Building 4GL Applications using the WWW

Herbert Groiss
Institut für Informatik
Universität Klagenfurt
Klagenfurt, AUSTRIA
email: herb@ifi.uni-klu.ac.at
http://www.ifi.uni-klu.ac.at/home?herb

The World-Wide Web has become a very popular media for accessing information from various sources. Structured information is typically stored in databases, therefore it is natural to use the WWW for accessing relational or object-oriented databases. Some commercial database producers are already offering WWW Gateways for their products (e.g. [4]) and applications for accessing databases via the Web have been developed, [2,7]. In this work we go a step further and use the WWW not only for querying databases but as interface for database applications performing all kinds of data manipulations. We describe WWW4GL, a language supporting the easy development of such programs based on the tradition of fourth generation languages [1].

The architecture of WWW4GL

The interaction of a user with the WWW runs via exchange of documents and forms. When viewing a document the user can interact with the server in two ways:

- follow a link or
- fill in a form and submit it.

Based on this basic modes of interaction a database application using WWW4GL is separated into several pages, which are connected via links and the via the results of procedures called when submitting a form.

Each page is defined using an extension of HTML containing some special tags for querying the database and inserting the returned values into the HTML document. The defined tags are:

`<db_head header>`	gives the name of the page and a list of parameters (for use in queries).
`<db_query query>`	contains a database query which is executed when the form is requested, the result values can be used in the rest of the page. The query should return exactly one row.
`<db_loop cursor_spec>`	*cursor_spec* is a query returning *n* rows, the loop block (ended with `</db_loop>`) is executed *n* times.
`<db_val varname>`	is replaced with a value returned from the query.
`<db_arg argn=varname>`	is used for concatenating arguments to hyperlinks, *varname* is replaced with the query result.

Input fields in forms are filled with the corresponding values of the database query. The enhanced HTML code is compiled to a Oracle PL/SQL [3] procedure and stored into the database.

In the WWW4GL page definition, the only communication mechanism with the database are queries. When an update or any other operation is needed, a submit procedure has to be defined in PL/SQL. Such procedures have the call of a WWW4GL page as their last statement. This page is sent to the client as result of the submit click. Because of the loose coupling between the client and the server no locking of selected values is possible and the user must be aware that his view of the data is a snapshot and the real values probably have changed already. If an update is performed, the program checks whether the old values are still valid and issues a warning if something has changed.

For simple database maintaining applications and as starting point in the development of a WWW4GL program a table editor is provided. For a given database table the WWW4GL definitions of a search form, a search-result page, a new-row form, and an update/delete form with the necessary links between them are created.

A small example shall show the possibilities of our language and how a page definition looks like.

Example

In an university (or a school) rooms for lectures must be reserved. An application for this task contains four pages typically used in the following sequence:

room search form → room-list → detail & booking form → confirmation

The applicant fills in a search form describing his needs and gets a list of available rooms fitting the requirements. In this list a click on a link to an individual room shows the details and provides a submit button for reserving the room. In the following the code of the page for the room list is shown:

```
<db_head roomlist_page(cap number, date1 date, begin_t number, end_t
       number)>

<head><title>Search result</title></head>
<h2>Available Rooms:</h2>

  <ul>
  <db_loop select name, description from r_rooms
            where cap <= capacity
              and name not in
                   (select b.name from r_room_res b
                     where date1=b.dat and (begin_t between begint and endt
                                        or end_t between begint and endt
                                 or begint between begin_t and end_t))>

  <li><a href=roomdetail?><db_arg room=loop1.name>
            <db_val loop1.name><db_val loop1.description></a>
  </db_loop>
  </ul>
```

The tag `dbhead` specifies the name of this page and the arguments, which must be provided (by the search form). The query in the `db_loop` tag returns a list of fitting rooms. For each room the name and the description is shown and every room is a link to the details page, named `roomdetail` and having the parameter `room` containing the room name.

Note, that is is possible to use nested loops, where the values are accessed as `loop`*`n`*`.`*`var`*, where *n* is the depth of the nest.

Conclusions

Using the WWW for database applications has several advantages: they are suitable as interfaces to users outside an organization: e.g. for booking tasks (flights, hotels, etc.), or for organizations distributed over many locations, where access to a central database is needed. Furthermore Web applications are not bounded to a special hardware or operating system.

WWW4GL allows the rapid generation of such applications: The forms and result pages can be designed and previewed using a WWW browser. The database queries can be embedded directly into the pages, update (or any other) operations are placed into PL/SQL procedures. We have implemented a prototype of WWW4GL using ORACLE and its WWW Gateway (WOW, [4]).

For further development, it is planned to take the advantages of new features in HTML 3.0 and of browser technology (e.g. [5], [6]), which will widen the spectrum of use of database applications over the Web.

References

[1] Robert Klepper and Douglas Bock. Third and Fourth Generation Language Productivity Differences. *Communications of the ACM*, 38(9), September 1995.

[2] J. Ng. *A Mosaic-SQL Gateway*. Technical report, National Center for Supercomputing Applications, 1993. *http:/www.ncsa.uiuc.edu/SDG/People/jason/pub/gsql/back/starthere.html.*

[3] ORACLE. PL/SQL Users Guide, Version 2, 1993.

[4] ORACLE. The Oracle World Wide Web Interface Kit, 1995. *http://dozer.us.oracle.com:8080*

[5] Sun. *Java Home Page*, 1995. *http://java.sun.com*

[6] M. van Doorn and A. Eliens. Integrating applications and the World-Wide Web. Computer Networks and ISDN system, 27(6), 1995. *http://www.elsevier.nl/www3/welcome.html*

[7] C.A. Varela and C.C. Hayes. Zelig: Schema-Based Generation of Soft WWW Databases. In *Proc. of the First International World-Wide Web Conference*, Geneva, 1994. *http://www1.cern.ch/PapersWWW94/cvarel.ps*