teachers, but either do not allow the author full control, or do not involve very much interactivity with the student. For instance, in REDEEM [8], interactions are limited to asking several types of questions, prompting the student to take notes, or directing the student's attention to another activity. REDEEM allows fine control over teaching strategy, but the teaching content is not very interactive. EarthTutor actually monitors the students as they complete real-world tasks, which enables the tutor to give more useful feedback.

In a multi-layer model, the inner layer can be thought of as a meta-authoring tool, which advanced authors use to create specialized, easy-to-use authoring tools for novice authors. As Murray notes in [3], this approach allows the authoring tool to "maintain depth, breadth *and* usability." Like the three-level model used by EASE [1], our model exploits the meta-authoring concept to provide power and usability. However, since the EarthTutor authoring tool allows users to switch between novice and advanced authoring interfaces, it is more flexible than a model which separates the meta-authoring level completely.

Separating the layers allowed the authoring work to be split between a small advanced group that designed behavior libraries, and a larger group of novice authors that used the flow charts. The behavior libraries also made it easier for SMEs to develop content, because the core Earth Science behaviors could be shared across the various labs and sub-domains (Oceans, Ozone, etc). Efficiency is a key factor in the authoring process, since creating an ITS is very labor intensive [4]. We believe that the multi-layer approach to ITS authoring increased authoring efficiency for EarthTutor, and detailed evaluations are planned in summer 2005 to get quantitative results on efficiency as well as usability.

Acknowledgements. This research is supported by NASA Small Business Innovation Research (SBIR) grant NNA04CA11C.

References

- [1] Ainsworth, S. & P. Fleming: Teachers as Instructional Designers: Does Involving a Classroom Teacher in the Design of Computer-Based Learning Environments Improve Their Effectiveness? In Gerjets, P., Kirschner, P. A., Elen, J. & Joiner, R. (Eds.) (2004). *Instructional design for effective and enjoyable computer-supported learning. Proceedings of the first joint meeting of the EARLI SIGs Instructional Design and Learning and Instruction with Computers*, pp 283-291.
- [2] Aroyo, L., Inaba, A., Soldatoba, L. & Mizoguchi, R.: EASE: Evolutional Authoring Support Environment, Proceedings of Intelligent Tutoring Systems (2004), 140-9.
- [3] Fu, D., Houlette, R., Jensen, R. & Bascara, O.: A Visual, Object-Oriented Approach to Simulation Behavior Authoring, Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (IITSEC) 2003.
- [4] Murray, T.: Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art, *International Journal of AIED* (1999), 10(1), 98-129.