Learning State
An XML based course editor for online instruction

Terence C. Ahern
David Dean
Roman Taraban
Ben Walton
Texas Tech University

Abstract: Learning State is a lesson editor that creates a control file in XML so that a designer can not only sequence lesson content but also determine how a student should engage the content. Further because of its inherent structure of XML, lesson content will be open and easily modifiable. Learning State is designed to integrate three separate parts: content, state sequence and the delivery model. We used XML to model the content by creating a DTD that orders the content to a specific type of delivery model. For example in factual information we used a linear-temporal model whereas for the concept attainment model we modeled the various examples and non-examples of specific attributes.

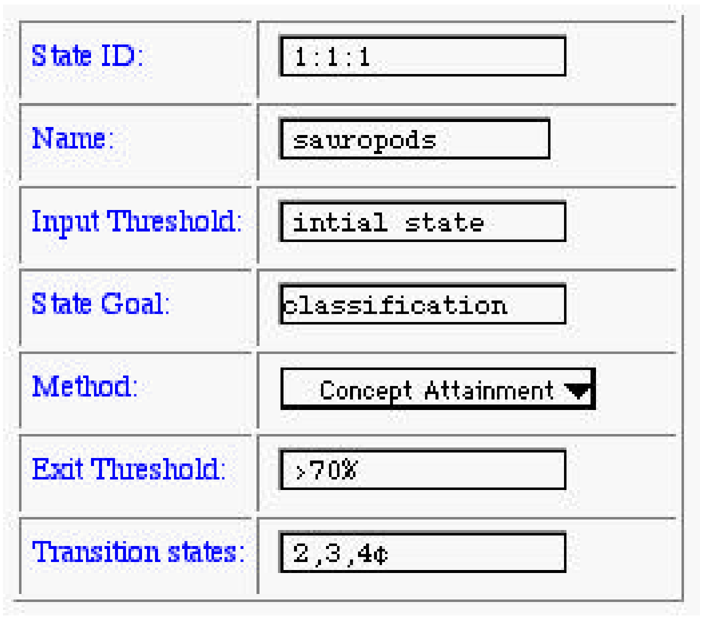
Problem

Courseware is different from other types of applications. It not only has to work flawlessly but also has to orchestrate the correct instructional delivery strategies. Originally, the delivery model was limited to direct instruction so that most courseware was nothing more than electronic workbooks or at best simple tutorials. As hardware improved, programming environments evolved so that delivery models were integrated with authoring environments. Even though these development systems revolutionized the way we create courseware there was tradeoff. The developer must Òbuy intoÓ the underlying delivery model. The developer had either tailor the intended outcome to the available technology or create proprietary solutions built from a higher level language. Either solution made the instructional content difficult to modify increasing development and delivery costs. The solution lies in creating an open development system that will be responsive but easy to use.

LearningState

LearningState is built around the Extensible Markup Language (XML) and Java 1.2 making it flexible and easy to use. We used XML to model the content by creating a DTD that orders the content to a specific type of delivery model. For example in factual information we used a linear-temporal model. Further, by using the discrete state model (Ahern 1999) we can create an appropriate sequence of sub-goals based on individual student states.

LearningState allows the designer to add, edit or delete state-nodes using a tree model. A popup window appears to create or edit a state description.(see Figure 1)



State ID:	1:1:1
Name:	sauropods
Input Threshold:	intial state
State Goal:	classification
Method:	Concept Attainment
Exit Threshold:	>70%
Transition states:	2,3,4¢

Figure 1: Example of state description window

In this example we have created an initial state node built around the dinosaurs. The goal of this state is to have the student understand the process of classification. To achieve the state, the designer selected the concept attainment model using a pull-down menu. A new window will which prompts the lesson designer to enter the appropriate content links for the concept attainment model. We can add additional delivery methods as necessary. Finally, the developer has to decide on a criteria or series of criteria that links this state to the next appropriate state as evidenced by the student performance. Once the state is defined the designer can then save the course and it is written out as an XML file.

Implications

The implications for an open protocol-based educational technology system are crucial. First it separates content from the delivery technology allowing for the rapid development of appropriate educational material regardless of the delivery technology. Developers would be free to concentrate on effective and appropriate instructional materials targeted to what teachers need for the classroom. Further, it would provide teachers with the ability to construct, modify or share computer-based materials for their classrooms thereby reducing the time needed to create course materials while substantially lowering costs.

Further development

Further development of a client as well as Java Servlet are planned to complete the development phase of the project. Further we plan more extensive development of different delivery models ranging from direct instruction to the social constructive paradigms.

References

Ahern, T. C. (Spring, 1999). The discrete state theory of instructional design. Paper presented at the annual meeting of the American Educational Research Association, Montreal