From State to Structure: an XML Web Publishing Framework

M.C. Norrie and A. Palinginis

Institute for Information Systems, ETH Zurich, CH-8092 Zurich, Switzerland
{norrie,palinginis}@inf.ethz.ch

Abstract. We present the main features of a system designed to support the development and delivery of web applications through concepts for modularity, reuse and rapid prototyping. The system is based on an extended object-oriented database system that manages both application data and publishing information in terms of content, structure and presentation. The process of information delivery uses dynamically generated XML documents for both content and presentation based on state-dependent composition of components and templates.

1 Introduction

Although a large variety of tools and technologies are available to support the publishing of static and dynamic data on the web, many of these lack a well-defined declarative model. The focus of many commercial systems such as [Inc00,Cor01,AG02] has instead been to offer sophisticated tools and architectures for delivering the information in a fast and scalable way. Although they offer component reuse, they do not rely on an explicit model, but rather on specifications and usage guidelines drawn from experience gained in commercial projects.

Within the research community, this problem has already been identified and a number of models and systems [FFK+97,FLM98,MAM+98,BCF00] have been proposed that exploit knowledge gained over many years in the database and hypertext fields. These systems use structured data models for modelling both the underlying data and the web site itself. Most of them use a data model very close to relational databases and an SQL flavour to communicate with the database. They offer a general modelling and querying schema for the web and WebML [BCF00] even defines a powerful and detailed model for the application visual interface.

In contrast, our work focuses on the construction and management of web applications, rather than purely with design. Flexibility is a major requirement in such applications and a strict database-like environment for the structure and presentation of the site proves to be quite limiting. We want to parameterise the structure and presentation of the contents to enable the flexible and dynamic composition of pages in a state-dependent manner according to user interaction or development prescriptions.

Based on our experience in developing small to medium web-sites and experience working with a major content management system [Cor01], we believe that the design and implementation methodologies in such projects are of great importance. Many projects failures are caused by the lack of detailed requirements and their inappropriate late adaptation to the web environment. Additionally, customers often desire a running prototype even before contracting the project. We therefore advocate the use of flexible prototyping tools that support the entire development process, while ensuring the scalability of the approach.

Our approach was to integrate full web publishing support for both static and dynamic web sites into the OMS Pro object-oriented data management system [Wue00,KNW98]. The resulting OMSwe system was able to exploit both the semantically rich object model on which OMS Pro is based and its support for rapid prototyping. Our aim was to keep the model both simple and flexible, introducing a minimum number of new concepts into the system. It was also important that the system could be used either to develop new web sites from scratch or to publish existing databases on the web [NS96]. In this paper, we present the main features of OMSwe. We begin in Section 2 with a discussion of how XML technologies are used within the system. Section 3 then introduces the concepts of *contents* and *templates* which represent the logical components of web pages and their presentations, respectively. Section 4 describes the representation of state and support for context-dependent delivery of information. Concluding remarks are given in Section 5.

2 Use of XML Technologies

XML technologies are used extensively in the web publishing world to support both the re-use of content and context-dependent delivery [Gol02]. Content may be represented as XML documents and selected content and presentation generated through the use of XSLT templates and transformers. Universal client access is achieved by generating different XML formats such as XHTML for desktop browsers, WML for WAP-enabled phones and VoiceXML for speech phones.

Existing approaches tend to be document rather than information based. The concepts at the information level are primarily document concepts of text, image, video, URL etc. and document structure (chapter, section, paragraph etc.). The lack of semantic information makes it difficult to ensure consistency and to link pieces of content together that represent the same logical entity. For example, if information about a person appears on several pages of a web site, the content editor has to manually keep track of the fact that the associated content actually belongs together as a semantic unit. If information about a person is updated, or even the entire person deleted, it is difficult to track where to make the necessary changes and tedious to make them.

We instead manage information about logical entities as database objects and dynamically generate XML documents to represent particular context-dependent views on those entities. This approach facilitates changes, avoids inconsistencies and promotes further forms of modularity and reuse.

We therefore had to integrate an XML generator into OMS Pro to publish database objects as XML. The key constructs of the OMS Pro model include collections for the classification of objects and associations over collections to represent relationships between objects. Multiple instantiation and classification is supported and objects can be accessed through a particular type to create a *type view* of the object. A general OMS XML schema defines the XML representation of OMS objects and options control such things as the type view and the inclusion of associated objects as child elements. An application developer also has the option to define a specific XML format. Note that all data in OMS, inclusive of operations and metadata, is represented as objects and can therefore be exported as XML using the general purpose OMS XML schema.

3 Contents & Components

Rather than specifying pages, the model uses the notion of components to define abstractions for content and its structure. Complex nested structures can be defined through component composition. A component defines slots as extension points which can then be used to dynamically bind it to other component instances. Thus a page is a root component that includes further data or structural components. Every page, or part of it, is dynamically composed from component instances that best match the current context as defined by the request state.

Furthermore links can be defined for each component. Links are the state definition for the next request. These definitions are subject to an inheritance mechanism to allow the creation of links to components which have parent components that are unknown to the component generating the link. In this way, a component that is part of a page can link to a similar page that inherits the parent state and differs only in one subcomponent or its behaviour.

Components play the role of the controller according to the MVC pattern [GHJV95]. As described in the next section, they are context-aware and adapt to the current situation. Components are the content producers that are visualised by associated templates that are also subject to context controls.

4 Context & State

A general model of context enables applications to define their context state in terms of a set of characteristic variables. These variables are specified in the configuration object for the application, along with a set of default values. Our model is similar to that described for semi-structured data in [SG02].

Since we want both content and presentation to be context-dependent, we use a *web element* to represent an element of web component specification, whether it is a specification of content or presentation. A web element is actually defined by a set of `webElement` objects with the same alias and type (content or template), but different *activation contexts*. An activation context defines the set of context states in which the `webElement` can be used and it is specified in terms of a set of characteristics and a valid time interval.

For a given request, the particular webElement object used depends on a best match between the current context state and the characteristics and valid timespans of the set of webElements with the specified alias. The specification of defaults means that there is always at least one webElement that matches.

5 Concluding Remarks

We have described the main concepts integrated into the object data management system OMS Pro to provide full web publishing support. A notion of components as well-defined typed abstractions was introduced and these can either be bound to data or define structures which can be parameterised through the concept of slots. The generation of content, structure and presentation may all be subject to context-dependent factors. The resulting OMSwe system is well-suited to prototyping due to its flexibility and associated development tools. The prototyping data can either serve as a skeleton to build up a production scalable site, or, in the case of smaller applications, form the basis of an operational site.

References

[AG02] Inc. Aberdeen Group. ColdFusion MX: Raising the Return on Investment of Internet Application Development. White Paper, Apr 2002.

[BCF00] A. Bongio, S. Ceri, and P Fraternali. Web Modeling Language (WebML): a Modeling Language for Designing Web Sites. In *Proc. 9th Int. World Wide Web Conf.*, 2000.

[Cor01] Vignette Corp. Vignette Content Management Server. White Paper, Feb 2001.

[FFK+97] M. Fernandez, D. Florescu, J. Kang, A. Levy, and D. Suciu. STRUDEL : a Web Site Management System. 1997.

[FLM98] D. Florescu, A. Levy, and A. Mendelzon. Database Techniques for the World-Wide Web: A Survey. *SIGMOD Record*, 27(3), 1998.

[GHJV95] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns, Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[Gol02] C.F. Goldfarb. *XML Handbook*. 4th Edition, Prentice Hall, 2002.

[Inc00] Interwoven Inc. Application Development using Interwoven: Version 1.1. White Paper, Dec 2000.

[KNW98] A. Kobler, M. C. Norrie, and A. Würgler. OMS Approach to Database Development through Rapid Prototyping. In *Proc. 8th Workshop on Information Technologies and Systems (WITS'98)*, Helsinki, Finland, Dec 1998.

[MAM+98] G. Mecca, P. Atzeni, A. Masci, P. Merialdo, and G. Sindoni. The Araneus Web-Base Management System. In *Proc. ACM SIGMOD*, 1998.

[NS96] T. Nguyen and V. Srinivasan. Accessing Relational Databases from the World Wide Web. In *Proc. ACM SIGMOD*, 1996.

[SG02] Y. Stavrakas and M. Gergatsoulis. Multidimensional Semistructured Data: Representing Context-Dependent Information on the Web. In *Proc. CAiSE 2002*, May 2002.

[Wue00] A. Wuergler. *OMS Development Framework: Rapid Prototyping for Object-Oriented Databases*. PhD Thesis, Dept Computer Sc, ETH, CH-8092 Zurich, Switzerland, 2000.