

A Component-based Approach for Adaptive, Dynamic Web Documents

Zoltán Fiala

Michael Hinz

Klaus Meissner

Frank Wehner

Dresden University of Technology, Department of Computer Science

Heinz-Nixdorf Endowed Chair for Multimedia Technology

D-01062, Germany, +49-351-463-38516

{zoltan.fiala, michael.hinz, klaus.meissner, frank.wehner}@inf.tu-dresden.de

ABSTRACT

Today's WWW lacks of technologies compensating the additional effort of creating and maintaining Web applications, which are automatically adapted for different clients and user preferences. This paper introduces a declarative, component-based approach for adaptive, dynamic web documents on the basis of XML. Adaptive web components on different abstraction levels are defined in order to support flexible reuse, effective Web page authoring and generation. Finally, the stepwise process of document generation is explained.

Categories and Subject Descriptors

D.2. [Software Engineering]: - D.2.2. Design tools and techniques – *user interfaces*. D.2.13 Reusable Software - *reuse models*. H.5.4 [Hypertext/Hypermedia]: Architectures, User issues. I.3.6 [Methodology and Techniques]: Device Independence.

General Terms

Management, Design, Languages

Keywords

Component-based Web Engineering, Adaptive Hypermedia

1. INTRODUCTION

The World Wide Web is rapidly changing from a presentation to a cooperation and communication medium offering highly personalized interactive Web applications, accessible via different multimedia end devices. This necessitates the quick generation and delivery of up-to-date information, which is automatically adapted to the respective presentation interface and user preferences. However, conventional document formats for the Web are hardly applicable for managing adaptive Web applications. Their coarse-grained implementation model suffers from the lacking separation of content, structure and layout, thus preventing the reuse of fine-granular functional, semantic and layout elements in a flexible way. Furthermore, no sufficient mechanisms for describing adaptivity are provided.

Existing approaches supporting object-oriented Web engineering, e.g. OOHD [3] or RMM [2] aim at reusing design concepts, but

do not consider their automatic mapping to reusable implementation artefacts. In [1] the application of component-oriented methods to Web engineering is proposed, allowing for reuse on both the conceptual and the implementation level. However, there is no support for defining adaptive behavior in a generic way. The approach proposed here introduces a component-based document model for efficiently creating and delivering highly adaptive web presentations. It does not focus on the conceptual design of Web applications, but on the challenge to reuse adaptable implementation artefacts.

2. THE DOCUMENT MODEL

2.1 Declarative Document Components

The model enables the composition of Web sites of configurable content components. These are documents or document fragments, instances of a specific XML-grammar representing adaptable Web content on different abstraction levels. Components' interfaces are described by attached metadata specifying their properties and adaptive behavior. Web sites are constructed by aggregating components and linking them to complex document structures. During generation these document structures are translated into Web pages in a specific output format adapted to a certain user model and/or client. Components are declared by a specific description language defined by XML schema. Figure 1 depicts the abstraction levels of components.

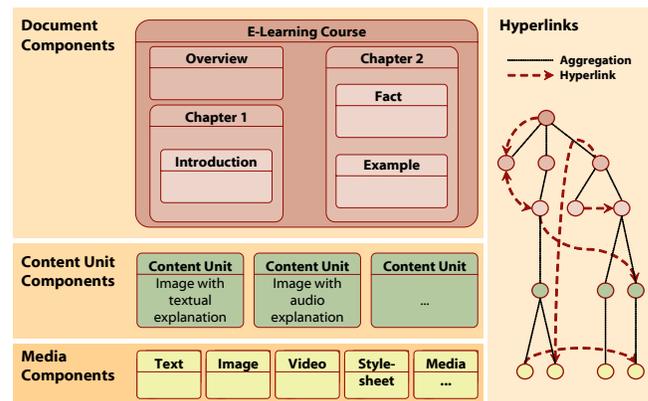


Figure 1. The Document Model

On the lowest level there are *media components* encapsulating media assets. These comprise text, structured text (e.g. HTML), images, sound, video, java applets and may be extended arbitrarily. Even whole documents (such as HTML-pages) can be

handled as media assets by extending them with appropriate metadata.

The second level combines media components belonging together semantically - e.g. an image with textual description - to so called *content units*. Defining such collections of components is a key factor of reuse. The spatial adjustment of the contained media components is described by specific client-independent layout properties allowing to abstract from the exact resolution and presentation style of the display or the browser's window.

Thirdly, *document components* are specified as parts of Web presentations playing a well defined semantic role - such as a news column, a product presentation or even a Web site. A content unit containing an image object and two text objects could be encapsulated to a document component "newspaper article". Document components can either reference content units, or aggregate other document components. This hierarchy of document components describes the logical structure of a Web site and is strongly dependent from the application context.

Whereas aggregation relationships are expressed on the level of document components, links are defined on the *hyperlink view* spanned over all component levels. Uni- and bidirectional typed hyperlinks based on the standards XLink, XPath and XPointer are allowed.

2.2 Supporting Adaptivity

The document model aims at modeling adaptivity by attaching metadata describing adaptive behavior to components on each abstraction level.

Firstly, adaptation is required on the level of media components in order to consider various client capabilities or other technical preferences by providing alternative media instances with varying *quality*. Secondly, on the level of content units the *number*, *type* and *arrangement* of inserted media components can be adjusted. Thirdly, adaptation of document components concerns the whole *component hierarchy* resulting in different variations of component trees for different users and devices. Finally, adaptation on the hyperlink view enables to define personalized *navigation* structures.

In order to define adaptive behavior each component definition may include a number of variants. As an example, the definition of an image object component might include two variations, one for color and another for monochrome displays. Similarly, the number, structure and arrangement of subcomponents within a document component could also have different variants according to a certain adaptation aspect. The decision, which alternative is chosen and inserted into the current Web presentation is always made during document generation by a corresponding transformer (e.g. an XSLT stylesheet) implementing a certain selection method parametrized by the current user model. At authoring time the creator of a Web application selects the appropriate adaptation method from a repertoire of such methods. This separation of describing component variants (in the component itself) and adaptation logic (in transformers) allows the reuse of a given component in different adaptation scenarios.

3. DOCUMENT GENERATION

Document generation is based on a stepwise pipeline concept in order to achieve code reuse and higher performance through caching mechanisms. In the beginning, a complex document according to the proposed model is dynamically assembled from a component repository according to a user's request. It contains

information about subcomponents to be included and metadata describing adaptation rules. All possible presentation forms of the component concerning content, layout, and structure are encapsulated. Depending on the current user model - which comprises both the user's semantic preferences and her client's technical capabilities - it is subdued to a series of transformations. Each step considers a certain aspect of adaptation by performing conversions to the document (selection and configuration of components). Finally, a Web document in a specific output format (HTML, WML etc.) is generated. Figure 2 depicts a possible pipeline with three steps, namely adaptation to a client class, to semantic user preferences and lastly to specific technical capabilities. The document generator was realized on the basis of the pipeline-based publishing framework Cocoon.

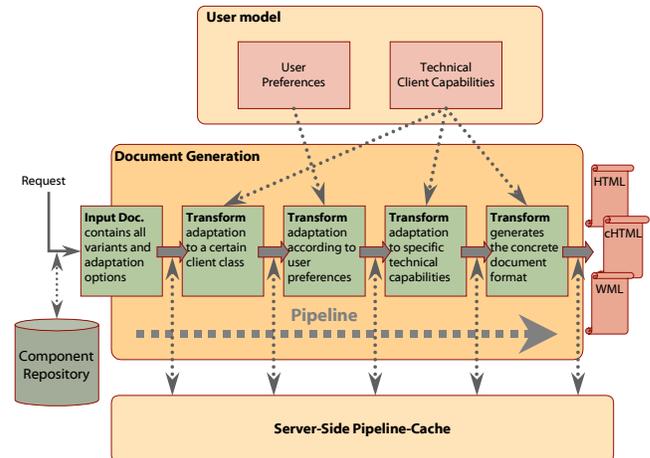


Figure 2. Stepwise Document Generation

4. CONCLUSION AND FUTURE WORK

This work introduces a component-based document model for dynamic, adaptive Web applications. Fine-granular components on different abstraction layers are defined in order to support flexible reuse during page composition and effective generation of adaptive Web documents.

Future work includes the refinement of metadata schemas for representing user models and client information. A detailed performance analysis of the pipeline-based document generation is also to be done. Another activity will focus on the authoring process, especially on the adoption of existing hypermedia design models for adaptive, component-based Web sites. Furthermore, a component repository for storing, retrieving, configuring, and publishing adaptive content components needs to be designed and implemented.

5. REFERENCES

- [1] Gaedke, M., Segor, C., and Gellersen, H.-W. WCML: Paving the Way for Reuse in Object-Oriented Web Engineering. ACM Symposium on Applied Computing (SAC 2000), Villa Olmo, Como, Italy, March 19-21, 2000
- [2] Isakowitz, T., Stohr, E.A., and Balasubramanian, P. RMM: A Methodology for Structured Hypermedia Design; Communications of the ACM, 1995
- [3] Schwabe, D., Rossi, G., and Barbosa, S.D.J. Systematic Hypermedia Application Design with OOHDM. UK Conference on Hypertext, 1996