A Generic Presentation Modeling System for Adaptive Web-based Instructional Applications

José A. Macías and Pablo Castells
E.T.S. de Informática
Universidad Autónoma de Madrid
Campus de Cantoblanco, 28049 Madrid, Spain
+34-91-3482241, +34-91-3482284
{jose.a.macias, pablo.castells}@ii.uam.es

ABSTRACT

We propose a generic presentation system for adaptive educational hypermedia that is highly independent from domain knowledge representation and application state management. Generality is achieved by providing a framework for the definition of ontologies that best fit specific domains and/or authors. Presentations are described in terms of ontology object classes and relations. An explicit presentation model, separate from course contents, is used to provide course designers with extensive control over the generation of all aspects of presentation, at a moderate development cost.

Keywords

Adaptive Hypermedia, Computer-Based Learning, Presentation Design, Knowledge Representation, Authoring Tools.

INTRODUCTION

With the rapid development of computer-based online education, learners are gaining increasing autonomy, and instructional applications are reaching an unprecedented diversity of users. In this context, a growing interest has been raised on the development of web-based educational software that adapts automatically to student and platform characteristics. A common (explicit or implicit) prioritary concern to all research work in this area is that of finding an appropriate representation for pedagogical knowledge. Our work aims at providing a generic presentation system for adaptive hypermedia that makes minimum assumptions about how domain and instructional knowledge are represented. It is our purpose to provide courseware designers with a simple specification paradigm for non-trivial adaptive presentation constructs, that can be used with different course management systems.

There are many ways to structure knowledge. To name a few, ELM-ART [1] describes course curriculum in terms of hierarchical lessons with prerequisite, related and outcome concepts; the AHA [2] domain model consists of interrelated concepts, fragments and pages; in Tangow [3] courses are structured using conditional task hierarchies. In order to allow for different approaches, our system supports the definition of made-to-measure domain ontologies for the description and conceptual structuring of subject matter (as in [4]). Once an ontology is defined, designers build courses by creating domain objects and relating them together using the conceptual vocabulary defined by the ontology. Course presentation is designed by defining an explicit presentation model where presentations are associated to ontology object classes and relations.

DOMAIN ONTOLOGIES

Domain ontologies consist of a set of classes that best suit the nature of a specific domain or that reflect the particular vision of a specific author on the domain. In our approach, ontologies can be defined with a high degree of freedom. Classes can be very generic, like Concept, Lesson, or more specific, like Algorithm, Theorem, or Definition, as the designer sees fit. Ontologies include terms for subject matter information (e.g. a theorem has a statement and a proof), pedagogical information (e.g. lessons have levels of difficulty), and run-time state information (e.g. whether a concept is known by the student).

All this knowledge is captured by defining attributes for classes, and relations between classes. Two predefined root classes are provided: Topic and Fragment, for ontology designers to subclass. Topic's are presented to the end-user in a separate page, while several Fragment's can be inserted in the same page. A predefined subclass of Fragment, AtomicFragment, carries content media (HTML code). In addition to a domain ontology, simpler data structures are defined by the designer to describe user profiles, platform characteristics, and other aspects considered relevant for adaptive presentation.

Courses are constructed by creating semantic networks of domain objects, using the classes and relations defined in the ontology. Domain objects are used as an overlay model at run-time to reflect the progress of the user through the course and her/his changing state of knowledge. Many adaptive hypermedia systems allow the topic network itself to be dynamically set up and modified at runtime (e.g. [3]), and so do we. Our presentation system takes care of how this may affect presentation, but how course structure and state are updated is external to the presentation system.

ADAPTIVE PRESENTATION MODELING

Existing adaptive hypermedia systems support adaptive content selection and adaptive course sequencing, but they miss an explicit presentation model. As a consequence, presentation is partly intermingled with contents (as in [2]), and partly set up automatically by the system according to rigid design choices (e.g. link annotation) that the designer cannot configure (see [3] for instance).

In our system, separation of content and presentation is achieved by defining a *presentation template* for each class of the ontology. Templates define what parts (attributes and relations) of a topic must be included in its presentation, their visual appearance and layout. Though we plan to build graphical editing tools for presentation authoring, templates are currently defined using an XML-based textual language similar to XSL style sheets. At runtime, when the student moves to a course topic by traversing a link, the topic instance and its matching template are found by the system, from which an HTML page is generated automatically and sent back to the student's web browser (see Figure 1).

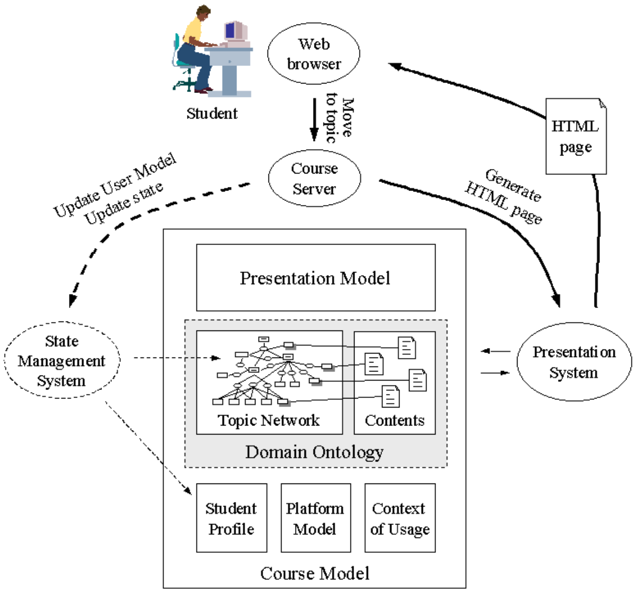

Figure 1. Course model and runtime architecture

Templates are complemented with *presentation rules* that govern aspects like link generation, correspondence between link styles and topic states, ordering and layout of (fragment or link) lists, and the generation of built-in presentations for topic network subsets like linked lists and trees. Rules simplify template syntax by generating adaptive presentation constructs involving relations between domain objects from very succinct high-level descriptions. For instance, assuming an ontology for Geography courses has been defined including classes like Country and City, to include information about important cities in a country's page the designer only needs to refer to the 'has-city' relation in the presentation template for class Country. The system automatically takes care of deciding whether to insert city details into the generated page, or to generate a link for each city, which style and annotation are used in the latter case, and how all the pieces are laid out. In doing so, the system analyzes whether the relation is simple or multivalued, the class of the involved topics or fragments, their state, and other conditions, if any, stated by the designer. The designer can modify existing rules and define her own.

Adaptivity is achieved by setting conditions on templates, on parts inside templates, in presentation rules, and over relations between objects in the domain model itself. These conditions can test properties of the user model (overlay model and user profile), characteristics of the platform, and any other aspect that should influence presentation, like course requirements, student's goals, usage modes (e.g. learning vs. consultation), etc.

CONCLUSION

In most cases, our presentation system will not work alone. After a topic ontology is built, a runtime system is needed to set up and update topic networks, as illustrated in Figure 1. Because the dynamic generation of presentation is a separated mechanism from the user and domain model update mechanisms, our presentation system can be used with different courseware support tools.

Our approach allows the specification of presentation independently from content construction, enhancing presentation reuse and consistency, thus reducing the development cost. Among our future work plans we include the development of a graphical editing tool where authors can customize presentation models by example, by editing generated HTML pages. The creation of this kind of tool is not possible without an explicit declarative model of presentation.

While our research is based on observations taken mainly from the educational field, no essential assumptions have been made as of yet that would prevent it from being applied to other domains.

ACKNOWLEDGMENTS

The work reported in this paper is being partially supported by the Spanish Interdepartmental Commission of Science and Technology (CICYT), project number TEL1999-0181.

REFERENCES

1. Weber, G. and Specht, M., User modeling and Adaptive Navigation Support in WWW-based Tutoring Systems, in *Proceedings 6th International Conference on User Modeling (UM97)*, Sardinia, Italy, 1997.
2. De Bra, P., Design Issues in Adaptive Hypermedia Application Development, in *Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the World Wide Web*, Toronto, Canada, 1999, 29-39.
3. Carro, R. M., Pulido, E., Rodríguez, P., Dynamic generation of adaptive Internet-based courses. *Journal of Network and Computer Applications 22*, 1999, 249-257.
4. Murray, T., Authoring Kowledge Based Tutors: Tools for Content, Instructional Strategy, Student Model, and Interface Design. *Journal of the Learning Sciences 7*, 1, 1998, 5-64.