Template Server Architecture

Alexei Tretiakov, Yanmin Shi, Hong Hong and Kinshuk

Department of Information Systems, College of Business, Massey University, New Zealand

Abstract

Reuse of teaching resources is recognized to be essential for cost-effective development and maintenance of on-line teaching environments. We offer a novel approach to the reuse and sharing of on-line teaching unit designs, based on serving unit templates in a distributed environment. The approach suggested will increase the level of reuse, simplify and speed up material creation by non-technical domain experts, and promote the availability of teaching materials in a variety of formats.

1. Introduction

In today's teaching, teachers spend lots of time on courseware development activities, such as creation of lecture slides, study guides, and tutorial materials. Increasingly, teachers also have to make their materials available on-line. These efforts are very time and resource consuming. The reusability of courseware has been explored to address the difficulty and cost in courseware development.

The concept of reuse has been successfully deployed in software development process. In that discipline, the concept of reuse is defined as having two phases: development for reuse and development with reuse [1]. Similar to software development, the reuse of courseware largely impacts on courseware development phase of computer-assisted education delivery, in particular in Web-based teaching/learning [2].

At present, courseware reuse is quite common at two levels: at design level, when teachers reuse approaches to course organization recognized to be a good practice (e.g., providing a short review of the previous lecture and giving main points to be covered in the current lecture at the start), and at the level of re-using labor-intensive multimedia, such as images [3]. Thus, reuse is practiced at rather high ("design patterns") level, and at rather low level (reuse of what are essentially primitive data items). In addition, standards for educational content packaging and educational content metadata are mainly oriented at supporting the ability to reuse entire courses or rather large parts of courses [4, 5].

In this paper, we describe an architecture that supports the reuse of online content at the level of individual screen designs. A prototype implementing this architecture is also described.

2. Template Server Architecture subsystems and their responsibilities

Template Server architecture is comprised of three major subsystems, outlined in Table 1: Template Server, Content Server, and Client.

Table 1: Subsystems and their responsibilities

Template Server	Content Server	Client
Serve templates Serve utilities for rendering content units Store usage history and other user-specific information Serve adaptive template prompts	Store content units Render and serve content units Expose stored content units via value-added interfaces, such as full text search	Use templates to offer user interface for creating and editing content units Prompt for a template to be used in the creation of a new content unit

Template Server is the subsystem responsible for storing templates and making them available over the network.

Templates for a large variety of documents are stored in the Template Server. Here, a template is a program, providing to the user an interface for entering information to be formatted into a document (a "content unit"). One simple form of template we envisage is an HTML form.

Once a template is displayed, the user enters the information necessary to complete the document, and commits the resulting document, which is forwarded to another subsystem, the Content Server. Documents may be submitted as semantic markup, or as presentation markup (e.g. as raw HTML). If the document is submitted as semantic markup, it should include a URI of the template used to create it. The Content Server generates a unique URI for the document and stores the document under it. Optionally, the URI may be returned to the client that submitted the document, to be included in a web-based course.

Proceedings of the The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03)
0-7695-1967-9/03 $17.00 © 2003 **IEEE**

The Content Server makes the document available at the document's URI. Optionally, the Content Server can also expose the documents it holds in other ways, such as via search engines or via hierarchical directory-like structures.

For documents stored as semantic markup, the Content Server may offer services allowing to render and to download them on demand in a variety of formats, e.g. PDF, HTML or Microsoft Word. New formats can be accommodated as they appear. The utilities used to render the document (e.g. XSLT transforms), are to be deposited at the Template Server with the template used to create the document. To render the document, the Content Server has to retrieve the transform utilities using the template URI, stored as part of the document.

A semantic markup document stored in the Content Server can be modified any time by using the template that was used to create it. Thus, every time the document is modified, the corresponding template should be obtained from the Template Server.

Documents that are stored by using presentation markup (e.g. HTML) can only be edited directly, by modifying the markup source. For them, the ease of use provided by the template's user interface is only available at creation time. On the other hand, semantic markup documents rely on their templates for creation, for rendering (in any number of formats supported by the transforms provided with the template) and for editing (modification).

Apart from serving specific templates, Template Server may serve adaptive prompts simplifying the task of picking a template for a particular purpose. Adaptive Prompt service would simplify the task of choosing an appropriate template by offering an interface based on the access history, or on other factors, such as user position in an organization.

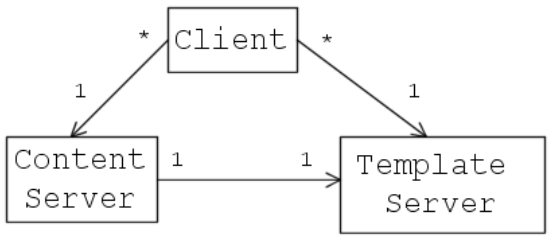

Figure 1. Static view of the Template Server Architecture

3. Prototype implementation

A prototype system has been developed that implements the architecture discussed above. The prototype is incorporated into a Web-based course authoring software, that was developed for our broader teacher modeling research project, and can be trialed by accessing the project web site [6]. The current prototype involves only the case of presentation markup (units are stored as HTML files).

4. Conclusions and future work

We presented a novel architecture for the creation of on-line educational content, based on re-using content unit designs by making them available as on-line templates. The architecture allows the incorporation of adaptive template selection based on teacher modeling, thus providing opportunities for further increase in content authoring productivity.

We provided a proof-of-concept implementation realizing the most important features of the architecture.

In the near future we expect to implement the full architecture, including semantic markup unit storage and template-based unit editing, and to conduct a formal evaluation of the impact of the proposed architecture on content creator productivity.

References

[1] H. Mili, F. Mili and A. Mili, "Reusing software: Issues and Research Directions", *IEEE Transaction on Software Engineering*, v. 21, 1995, pp. 528-561.

[2] P. Vittorini and P.D. Felice, "Issues in Courseware Reuse for a Web-based Information System", *NAWeb Proceedings*, New Brunswick, Canada, 2000.

[3] J.C. Teixeira, C. Páris, J. Madeira and J.B. Lopes, "Courseware re-use or courseware to reuse?", *Graphics And Visualization Education*, Coimbra, Portugal, 1999.

[4] Dublin Core Metadata Initiative, `http://dublincore.org/groups/education`.

[5] IMS Global Learning Consortium, `http://www.imsproject.org`.

[6] Teacher modeling project, retrieved 2002 from `http://ispost.massey.ac.nz/teacher_model/logon.htm` (using 157736 as the course name for access).