Designing Database-Intensive Web Pages in the WYSIWYG Interface

David Wolber, Yingfeng Su, Yih Tsung Chiang,
Department of Computer Science, University of San Francisco
2130 Fulton St., San Francisco, CA. 94117
(415) 422-6451 wolber@usfca.edu

ABSTRACT
WebDev is a programming in the WYSIWYG interface tool for building dynamic web pages that connect to databases. The system allows designers to “program” by entering query by example (QBE) and spreadsheet formulas into visual components. The system then automatically generates a dynamic web page (a template HTML file and a Java servlet invoking a DOM tree generated by the Lutris XMLC library[3]).

INTRODUCTION
The Spreadsheet is the prime example of a programming in the interface system, i.e., a system in which the development interface and the target interface are the same. Other examples include SmallStar Eager and Mondrian, all found in [2]. These systems allow users to extend graphics and text editors by demonstrating operations which are recorded and automatically generalized.

Programming in the WYSIWYG interface is different. Here, the development interface is not the target run-time interface, but a facsimile of it. The developer still gains some benefit from visualizing the development interface as the target interface, but he must mentally map the operations he performs during development to the actual operations that will occur at run-time. These tools are freer in a sense, as they can provide guides and even dialogs that help the developer. Of course the more guides and helper dialogs, the less WYSIWYG the interface is.

Macromedia's Ultradev 4 [4] is perhaps the most popular programming in the WYSIWYG interface tool for building web pages. Ultradev 4 is quite WYSIWYG for building static pages, but for dynamic pages, its development time interface is conceptually distant from the target interface--many dialogs are introduced and the programmers has to think of programming concepts such as parameters and result sets.

We have built a tool, WebDev, which allows most "programming" to occur in the WYSIWYG interface, not a dialog, and which allows the designer to work directly in the context of the target interface, not an abstract program. The tool applies techniques from the QBE, spreadsheet, and programming in the interface worlds to an HTML editor. Experience with the first users of the system suggest that the tool can significantly reduce development time for programmers, and increase the pool of designers that can build complete dynamic web pages.

With WebDev, the designer enters query expressions and formulas, but always in the context of the user interface, that is, directly in the cells of HTML tables (or other visual components). The designer never need enter an entire SQL or programming statement, or refer to result sets, parameters, or other items other than named items from the user interface.

The expressions and formulas entered in the development interface have a straightforward mapping to run-time operations. The key mapping is that each entry of a formula in the development-time HTML table (component) maps to two run-time operations--modifying the run-time HTML table, and modifying/accessing the underlying database table.

Development Process
When an HTML table is inserted into the editor, the designer can either map it to an existing database table by entering information in a dialog, or enter sample data for it, and tell the system to create the database table automatically. The development-time HTML table then represents both a run-time visual component and the mapped database table.

If the designer were to save the program immediately after this mapping, the resultant web page will display all records of the mapped database table in the visual table. Most the time, of course, the designer desires a more complex web page.

After the mapping, the development-time table appears with one sample row of "live" database data, and four other rows, empty except for labels in the left-most column that help guide the designer (see Figure 1).

The designer formats the sample data in each cell of the sample row. At run-time, the sample formatting of each column is applied to all records (rows) that appear (e.g., all titles in Table 1 will appear italicized).

The designer enters QBE expressions in the view select row to specify the database rows should be displayed. The designer need not understand SQL syntax, as all expressions are entered in the context of table cells. The designer need not even name the column data that the expression is entered for-- the column name is implied by the location of the entered expression. For instance, if the designer enters expressions in the view select row as in Table 1, the system will build the query:

\[
\text{SELECT * from BookTable} \\
\text{where InStock}>10 \text{ and Price}>22.00}
\]
and use it in the generated servlet so that only the selected books are displayed.

The delete selection row is similar to the view selection row, but the QBE entered is used to choose which rows are deleted from the database, not viewed. If “Kafka” were entered in the author column of the delete selection row in Figure 1, the following SQL statement would be generated:

\[
\text{DELETE * from BookTable where Author=Kafka}
\]

The add row facilitates the creation of pages that allow the end-user to enter new records in a table. If one enters constant data in the add row, a record containing that constant data is entered in the database table each time the page were invoked.

Input Forms

Selects, deletes, and adds are often based on something other than the constant data provided in the samples above. In many interfaces, the operations will depend on data entered by the user in an input form.

WebDev allows QBE expressions and the expressions entered in the add row to include references to the names of input form data. For instance, in the add row of Figure 1, the designer has entered the names of the input form text boxes (the designer named these boxes using a property box).

At run-time, when the Add Button is selected, the values in the text boxes will be used to add a record to the database. Note that the visual table will then be updated using the criteria in the view select row (InStock>10 and Price>22.00).

Query formulas and add expressions can also refer to input form data from another page in the development environment. The designer just qualifies the component name with the page name, e.g., "page1.minimumPrice".

Entry Points

A dynamic web page can be invoked from various other pages and even from the page itself. WebDev allows the designer to specify the result page of a button by right-clicking on the button and specifying the name of any open page in the development environment.

When a page is specified as a result page of a button, a new tab appears in the page's development-time window. In Figure 1, there are two tabs—the selected one corresponding to the Add button in the page, and the “Page1.submit” one corresponding to a button in another page. When the designer adds new components or modifies the layout of a page, these changes are reflected in all tab panels. But the QBE expressions and spreadsheet expressions entered are tab panel dependent, i.e., each tab panel specifies the actions that will occur when the page is loaded as a result of clicking the button the tab represents.

Spreadsheet Formulas

The facilities described above allow for the development of web pages connected to databases, but they don't allow computations on the data. Fortunately, WebDev provides enhanced spreadsheet functionality, similar to [1], that allows computations to be entered. The user can enter formulas both in the add row and in table columns that are not mapped to database fields. When the initial mapping to the database is made, any number of columns can be left unmapped-- these are "expression" columns. In expression columns of the sample row, the designer can enter formulas that refer to any of the column names of the table. For instance, in Figure 1, the designer has entered “Instock*Price” in the Total column.

At development time, the formulas are evaluated using the live database data in the other columns of the sample row. The computed value (e.g., 340.00) is placed in the cell and the formula appears at the far right of the toolbar. At run-time, the formula is evaluated for all rows in the database that are to be displayed (that are selected).

SUMMARY

WebDev allows database operations and computations to be specified in the context of the HTML interface. Our next step is to enhance the system so that complex table relations can be specified in the same manner.

REFERENCES
