WebCon: Design and Modeling of Database Driven Hypertext Applications

U. Sommer, P. Zoller
FORWISS (Bavarian Research Center for Knowledge-Based Systems)
Orleansstraße 34, D-81667 Munich, Germany
[sommer,zoller]@forwiss.de

Abstract

As the Web has evolved from its origins as a simple delivery mechanism to a platform for complex applications, the connection of databases and the Web has become more and more important. The design process for a database driven hypertext application is usually divided into modeling of the database and modeling of the hypertext structure. In most of these applications information contents is stored within the database, while the hypertext structure is located outside the database system in flat files. This leads to inconsistencies, labor-consuming maintenance and unsatisfying scalability. In this paper we present the WebCon approach which uses a traditional data model for storing both the application domain and the hypertext structure. A special repository is introduced for handling meta data which describes the hypertext structure of an application. We implemented a set of universal templates for creating HTML pages out of database contents based on the WebCon repository. The main benefits of WebCon are scalability, high consistency, easy maintenance, and rapid prototyping.

1. Introduction

With its rapid growth and acceptance, the World Wide Web has become an important media to provide information and services for the public. By employing well accepted standards such as HTML, HTTP or TCP/IP, the Web provides a platform-independent basis for world wide distribution of information. However, the Web does not support efficient and reliable administration of large amounts of data. Considering the Web’s immense growth, it is no longer enough only to offer pure navigation capabilities to the Web. Easy administration and efficient information retrieval play the key role in providing high-quality services to the Web. Apart from the development of special data models and query languages for hypertext applications [11] [14], an even more interesting approach is to store the information contents within database systems. This technique is particularly suitable to support the administration of large volumes of structured data that change quickly over the time.

The hypertext structure of a database application is usually stored in flat files outside the database system. As a consequence, these applications suffer from bad scalability, labor-consuming maintenance and inconsistencies between the hypertext structure and the database structure.

In this paper, we present our approach on how to solve this problem by handling both information contents and hypertext structure within a database system. Our solution is a program package named WebCon, which allows modeling of the hypertext structure within a relational database and thus ensures consistency and efficient administration of the whole application. Relational databases are capable of storing data in a highly structured way. When attempting to generate hypertext pages from database records, the system catalog of the database can provide information on relationships between database tables on a meta-level. Beside this meta information which is stored within the database’s data dictionary, WebCon provides additional meta data describing the hypertext structure. The WebCon repository can be maintained on-line with several tools developed during the project.

This paper focuses on the WebCon repository meta schema used to define layout and content of the HTML pages created dynamically. After describing different architectural and logical aspects of connecting databases to the Web in Section 2, Section 3 presents in detail the development of database driven hypertext applications and the WebCon meta schema. Section 4 takes a closer look at WebCon implementation issues and architecture, followed by a brief overview of related work in Section 5. The paper ends with conclusions.

2. Databases and World Wide Web

Several different techniques and architectures have been developed for connecting databases to the Web. An important issue in all these techniques is to bridge the gap
between HTML (the standard markup language) and SQL (the standard query language used to access relational database systems). Normally a Web page accessed using an URL is static and stored in the server location. In order to implement Web applications that access a DBS, Web pages have to be generated dynamically as result of the user input and the database access.

Regardless of implementation details, the connection to the underlying database system can be seen from two different points of view: the technical view and the logical view.

2.1. Technical View

The technical view concentrates on different technical aspects of connecting databases to the Web, each of them having its strengths and weaknesses with respect to criteria of the individual application domain. However, since this is not the main issue of this paper, only a few basic considerations will be given without going further into details.

One of the major differences between the various database-Web architectures is the logical separation of different architectural layers (application interface, application logic, data storage, etc.) and their distribution on different physical units (client, server, application server, etc.) [18]. The connection between Web and database can be established either on the client side (e.g. with the use of Java or ActiveX), or on the server side (e.g. with the use of the Web server Application Programming Interface (API) or the Common Gateway Interface (CGI)).

The client side connection allows direct communication between Web browser and database system. The Web browser loads and executes additional program code available as Java Applets or ActiveX Controls. This approach offers full transaction management and concurrency control.

In case of a Web server based connection, the communication between Web browser and database server is managed exclusively by the Web server. A middleware located between Web server and database server extracts user inputs from the HTML forms and queries the DBS. After the query has been executed, the middleware turns the query results into a dynamically generated HTML page and sends it back to the Web server which delivers it to the client. The Common Gateway Interface is a standard for external programs to interface with Web servers and perhaps the best known approach for adding services to the Web. WebCon is based on this technique.

2.2. Logical View

In contrast to the technical view, the logical view deals with the information modeling, in particular content modeling, the hypertext structure and the hypertext presentation. The modeling of database applications in the Web can roughly be divided into two different areas: on the one hand there is a need for modeling the underlying database and on the other hand there is the aspect of modeling the hypertext interface itself. Therefore, the development life cycle of database driven hypertext applications is a combination of traditional database design methods and hypermedia design and analysis. The following design approaches can be applied:

- Bottom-Up: The database is already designed, the hypertext design is based on the conceptual database model.
- Top-Down: The Web application is designed first, the database is designed to support it.
- Mixture: Parallel and iterative conceptual modeling of hypertext application and database structure.

Various steps in the design process of hypermedia applications depend on whether there is an existing database to be coupled to the Web or whether the Web database has to be developed from scratch. In the second case the information contained in the database usually shows less structure than in traditional information systems, and contains multi-media objects and free-text, as well as hypertext links.

There are various papers referring to several hypermedia design methodologies. Some of the well known representatives are the Hypertext Design Model (HDM) [7], the Relationship Management Methodology (RMM) [10] and the Object-Oriented Hypermedia Design Method (OOHDM) [20]. All these methodologies are based on well known semantic data models such as the ER-Model [4] or OMT-Model [19]. These models are usually extended by means of additional components allowing the modeling of hypertext structures and navigation. In contrast to classical database design, hypertext design methodologies cover the following additional design steps:
• Clustering of Information: clear and sensible structuring of entities and attributes in meaningful pieces regarding the later hypertext network.
• Navigational Design: development of a consistent navigational structure within the hypertext network.
• Layout Design: definition of the user interface (e.g. color, background, header).

In contrast to database design, an essential problem regarding database driven hypermedia design is the absence of a logical data model. The usual techniques for connecting databases to the Web allow only to model and store the application domain within the database system. Because of the lack of an appropriate logical hypermedia data model, hypertext and hyperlink structures are usually stored outside the database in flat files. Since the application domain is already modeled completely within the database, it would be desirable to model the hypermedia context within the same database too. An interesting question is in which way traditional database models (e.g. relational or object oriented) are suitable for this task. One of the main challenges of the WebCon project was to develop an appropriate meta model for hypertext structures based on the relational data model.

3. The WebCon Approach

The core feature of WebCon is its ability to model hypertext structures within a relational database. Structuring and storing the entire hypertext information within a database guarantees scalability, easy maintenance and consistency of the application. The program code isn’t affected by changes in the database schema or hypertext structure because all changes only affect the meta data inside the repository. This supports rapid prototyping, which is an important factor considering the highly iterative process of designing hypertext applications [15].

Another important feature of WebCon is the presentation of dynamic, query dependent information in a consistent hypermedia format. The advantage of consistency of the hypertext interface is considered as one of the most important evaluation criteria for hypertext applications [6]. Similar concepts should be treated in a similar fashion, different concepts should be treated differently. This can be guaranteed by a logical design of the hypertext interface.

The following section proposes the WebCon solution for building a large class of Web applications using DBS.

3.1. The WebCon Design Process

The design of almost every database consists of two steps: the conceptual design and the logical design. In contrast to this, there are various different approaches for the modeling of hypertext applications:

1. Meta Schema Modeling: The modeling of hypertext structures and navigation is transferred to a specific meta schema in a traditional database model. This alternative is used within the WebCon project [22].
2. New Logical Hypertext Model: A new logical model is developed to meet the requirements of hypertext (e.g. Araneus [2]).
3. Modeling Within Flat Files: The entire hypertext application is hardcoded within HTML templates in flat files to be parsed by the Web gateway [9].

Figure 2 shows these three design approaches, where each variant is clarified by numbered lines. The WebCon design process is drawn with the lines labelled with number one.

3.2. The WebCon Meta Model

The WebCon Meta Model has been developed especially for the design of database driven hypertext interfaces. Meta data about structure, navigation and layout of each HTML page is stored within a relational database. Figure 3 shows the core of the WebCon meta model using the notation of OMT [19].

In order to reduce the amount of information presented on one HTML page, the number of hits is usually decreased by offering some kind of search form to the user. Thus the process of navigating through a database can be divided into three steps:

• the user gets a form to enter his search criteria (the Query Page);
• the result of the user’s query is displayed in short form as a list (the Results Page);
• after selecting a hit from the Results Page, the user gets detailed information about that object including links to related areas within the database (the Details Page);
According to this concept, WebCon introduces three different kinds of HTML pages for navigating through a database: Query Page, Results Page, and Details Page. Each Query Page is associated with a corresponding Results Page and Details Page. WebCon also provides a fourth kind of page type (Input Page) for populating a database table. Due to space limitations, this page type cannot be discussed in this paper.

Numerous parts of information, called page attributes, are clustered on each HTML page (see Figure 3). Depending on the kind of HTML page (Query, Results, Details, or Input Page), different meta types can be assigned to the page attributes. WebCon provides predefined meta types which determine the presentation of a given page attribute on the HTML page (see Table 1 and Table 2).

The join meta type details(inPlace) allows to show a set of attributes from the joined table directly on the current HTML page, where the presentation of these attributes is again determined by their individual meta types. Another purpose of join meta types is to allow navigation from the Details Page to related Results or Details Pages using any of the following access structures:

- Reference: hyperlink to another Details Page;
- Conditional Index: list of hyperlinks to Results Pages of objects related to the current Details Page;

In addition to its page attributes and corresponding meta types, each HTML page has also its own layout information (header, footer, background) being stored within the database. Therefore WebCon Meta Model guarantees identical representation of conceptually identical elements.

---

**Table 1: Meta types for Query Pages**

<table>
<thead>
<tr>
<th>Meta type</th>
<th>HTML representation</th>
<th>Available for DB types</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>textual input line</td>
<td>CHAR(*), CHAR</td>
</tr>
<tr>
<td>num input</td>
<td>numeric input line with three buttons for comparison operator (‘=’, ‘&lt;’, ‘&gt;’)</td>
<td>INTEGER</td>
</tr>
<tr>
<td>float</td>
<td>textual input line with three buttons for comparison operator (‘=’, ‘&lt;’, ‘&gt;’)</td>
<td>FLOAT, DOUBLE</td>
</tr>
</tbody>
</table>

**Table 2: Meta types for Results/Details Pages**

<table>
<thead>
<tr>
<th>Meta type</th>
<th>HTML representation</th>
<th>Available for DB types</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>plain text</td>
<td>any type</td>
</tr>
<tr>
<td>email</td>
<td>link of type email</td>
<td>CHAR(*)</td>
</tr>
<tr>
<td>url</td>
<td>link of type url</td>
<td>CHAR(*)</td>
</tr>
<tr>
<td>bool</td>
<td>yes or no</td>
<td>BOOL</td>
</tr>
<tr>
<td>download</td>
<td>file download</td>
<td>CHAR(*)</td>
</tr>
<tr>
<td>image</td>
<td>inline image</td>
<td>BLOB</td>
</tr>
</tbody>
</table>

**Figure 3: WebCon meta model**

It is possible to display related information contained in different tables by defining joins for the Details Page. For this purpose predefined join meta types can be selected and stored in the meta schema (see Table 3). The join meta type details(inPlace) allows to show a set of attributes from the joined table directly on the current HTML page, where the presentation of these attributes is again determined by their individual meta types. Another purpose of join meta types is to allow navigation from the Details Page to related Results or Details Pages using any of the following access structures:

- Reference: hyperlink to another Details Page;
- Conditional Index: list of hyperlinks to Results Pages of objects related to the current Details Page;

In addition to its page attributes and corresponding meta types, each HTML page has also its own layout information (header, footer, background) being stored within the database. Therefore WebCon Meta Model guarantees identical representation of conceptually identical elements.
### 3.3. Mapping applications to the WebCon Meta Model

Simply mapping complete database relations to HTML pages is not appropriate. Depending on the application, it may be desirable to display only selected attributes in a certain order or to include additionally some of the attributes from related tables connected via foreign keys. Each HTML page belongs to one relation, but it is important to allow different versions of HTML pages per relation in order to display a table’s content in different ways according to different purposes. For this reason, each HTML page can consist of:

- all attributes of a given relation,
- a defined set of attributes from the relation,
- a defined set of attributes from other database tables connected via foreign keys.

Access structures are implemented by foreign keys between tables and defined via join meta types. The database concept of referential integrity guarantees the absence of dangling links. Information about the access structures is stored within separate meta tables used to define join meta types. The result of a join is displayed according to the selected kind of access structure, defined by the meta type of the join.

Table 4 shows the information necessary in order to define the content of a HTML page.

<table>
<thead>
<tr>
<th>Meta type</th>
<th>HTML representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>links(inPlace)</td>
<td>one link for each result of a join</td>
</tr>
<tr>
<td>links(extraPage)</td>
<td>one link to a Results Page containing a reference list with the join results</td>
</tr>
<tr>
<td>details(inPlace)</td>
<td>set of values of the attributes defined for each result</td>
</tr>
<tr>
<td>details(extraPage)</td>
<td>one link to the Details Page of the join result</td>
</tr>
</tbody>
</table>

### 4. WebCon Implementation

The current implementation uses a commercial relational database system (TransBase from Transaction Software) and a CGI based gateway (a part of the WebCon program package) to connect it to the Web server. The gateway is written in the program language C and is implemented on a Sun workstation.
The WebCon gateway has been built as a parser for HTML templates similar to commercial products like Cold Fusion [1], Net Dynamics [8] or the Informix Web Data-blade [9]. As already mentioned, an important issue of the WebCon project was the development of browser based maintenance tools. These tools allow both an easy mapping of applications to the WebCon Repository and an automatic generation of HTML pages out of it.

For mapping hypertext models to the WebCon Repository, several administration templates are provided which make extensive use of the database’s data dictionary (see Figure 4). Thus WebCon offers a user friendly browser based interface for the definition of the WebCon meta data described in Section 3.3. Input fields for its position and label and a popup menu for its meta type are assigned to each attribute of a given table (see Figure 5). The popup menu for the meta type is created dynamically depending on the database type of the given attribute.

The WebCon package is successfully applied in the following database driven hypertext applications from different domains:
- abayfor-online, an information system for the organization of bavarian research centers.
  URL: http://www.abayfor.de
- M-Line, a german national information system containing data about new materials, about research institutions and companies producing or working with them and about working techniques applicable to these materials.
  URL: http://format.mwn.de/
- TEX-Line, an information system for the european textile industry which is developed within the ITEX project funded by the European Commission.
  URL: http://texline.forwiss.tu-muenchen.de/
- Line42, an on-line datawarehouse bibliography.
  URL: http://line42.forwiss.tu-muenchen.de/
- TTM-Line, an information system containing mainly german research papers in the field of traffic and transport management which are stored according to a new conceptual model for this research area. Different search approaches beyond common literature search are offered.
  URL: http://ttmline.forwiss.tu-muenchen.de/
Experiences from existing applications have shown that this technique is not performance critical when applied to small or medium size databases.

5. Related Work

In recent years various research work has addressed the problem of combining database technology with the World Wide Web. Some approaches deal with the creation of design methodologies, models and query languages for semistructured data in the Web. Two well known examples are WebSQL [14] and W3QL [11]. WebSQL relies on a relational schema for the Web consisting of two tables: documents and anchors. The document’s internal structure is only roughly described by a predefined number of attributes. In the W3QL approach the Web is viewed as a marked graph using Web pages as nodes and hyperlinks as edges. Both approaches are mainly based on information retrieval techniques. In contrast to the WebCon approach, the basic models used in both projects aren’t suitable for highly structured data generated out of a database system.

Other approaches, like the TSIMMIS System [3], mainly address the topic of integrating and querying heterogeneous, semistructured data sources within the Internet. TSIMMIS provides a query language (LOREL) and an object repository. The focus lies on integration and query processing for partly inconsistent, incomplete information without a uniform structure. In contrast to this, WebCon relies on highly structured information stored in a database system.

The Araneus project [2] introduces a page oriented data model for the Web. The main focus is not on connecting sources with structured information to the Web, but on structuring and querying hypertext sources. The information about the structure of such a source is stored within a logical data model. Within this approach, data from the Web can be extracted, stored inside a database, restructured and then converted to HTML pages again. WebCon also sees the Web from a logical point of view, but stores all the information about the hypertext structure within a meta model.

There are various different products and research approaches for connecting databases to the Web, dealing mainly with the technical aspects of the connection. Examples are DB2WW [16] from IBM, Oracle’s Web Application Server [17], or the Web DataBlade Module from Informix [9]. However, none of these products allows modeling and storage of Web related information within the database system itself like WebCon does.

6. Conclusions

In this paper we have shown an approach to the automatic generation of hypertext from relational databases using the WebCon repository. The main benefits of WebCon are the rapid and simple creation of hypertext applications, its easy maintenance via any web browser and the dynamic concept making it rather insensitive to changes within the database schema. Structuring and storing the entire hypertext information within a database guarantees scalability, easy maintenance and consistency. The WebCon project meets all these requirements using existing design methodologies as a basis.

WebCon introduces a hypertext meta model stored within the WebCon Repository, which allows a dynamic generation of four different kinds of HTML pages (Query, Results, Details, Input) based on underlying application data and repository definitions. A concrete implementation provides both a user friendly interface for mapping the application data to the WebCon Repository and a set of universal templates for creating HTML pages of any of the four types on the fly. WebCon has been applied to several database driven hypertext applications from different domains.
References


[18] H. Roderus: Auf die Netze, fertig, los ... Wie geeignet sind bestehende Anwendungen für die Migration ins World Wide Web?. Online ’97, Congress VI, C625.


