erscheint in: Proc. of the Int. Workshop on Information Integration on the Web,
Rio de Janeiro, Brazil, April 2001

Data-Driven, XML-Based Web Management in Highly Personalized Environments (Position Paper)

Marcus Flehmig
University of Kaiserslautern, Germany
Department of Computer Science
email: flehmig@informatik.uni-kl.de

Abstract: In the domain of XML-based Web applications, sets of XML documents derived from heterogeneous data sources are to be managed. The deployment of XML offers a single and common data model which allows these sets to consider under a data-oriented perspective and to compose their elements to a single, logical XML document. The introduction of a logical representation of the underlying data by a so-called unified view enables comprehensive personalization and customization, because the document structure as well as its graphical presentation can be changed and adapted independently. The intention of this paper is to discuss the idea of building the unified view. Therefore, an overview of the corresponding architecture is given and basic strategies for the building process are presented.

Keywords: data-intensive web application, personalization, data integration, one-to-one delivery, declarative Web-site management

1. Introduction

The eXtensible Markup Language (XML) is getting pervasive in a continuously growing number of application domains. It provides a generic representation of (semi-)structured data as serialized text in a standardized way [1]. XML documents are of textual nature, but represent data as well. For this reason, two different points of view have emerged: a document-oriented as well as a database-oriented view [2]. Therefore, on one hand, document operations are needed, because XML documents are considered in their entirety, e. g., in the area of electronic publishing (POP: *Presentation-Oriented Publishing*). On the other hand, database operations are required for data extraction, data integration or data storage [3]. Especially in the domain of MOM (*Message-Oriented Middleware*) or data-intensive applications, XML documents are considered under a data-oriented perspective. In our context, data orientation indicates that only the data contained in XML documents is considered. The surrounding structure of the document and its physical representation is less important. The definition of the XML Query Requirements [4] shows the distinction between document-oriented and database-oriented XML documents as well. In both cases, data in the broadest sense is represented by a set of XML documents, but the requirements on processing, querying or storing are completely different.

The intention of this paper is to discuss the idea of providing a data-oriented view to integrated sets of XML documents. Hence, a single, logical document view helps to improve query support and to get a higher degree of data integration in today‘s XML-based Web applications. Based on that view, comprehensive personalization mechanisms can be applied. For example, Web Information Systems (WIS), Web Based Teaching (WBT), or portal systems could benefit in such a way that a personal view or a personal learning space can be provided. Furthermore, in the domain of WBT, users can manage learning units, discussion contributions, or annotations and can filter, restructure, or graphically present them in a personalized way.

In this paper, we begin with an overview of the related work. Based on the requirements of an appropriate architecture motivated in Sect. 3, we present our approach towards a system architecture in Sect. 4. Finally, we show basic strategies on how to build a single, logical document view.

2. Related Work

Declarative Web-site management has defined a new paradigm of Web management [5, 6]. The separation of physical and logical representations is identified to be a basic necessity by many approaches [2, 7, 8, 9]. These approaches rely on different data models for the representation of their data basis: the relational or entity relationship model [7], the model of (rooted) directed (labeled) graphs [9], or on XML itself. The W3I3 [7] project uses XML, but only to describe relational data, whereas the MIX [10] approach considers XML documents as the data basis itself and tries to find semantic similarities between elements of different content models. XML documents are integrated in such a way that documents with similar elements are combined [11].

3. Motivation

Because of the enormous growth of the World Wide Web (WWW), efficient management of Web data becomes more and more difficult. Up-to-date Web management must support many different authors providing content and integrating data sources, databases or whole application systems with transactional properties.

3.1 Data-Intensive, Web-Based Applications

In the domain of data-intensive, Web-based applications it is not sufficient to provide information exclusively by static or pregenerated documents. In contrast, data sources have to be integrated by using dynamic document generation (often in combination with aggregation of static documents, pregeneration or caching).

In general, data-intensive Web applications can be characterized as follows [7]:

- *simple functional requirements,*
- *basic transactional support,*
- *focus on interface organization and ease of navigation,*
- *support of one-to-one Web delivery (personalization),*
- *support of multi-device output generation (customization).*

3.2 Paradigm Shift in Web-Site Management

Declarative Web-site management has emerged as a new paradigm to cope with such requirements [5, 6]. Three main tasks of building a Web site are separated by this approach: data management, definition of the internal site structure and design of a proper external page presentation. This separation is obtained by the introduction of a logical representation on top of the underlying data layer. Differences in the various strategies of declarative Web-site management can be primarily found in the data model and in the query language used to define the view to the underlying data [12].

3.3 Implications

Due to the clear separation of physical and logical aspects, a change in the internal representation, e. g. an ordinary file copy, does not have to result in a change of the external representation (resp. the *Uniform Resource Identifiers* [13]). Furthermore, the introduction of a logical representation of the underlying data enables a comprehensive personalization and customization, because the document structure as well as its graphical presentation can easily be changed and adapted independently.

In the domain of XML-based Web applications, sets of XML documents are to be managed. If these sets are considered under a data-oriented perspective, their elements can be composed to a single, logical XML document. Such a composition assumes that the participating XML documents contain only small amounts of so-called *mixed content* [14] and that their element sequence is less important. The single, logical XML document establishes a basis for personalization and customization, because it enables an integrated view to the related data in its entirety. In particular, XML documents with similar *content models* [14] can be combined and accordingly integrated into the so-called *unified view.*

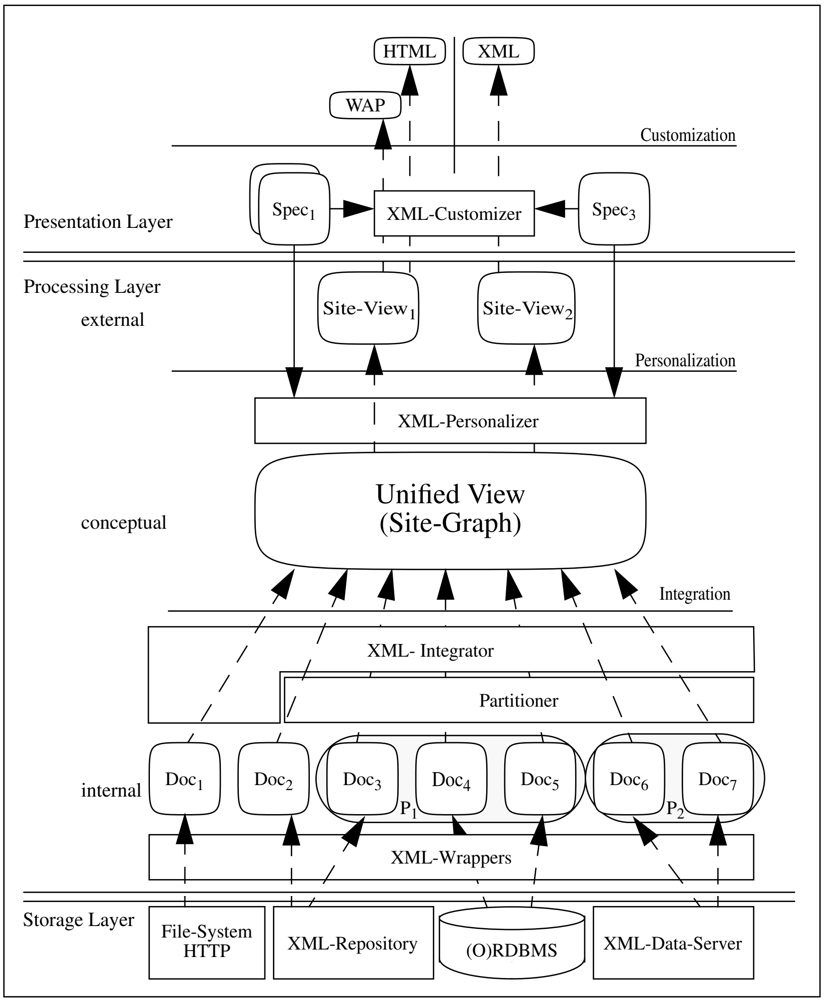

Fig. 1: Architectural Overview

In order to achieve the outlined properties of Web management, a system architecture for the support of comprehensive personalization is presented in the following section. The idea of building a unified view to sets of XML documents is discussed subsequently.

4. Architectural Overview

Our architecture is influenced by the mediator technology introduced by Gio Wiederhold [15]. Therefore, the above mentioned unified view corresponds to a mediated view and the component providing it is located in the center of the architecture. Based on the unified view, personalized views could declaratively be specified by a user as well as by the system. A system-driven personalization is helpful in the case of providing a default site structure or a default graphical data presentation and to facilitate the support of promotional requirements.

As illustrated in Fig. 1, the system consists of three major layers. At the back-end, the data sources are managed by the storage layer which could be implemented by one or multiple (distributed) data management systems. At the front-end, the presentation layer takes care of the device-dependent representation of the documents. For this purpose, it is possible to transform so-called site views into appropriate external representations. This transformation can also take place in the client or a third server component which does not have to be under control of the same system, for example to support more anonymity.

The heart of our architecture is the middle part. The processing layer is motivated by a common three-tier model of database design and realizes the different necessary steps of abstraction or processing, respectively: separation of physical and logical data representation, generation of the unified view and support of defining external views to enable data filtering, and restructuring. For this reason, the processing layer is divided into three sublayers.

At the internal layer, the different data sources are represented by XML documents whose fragments are made available by wrappers accessing different data sources. Since the data sources are conceptually heterogeneous, a single and common document-based XML model is the precondition of separating physical and logical aspects. In order to completely separate physical and logical data representation, a further step is performed at the conceptual layer: the XML documents are integrated into the unified view.

Unified view composition proceeds in multiple steps. The *partitioner* searches for similarities in the content models of the documents and builds partitions with sets of similarly structured documents (classification). Hence, documents belonging to different partitions can be distinguished by their content models. In order to combine these partitions, the *XML-integrator* performs two separate integration steps. All elements of a partition are grouped, before the resulting units are combined to the unified view. Since the unified view is independent from physical data structures and represents the underlying document basis in its entirety, comprehensive query support becomes possible.

At the external layer, *site views* of the whole document basis can be specified in a declarative way. Furthermore, specification of both data filtering and restructuring becomes feasible to derive personalized views. The *XML-personalizer* is responsible for maintaining and providing these views.

5. Unified View

The unified view is built by exclusively considering the underlying XML documents. The documents, respectively the contained elements, do not have to be connected or linked directly. In general, they are completely independent, but certain similarities can be found and exploited. Therefore, similarly structured documents are composed and integrated into the unified view. This section outlines the building process by describing the way from the document basis to the unified view.

5.1 Document Basis

The document basis consists exclusively of XML documents derived from the different data sources. The XML documents can be classified by functional criteria. In our system context, we have identified five different functional classes:

- *operational,*
- *structural,*
- *navigational,*
- *functional (or logical),*
- *derivational.*

The first class contains the actual operational data (e. g., documents containing bookmark lists or publication lists). The structural documents indirectly control the mechanism of building the unified view. For this reason, structural documents only contain nested placeholders (empty elements) in order to skeletally define the overall structure. Explicit links [1] between documents are found in navigational documents representing dependencies between documents which can be navigated or used to combine or aggregate them [4]. Functional documents specify simple application logic, e. g., defining dynamic content by using SOAP [16]. In order to support business rules, derivational documents are needed which contain data about user behavior and preferences. These data can be exploited by the personalization process to meet promotional demands.

5.2 Building Process

The separation of partitioning and integrating underlines the characteristics of the document basis and the building process of the unified view. As previously mentioned, the XML document basis is considered under a data-oriented perspective. Hence, it is feasible to find similarities between different operational documents and to build partitions by analyzing their content model (e. g., publication lists, RDF data [17], bookmark lists [18]). A DTD, if given, is analyzed, and the element names and their related namespaces are evaluated [19]. Documents with similar content models are grouped and composed. A given document can only participate in a single partition. After combining the operational documents of each partition, they are mapped into the unified view in such a way that they are plugged into the skeleton defined by the structural documents. These documents themselves build their own partition and are also grouped and composed. Up to now, only structural properties are considered. Similarities between elements are related to similarities in the underlying content models (i. e., identity in DTDs, namespaces, or element names), but semantic similarities among elements of different content models are not taken into account, yet.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PUBLICATIONS
  SYSTEM "file:///xml/dtd/pubs.dtd">
<PUBLICATIONS>
 <PUBLICATION YEAR="2000">
  <TITLE>t1</TITLE>
  <AUTHOR>a1<AUTHOR>
  <AUTHOR>a2<AUTHOR>
  <CONFERENCE>...</CONFERENCE>
  <ABSTRACT HREF="http://external.net/rs1"/>
  <DOWNLOAD HREF="http://external.net/rs2"/>
 </PUBLICATION>
<PUBLICATIONS>
```

Dokument 1

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PUBLICATIONS
  SYSTEM "file:///xml/dtd/pubs.dtd">
<PUBLICATIONS>
 <PUBLICATION YEAR="2000">
  <TITLE>t2</TITLE>
  <AUTHOR>a1<AUTHOR>
  <CONFERENCE>...</CONFERENCE>
  <ABSTRACT HREF="http://external.net/rs3"/>
  <DOWNLOAD HREF="http://external.net/rs4"/>
 </PUBLICATION>
<PUBLICATIONS>
```

Dokument 2

Fig. 2: Example Operational Documents

As already mentioned, navigational documents define relationships between elements. These relationships can be used for navigation or for restructuring data. In the latter case, elements and their corresponding *element tree fragment* [14] (resp. subtree) could be combined and integrated into the unified view within a further step.

Navigational documents are conceptually capable to cope with *outbound*, *inbound*, *third-party links* or *linkbases* [1]. All link types can simply be integrated into the unified view concept, because all of them can be transformed into semantically equivalent outbound links. Functional and derivational documents behave like operational documents, but they are expanded before their integration. An important design goal of our integration algorithm is that it should be generic, as far as possible.

5.3 Example of Building a Unified View

In order to illustrate the process of building the unified view, two operational documents describing publications are presented in Fig. 2. These documents do not contain mixed content and their content model is specified by an externally stored DTD. Because of their similar content models, they are assigned to the same partition and are grouped as shown in Fig. 3. In a similar way, other sets of documents can be handled: conference details, author information, staff members, or job offers.

As previously mentioned, a partition of structural documents (Fig. 4) is also built. The elements of this partition define skeletally the structure of the unified view. Together with the grouped documents in the related partitions, they are composed to the unified view (Fig. 5).

```
<PUBLICATIONS>
 <PUBLICATION YEAR="2000">
  <TITLE>t1</TITLE>
  <AUTHOR>a1<AUTHOR>
  <AUTHOR>a2<AUTHOR>
  <CONFERENCE>...</CONFERENCE>
  <ABSTRACT HREF="http://external.net/rs1"/>
  <DOWNLOAD HREF="http://external.net/rs2"/>
 </PUBLICATION>
 <PUBLICATION YEAR="2000">
  <TITLE>t2</TITLE>
  <AUTHOR>a1<AUTHOR>
  <CONFERENCE>...</CONFERENCE>
  <ABSTRACT HREF="http://external.net/rs3"/>
  <DOWNLOAD HREF="http://external.net/rs4"/>
 </PUBLICATION>
<PUBLICATIONS>
```

Fig. 3: Grouped Partition of Document 1 and 2

5.3.1 Exploiting Namespaces

Documents which are not of the the same document type, i. e., their root elements are not of the same type, can also be partitioned by considering the namespace used. Assume, a document contains data about publications in a form like Document 1, but without the <PUBLICATIONS> element. With a proper namespace specification, such a document can also be assigned to the same partition as document 1 and 2, provided that all three documents are defined in the same namespace context. If a namespace definition is not given, the element type is considered.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE SITEGRAPH
  SYSTEM "file:///xml/dtd/sitestructure.dtd">
<SITEGRAPH>
 <STAFF>
  <MEMBER/>
 </STAFF>
 <PUBLICATIONS/>
 <JOBS/>
 <PROJECTS/>
</SITEGRAPH>
```

Dokument 3

Fig. 4: Structural Document

5.3.2 About Semantics

The identification of semantic relationships, e. g. between authors and staff members, is not covered by the integration algorithm. This information has to be explicitly specified by providing navigational documents, e. g. deploying XLink techniques. Information about semantic similarities can be manually supplied by users or automatically derived from other processes based on special algorithms using Web mining, AI methods, CBR techniques, etc.

5.3.3 Personalization

Because a unified view of the entire document basis is given and information about semantic similarities between elements are made available to users, comprehensive personalization becomes feasible. Filtering data, e. g. selecting only publications of an individual author, choosing a special order for distinct elements, or restructuring the unified view in such a way that author and staff member information are combined, becomes conceivable (Fig. 6).

6. Summary and Future Work

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SITEGRAPH>
 <STAFF>
  <MEMBER/>
 </STAFF>
 <PUBLICATIONS>
  <PUBLICATION YEAR="2000">
   <TITLE>t1</TITLE>
   <AUTHOR>a1<AUTHOR>
   <AUTHOR>a2<AUTHOR>
   <CONFERENCE>...</CONFERENCE>
   <ABSTRACT HREF="http://external.net/rs1"/>
   <DOWNLOAD HREF="http://external.net/rs2"/>
  </PUBLICATION>
  <PUBLICATION YEAR="2000">
   <TITLE>t2</TITLE>
   <AUTHOR>a1<AUTHOR>
   <CONFERENCE>...</CONFERENCE>
   <ABSTRACT HREF="http://external.net/rs3"/>
   <DOWNLOAD HREF="http://external.net/rs4"/>
  </PUBLICATION>
 <PUBLICATIONS>
 <JOBS/>
 <PROJECTS/>
</SITEGRAPH>
```

Fig. 5: Providing a Unified View to Document 1, 2, and 3

In this paper, we have motivated an XML-based architecture for Web management supporting comprehensive personalization, i. e. data filtering, restructuring, and altering graphical presentations. For this purpose, we have presented the idea of a unified view to sets of XML documents. In particular, such a unified view facilitates querying the entire document basis. The presented approach addresses the integration of similarly structured documents and uses specially marked documents to build a unified view to the underlying data basis in a generic way. It describes a new combination of XML, declarative Web management, data integration, and mediation techniques for the management of data-intensive Web sites.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SITEGRAPH>
 <STAFF>
  <MEMBER>
   <NAME>a1</NAME>
   <PUBLICATIONS>
    <PUBLICATION YEAR="2000">
     <TITLE>t1</TITLE>
     <TITLE>t2</TITLE>
   </PUBLICATION>
  </MEMBER>
  ...
 </STAFF>
 <PUBLICATIONS>
  ...
 </PUBLICATIONS>
 <JOBS/>
 <PROJECTS/>
</SITEGRAPH>
```

Fig. 6: An External View to Document 1, 2, and 3

The possible application domains we are targeting are E-Commerce, WIS, WBT or portal systems, which can all benefit from a personalized environment. At the moment, however, there are many open problems. Change management, document buffering, generic data integration, and transactional aspects refer to only some of these problems. Other problems are related to the maintenance of the unified view. Materialization of the unified view, similar to the warehousing approach in [9], has to be compared with the provision of dynamic views based on query rewrite and mediation techniques. Finally, the specification methods concerning data filtering and restructuring for personalization purpose have to be developed.

References

1 XLink: Extensible Markup Language 1.0 (Second Edition). W3C Recommendation, http://www.w3.org/TR/2000/REC-xml-20001006, 2000

2 S. Abiteboul, P. Buneman, D. Suciu: Data on the Web - From Relations to Semistructured Data and XML. Morgan Kaufmann Publishers, San Francisco, California, 2000

3 D. Suciu: Semistructured Data and XML. In *5th International Conference of Foundations of Data Organization (FODO'98)*, Kobe, Japan, 1998

4 XML Query Requirements, W3C Working Draft, http://www.w3.org/TR/xmlquery-req, 2000

5 D. Florescu, A. Y. Levy, D. Suciu, K. Yagoub: Optimization of Run-time Management of Data Intensive Web-sites. In *Proceedings of 25th International Conference on Very Large Data Bases (VLDB '99),* Edinburgh, Scotland, UK, 1999, 627-638

6 D. Florescu, A. Levy, A. Mendelzon: Database Techniques for the World-Wide Web: A Survey. In *ACM SIGMOD Record 27:3*, 1998, 59-74

7 S. Ceri, P. Fraternali, S. Paraboschi: Data-Driven, One-to-One Web-Site Generation for Data-Intensive Applications. In *Proceedings of 25th International Conference on Very Large Data Base (VLDB '99),* Edinburgh, Scotland, UK, 1999, 615-626

8 P. Fraternali: Tools and Approaches for Developing Data-Intensive Web Applications: A Survey. In *ACM Computing Surveys Volume 31:3*, 1999, 227-263

9 M. Fernández, D. Florescu, A. Levy, D. Suciu: Declarative specification of Web sites with Strudel. In *The VLDB Journal Volume 9:1,* Springer, 2000, 38-55,

10 C. K. Baru, A. Gupta, B. Ludäscher, R. Marciano, Y. Papakonstantinou, P. Velikhov, V. Chu: XML-Based Information Mediation with MIX. In *ACM SIGMOD Conference*, Philadelphia, Pennsylvania, USA, 1999, 597-599

11 B. Ludäscher, Y. Papakonstantinou, P. Velikhov: Navigation-Driven Evaluation of Virtual Mediated Views. In *Proceedings of 6th International Conference on Extending Database Technology (EDBT '99)*, Konstanz, Germany, 2000, 150-165

12 C. R. Anderson, A. Y. Levy, D. S. Weld: Declarative Web-site Management with Tiramisu. In *ACM SIGMOD Workshop on The Web and Databases (WebDB'99)*, Philadelphia, Pennsylvania, USA, 1999, 19-24

13 T. Berners-Lee, R. Fielding, U. C. Irvine, L. Masinter: Uniform Resource Identifiers (URI): Generic Syntax. Network Working Group, Request for Comments 2396, 1998

14 XML: Extensible Markup Language 1.0 (Second Edition). W3C Recommendation, http://www.w3.org/TR/2000/REC-xml-20001006, 2000

15 G. Wiederhold: Mediators in the Architecture of Future Information Systems. In *IEEE Computer Magazine Volume 25:3*, 1992, 38-49

16 SOAP: Simple Object Access Protocol 1.1. W3C Note, http://www.w3.org/TR/SOAP/, 2000

17 RDF: Resource Description Framework Model and Syntax Specification. W3C Recommendation, http://www.w3.org/TR/REC-rdf-syntax/, 1999

18 F. L. Drake, Jr et.al.: The XML Bookmark Exchange Language. Corporation for National Research Initiatives (CNRI), Reston, USA, http://www-texdev.mpce.mq.edu.au/DRAKE/Doc/xbel/xbel.html

19 Namespaces in XML, W3C Recommendation, http://www.w3.org/TR/REC-xml-names/, 1999