

WEBCON: A TOOLKIT FOR AN AUTOMATIC, DATA DICTIONARY BASED CONNECTION OF DATABASES TO THE WWW

Peter Zoller, Ulrike Sommer
Bavarian Research Centre for Knowledge-Based Systems (FORWISS)
Orleansstraße 34, D-83667 Munich, Germany
Email: {zoller,sommer}@forwiss.tu-muenchen.de
fax: +49-89-48095-203

KEYWORDS

database to WWW connection, dynamic page generation, automatic page generation, automatic query generation, relational databases

ABSTRACT

Despite the availability of various tools for creating dynamic HTML pages, the development of database applications in the World Wide Web is still relatively costly. Many dynamic HTML pages are needed for making a database comprehensively accessible via the web. This paper presents a toolkit for connecting existing relational databases to the web without having to write code in HTML or any other language. It makes extensive use of the database's dictionary and of additional meta information, which can easily be defined with a web based maintenance tool. The meta schema used will be discussed in detail. It allows to control layout and content of the pages for querying and populating the database and to define arbitrary joins within the database. The corresponding queries are created automatically.

1 INTRODUCTION

With its rapid growth and acceptance, the World Wide Web has become an important medium for providing information and services to the public. Because of the web's incapability of efficiently maintaining large amounts of data and because of the existence of huge information sources in form of databases, a connection of these systems is desirable.

Several different techniques and architectures are offered for connecting databases to the web: the connection can be established on client side (e.g. with the use of Java or ActiveX), or on server side by using either the Application Programming Interface (API) of the webserver or the Common Gateway Interface (CGI).

Permission to make digital/hard copy of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or fee.

© 1998 ACM 0-89791-969-6/98/0002 3.50

The queries to be sent to the database and the rules for producing the HTML pages with the results can be defined as stored procedures [10], as templates which are parsed by the webgateway [7] or as a combination of queries specified in the URL and formatting rules specified as methods of the related objects [9].

But independent of architecture and technique used, a lot of work has to be done to develop database applications for the WWW. Static and dynamic HTML pages have to be created where the user can enter his search criteria, get a list of results, view a certain object of interest in detail or even insert entries into the database.

One popular way for creating dynamic web pages and offering database access is to enhance HTML by introducing new tags for handling variables, defining loops or dealing with SQL queries. Pages containing these new tags are called templates. A web gateway has to be provided for parsing these templates, dealing with the special tags and generating ordinary HTML.

There are many existing tools and products, often called *application generators*, which offer some kind of enhanced HTML and utilities for creating templates. Examples are Swoop[5], Cold Fusion[1], Net Dynamics[4] and SQL-Surfer[8].

Many of them have their own IDE with features like drag & drop or WYSIWIG editors, which ease the creation of templates. But nevertheless numerous templates for searching and populating a certain database have to be created and tested.

The WEBCON package already provides the templates necessary for querying and populating a database. It consists of

- a web gateway understanding more than 30 new tags
- five universal, configurable templates
- a tool for creating the necessary meta information in order to customize the templates.

It is different from other tools in two ways.

First, it is no general application generator, but a toolkit especially developed for connecting existing relational databases to the web (currently TransBase[11] from TransAction Software is supported).

Second, no templates for querying or populating the database have to be written by the administrator. Only some meta information for the five universal templates has

to be provided which can easily be specified via the web based maintenance tool.

The main features and advantages of WEBCON are:

- No code has to be written. A browser based maintenance tool is used for defining meta information for the automatic generation of HTML code.
- Insensitive to changes within the database schema. Since the maintenance tool makes extensive use of the data dictionary, this is a highly dynamic system which is rather insensitive to changes within the database schema. When adding or deleting attributes, relationships, or even whole relations, relevant meta information can be customized with the maintenance tool within minutes.
- Remote administration via any web browser is possible.
- Automatic query generation. Even very complex queries with many joins can easily be defined with the maintenance tool and are built automatically which would be rather difficult and time consuming if done manually.
- Easy maintenance. The WEBCON package eases the maintenance of the database connection to the web, because it consists of only five templates, no matter how complex the database schema may be.
- Full use of standard HTML 3.2 features. WEBCON fully uses HTML 3.2 features, like tables, inline images, hyperlinks, or email links, to display the contents of the database.

The remainder of this paper mainly focuses on the meta schema used to define layout and content of the dynamically created HTML pages, especially the pages for querying the database. It is structured as follows: Section 2 describes the web gateway used and the general WEBCON concept. Section 3 presents in detail the meta schema used for configuring the universal templates and Section 4 deals with related work. The paper ends with conclusions.

2. WEBCON ARCHITECTURE

2.1 Gateway

The connection of database and WWW is realized on server side by using the Common Gateway Interface of the web server [2]. This solution was chosen because of the flexibility and independence from the webserver used. For transmitting the query and creating HTML code, the technique of server parsed HTML is used. A grammar and the corresponding parser have been developed which offer the use of additional HTML tags for connecting to the database, executing queries, and defining variables or loops. The gateway parses the templates containing these new tags, transfers standard HTML tags and text to the web server without changes and executes the special tags. Each of these new tags has the following syntax:

```
<!-- WEBCON command_with_parameters -->
```

The new tags are defined as HTML comments in order to avoid conflicts with future tags.

If a command requires block structures (like *if* or *loop*), the end of a block is indicated by the tag

```
<!-- /WEBCON -->.
```

If variables are defined, their content can be retrieved using the `&` operator, for example `&(counter)` returns the value of the variable *counter*.

The results of a database query are returned within a loop. Each result tuple is stored in an array variable named *table*. Currently the gateway supports 31 special tags for connecting to a database, starting and ending transactions, defining arbitrary SQL statements, dealing with query results, creating and working with variables, and defining loops, blocks, and conditional statements. Some of the commands available can be seen in the sample template below:

```
<HEAD>
...
</HEAD>
<BODY>
<H1>List of suppliers</H1>

<TABLE BORDER=2>
<TH>supplier</TH>
<TH>phone</TH>
<TH>fax</TH>
<!-- WEBCON BT -->           <!-- begin transaction -->

<!-- WEBCON SQL select name,tel,fax from suppliers -->
<TR>
  <!-- WEBCON FOREACH table column -->
    <TD>&(column)</TD>
  <!-- /WEBCON -->
<!-- /WEBCON -->

<!-- WEBCON CT -->           <!-- commit transaction -->
</TABLE>
...
</BODY>
```

For security reasons, two gateways are used. The gateway for querying has only read permission on the database, whereas the gateway for administration and population uses a special account. While the first one is accessible by the public in most cases, the second one should be located in a protected area of the webserver.

2.2 Templates

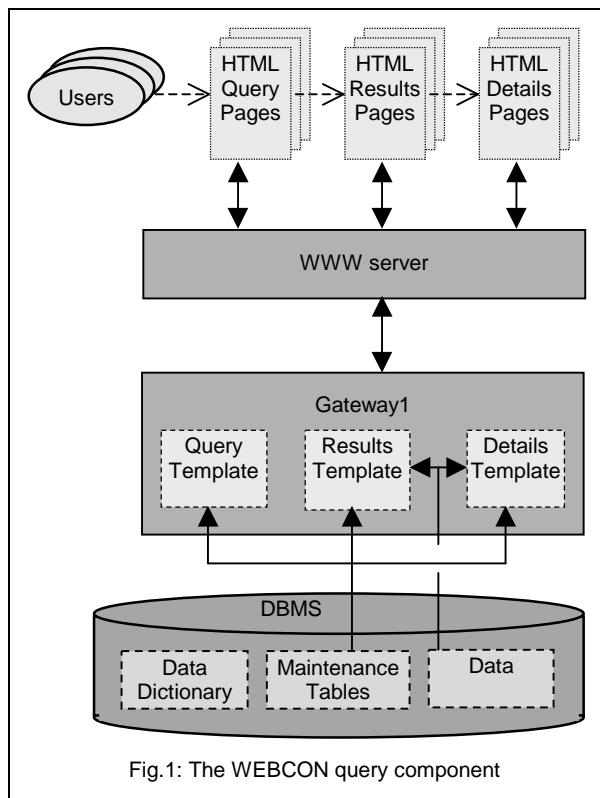
Usually, the process of querying a database can be divided into three parts. First a query page is presented where the user can specify his search criteria by typing in some text, choosing from some popup menus or selecting some options from checkboxes.

The query is then created according to the user's input from the query page, it is executed and the matching elements of the database are displayed in short form on the results page. In order to get detailed information about a certain result item of the query, in the third step the user selects the desired item and then gets all available information in the details page.

Conventionally realizing these three steps, HTML pages have to be written which contain input elements such as lines for text, radio buttons, or popup menus in order to gather the user's search criteria. Then code (for example as templates) has to be developed which transforms these user inputs into an SQL query for the database and displays its results in an HTML page. Finally, code for creating the details page has to be written, which displays a detailed view of the selected item.

All this work is done for only a single kind of search in the database. If another search (perhaps in different relations) is to be offered, everything has to be done again because now there are different attributes, input elements, queries, and presentations of results. Additionally, for a different schema all these steps have to be done again. These problems also arise when creating pages for populating a database.

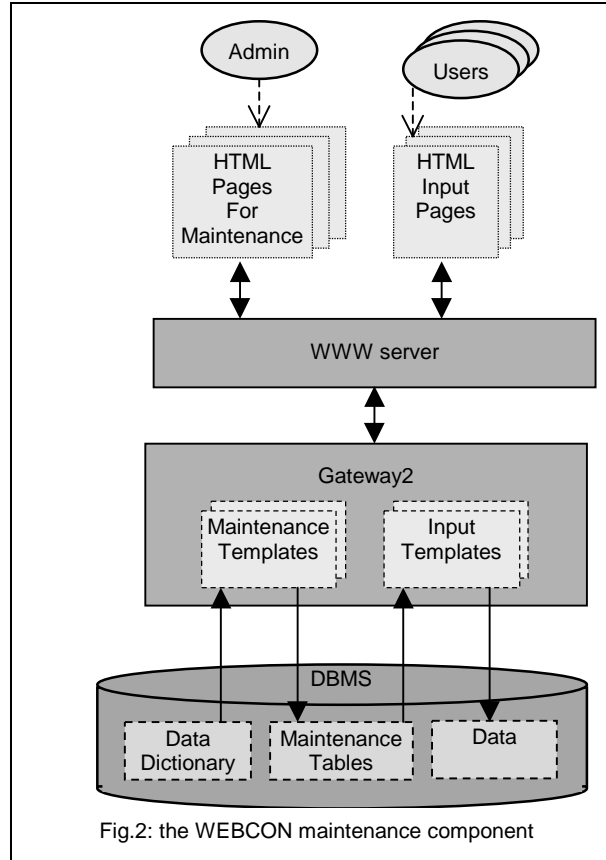
In order to avoid these efforts, WEBCON provides three generic templates for querying and two generic templates for populating a database. They are completely configurable by defining certain meta information located in the maintenance tables as shown in Figure 1.



The main idea is to map database types to WWW types. Each database attribute which is to be queried or shown is assigned a certain meta type. These meta types determine the HTML output for the values of this attribute.

In addition to this, arbitrary 1:1, 1:n or n:m joins for the details page as well as for the query and results page can be defined. The corresponding queries are automatically built and executed at runtime.

For creating and maintaining this meta information, a browser based maintenance tool is offered. It allows the administrator to configure the creation of HTML pages for both querying the database and inserting tuples into it without having to write code in HTML or any other language.



It makes extensive use of the database's data dictionary for many purposes: It displays all attributes of a given relation and offers a list of available meta types for each of them depending on its database type. It also detects foreign keys automatically and offers suitable meta types. When defining joins (see Section 3.1), the entered information about relations and attributes is checked against the data dictionary.

Since this tool can be run via any web browser, remote maintenance of the whole system is possible. The conceptual architecture can be seen in Figure 2.

3 WEBCON META SCHEMA

The WEBCON templates create HTML pages depending on the meta information they get from the maintenance tables.

For building a page displaying database contents (or displaying input forms for populating), the following information is necessary:

1. the name of the current relation
2. the attributes of this relation which have to be displayed
3. the order of the attributes to be displayed
4. the label for the content (or input formular) of each attribute
5. the way the attribute's content (or input formular) has to be displayed (the meta type)
6. the joins which have to be performed to get related information from other tables
7. the join result's attributes which have to be displayed
8. the way the contents of these join attributes have to be displayed (the meta type).

This information together with header, footer and a background image enables a completely automatic generation of HTML pages.

3.1 Principles

The WEBCON package is relation-oriented, in that each relation of the database schema which is intended to be searched or populated by the user gets its own query, results, details, and input page. Each page is created dynamically at runtime from one of the five templates in three steps. First header, footer and background are loaded, then attributes and their meta types are fetched and displayed, and finally defined joins are handled.

A join is defined by selecting the attribute to be joined in the current relation (*sourceAttr*), the corresponding attribute in the foreign relation (*targetAttr*), the name of the foreign relation (*targetRel*) and one or more attributes (*displayAttrs*) of the foreign relation which are to be displayed (or queried). If an n:m join is to be defined, also the name of the joinrelation (*joinRel*) has to be entered; the names of the corresponding attributes in this join relation are detected automatically from the data dictionary. For each attribute specified in *displayAttrs* a meta type can be selected which determines the kind of presentation of this attribute.

In order to enable different kinds of search on the same relation, several variant sets of query, results, and details pages (called *versions*) for the same table may be defined. For example, when querying a relation containing information about new materials, it could be useful to provide one query page for restricting the name, category or application area of a material and one for searching in physical or chemical parameters like density or melting temperature.

The whole meta information is kept within nine maintenance tables in the database system: One global table contains all headers, footers, and backgrounds. Each kind of page (query, results, details and input page) has one table containing a description of the attributes to be displayed and one table containing information about joins to be performed.

Keeping all meta information within the database (even though some parts, like header or footer, could have been

saved as files) ensures a maximum of consistency and security.

Due to space limitations, only the meta schema for the query and details templates is discussed in this paper.

3.2 Common Layout Information: Header, Footer and Background

Header and *footer* are strings of arbitrary length and content which are inserted at the top and end of the HTML page. Each page has its own header and footer which may contain ordinary HTML tags. In addition to this, an image used for the page background can be declared.

3.3 Meta Information for the Query Page

3.3.1 Attributes

Each relation attribute to be queried has an entry in the meta table for the query page. Besides the attribute's meta type, its label (the attribute's name is not always appropriate) and its position within the page are stored there. This meta table is looked up in the second step of creating the query page. For each entry belonging to the current relation (and version), a certain input field depending on the meta type is generated. Available meta types are:

Meta type	HTML representation	Available for DB types
char	Textual input line	CHAR(*), CHAR
num	Numeric input line with three buttons for selection of comparison operator ('=', '<=', '>=')	INTEGER
float	Textual input line with three buttons for selection of comparison operator ('=', '<=', '>=')	FLOAT, DOUBLE
popup	Popup menu	any type
multi(and)	Popup menu with multiple selections to be combined via <i>and</i>	any type
multi(or)	Popup menu with multiple selections to be combined via <i>or</i>	any type
bool	Radio buttons labeled <i>yes</i> , <i>no</i> and <i>don't care</i>	BOOL

If meta type *popup* is assigned to an attribute, a popup menu is created at runtime containing all different values of this attribute in the database, one of which can be selected by the user. This type must be chosen very carefully: first no more than about 25 or 30 different values for this attribute should exist, otherwise the popup menu would be too long to be useful. Second as the whole relation is scanned for creating this popup menu, very big relations can cause some delay when building the query page. In this case, meta type *char* would be the better choice.

For performance reasons, the contents of popup menus could have been stored somewhere in the maintenance

tables instead of calculating them at runtime. But this would have lowered the dynamics of the toolkit, caused more efforts for maintaining the system and created a source for possible configuration errors.

Similar to the previous type, for meta type *multi(and)* a popup menu is created at runtime, but here multiple options of this menu can be selected which are combined with a logical *and* operation. Type *multi(or)* is provided in order to connect the selections with a logical *or* operation.

The third option *dont't care* of meta type *bool* is necessary, because once a radio button has been selected by the user, it can't be unselected without choosing another option. So the user could only get all entries where the attribute of meta type *bool* is *true* or where it is *false*, unless a *dont't care* option is available.

3.3.2 Joins

It is often necessary to be able not only to restrict certain attributes of a relation, but also to restrict some value in a different table. One example could be a search for all customers who have made an order within the last month, where customers and orders are two different relations. For this purpose, joins for the query page can be defined by the administrator and are stored in a separate meta table. This meta table is searched for relevant entries as a third step of creating the query page.

The joins are defined as described in Section 3.1 and each join attribute to be displayed gets its own meta type determining its web representation. These meta types are identical to those mentioned above.

3.4 Meta Information for the Details Page

3.4.1 Attributes

Each relation attribute to be displayed in the details page has an entry in the corresponding meta table. This meta table is looked up after getting the header, footer and background as described in 3.2. The presentation of an attribute's value depends on its meta type:

Meta type	HTML representation	Available for DB types
normal	plain text	any types
email	link of type email	CHAR(*)
url	link of type url	CHAR(*)
bool	<i>yes</i> or <i>no</i>	BOOL
image	inline image	BLOB

If meta type *normal* is selected, the attribute's value is displayed as plain text. *Email* creates a mailto link labeled with the value of the attribute. By selecting it, a mail can be composed and sent to the specified address. Similar to this, meta type *url* creates a hyperlink with the value of this attribute as its URL. For an attribute with meta type *bool*, *Yes* or *No* is displayed depending on its value. If an attribute is assigned the meta type *image*, its content (which has to be a BLOB) is interpreted as an image in gif format

and an IMG-tag is created which displays this image in the HTML-page.

3.4.2 Joins

It's possible to display related information contained in different tables by defining joins for the details page. For example, if there are two relations *employees* and *projects*, when looking up a certain employee it would be useful to display also the projects the person is working in.

For this purpose arbitrary joins for the query page can be defined and stored in a separate meta table. The join is defined the same way as described in Section 3.1 except for an additional meta type which determines the kind of presentation of the join results:

Meta type	HTML representation
links(inPlace)	one link for each result of the join
links(extraPage)	one link to a results page containing the join results
details(inPlace)	set of values of the attributes defined in <i>displayAttrs</i> for each result
details(extraPage)	one link to the details page of the join result

If defined as *links(inPlace)*, the join is performed and for each result a link is created on the details page. It is labeled with the value of the first attribute specified in *displayAttrs* (see Section 3.1). Additional attributes contained in this field are printed as plain text behind the link. When following one of these links, a details page of the selected item is displayed.

For meta type *links(extraPage)*, the join itself is not performed, but a link is created, the selection of which leads to a page where the join is calculated, executed, and the results are displayed as a list (results page). When defining such a join, an option *joinCheck* can be selected. In this case, the join is performed before the link is created, and if there are no results, the link is omitted.

Joins of meta type *details(inPlace)* are immediately performed and the values of all the attributes specified in *displayAttrs* are displayed on the details page for each result of the join. This type should be chosen carefully, because if the join leads to hundreds of results, the details page would get very long and practically unreadable.

The last meta type *details(extraPage)* can only be selected for an n:1 join. The join is performed and a link is created which leads to the details page of the only result of this join.

4 RELATED WORK

Due to the space limitations for this paper and the variety of products available, only a short survey of some products using similar technology can be given:

Cold Fusion (from Allaire)[1]: One of the first products offering database access to the web. Like the WEBCON

gateway it offers the use of templates containing special tags. But it does not provide a data dictionary based toolkit for querying or populating databases. The next version (3.1) will contain a new visual development environment.

Net Dynamics (from Net Dynamics)[4]: Supports the construction of templates corresponding to the query, results, details, and input pages described in this paper. A list of all attributes of a given relation is presented and the desired attributes can be specified by drag & drop. Joins can be defined in a similar way. Net Dynamics also enables the administrator to define the way in which the information will be displayed (similar to the meta types of WEBCON). There are two main differences:

First, Net Dynamics does not use the data dictionary for an automatic preselection of meta types available for a given attribute (for example foreign keys).

Second, Net Dynamics physically creates new templates whereas WEBCON just stores configuration information for the five universal templates.

Other products similar to Net Dynamics are SQL-Surfer[8] and Tango Enterprise[3], but with each of these products a new template for every query, results, details or input page has to be created.

5 CONCLUSIONS

This paper presented the WEBCON package for connecting databases to the World Wide Web. Based on a CGI gateway using the technique of server parsed HTML, this toolkit provides five universal, completely configurable templates used to build an application without having to write code in any language. The meta schema used for this purpose has been discussed in detail. The main benefits of WEBCON are a rapid and easy creation of the database web connection by defining some meta data, its easy maintenance via any web browser, and the dynamic concept making it rather insensitive to changes within the database schema.

The WEBCON package is currently used in the following projects:

- *M-Line* is a 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://mline.forwiss.tu-muenchen.de/>
- *TEX-Line* is 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* is an online datawarehouse bibliography.
URL: <http://line42.forwiss.tu-muenchen.de/>
- *TTM-Line* will contain 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 will be offered.

6 STATUS OF THE SOFTWARE

The WEBCON package has been developed within several research projects and is being enhanced continuously. As FORWISS, heavily funded by industrial projects, cannot provide PD packages, availability of WEBCON is restricted to usage within our projects.

7 LITERATURE

1. Allaire Corp.: Cold Fusion 3.0
Available at <http://www.allaire.com/>
2. Clausnitzer A.: *Realisierung einer WWW- und einer ASCII-Version der OMNIS-Recherche (Realization of a WWW and an ASCII version of the OMNIS search)*. Diplomathesis, TU Munich, Institute for Informatics, Dec. 15th 1994
3. EveryWare Development Inc.: Tango Enterprise 3.0
Available at <http://www.everyware.com/>
4. Henderson J.: *Delivering Solutions for the Web-centric Enterprise*. White paper.
Available at <http://www.netdynamics.com/about/whitepaper.html>
5. Hunter A., Ferguson R.I., Hedges S.: *SWOOP: An Application Generator for ORACLE/WWW Systems*. Fourth International Conference on the World Wide Web, Boston, 11-14 Dec.1995.
Available at <http://www.w3.org/Conferences/WWW4>
6. IBM: *Database 2 World Wide Web Connection Version 1 Application Developer Guide*. White Paper, 1996.
Available at <http://www.software.ibm.com/data/db2/db2what.html>
7. Informix Software Inc.: *Informix Web DataBlade Module*.
Available at <http://www.informix.com/informix/products/techbrfs/dblade/datasht/webdb.html>
8. Netaway: *SQL-Surfer*
Available at <http://www.netaway.com>
9. O2 Technology: *O2Web User Manual, Release 4.6*. Jan. 1996
10. Oracle: *Oracle Webserver 3.0*. White Paper.
11. TransAction Software GmbH: *Transbase Manuals Version 4.2.2*. Munich, 1996