A Visual User Interface for the Specification and Scientific Analysis of Continual Queries

Mary Margaret Heimovics
University of Kansas
Department of Electrical Engineering and Computer Science
megheim@designlab.ukans.edu

Abstract

Currently, scientists in the biodiversity community are encountering obstacles while attempting to track, synthesize and analyze biological collections information. In order to follow changes and evaluate trends in the biological information scientists are required to learn how to perform many complicated and cumbersome tasks. They must work to interface between database query environments and analysis applications and are hindered by the need for systems that track and recognize changes in database information. WebFormulate is an Internet-based system that provides an integrated environment for the facilitation of the development of analyses using information obtained from databases on the Internet. My thesis work is to develop the interface for the WebFormulate system; one that meets the complex needs of the biodiversity community yet maintains a simple interface that requires no prior programming knowledge or experience.

1.Introduction and Related Work

WebFormulate is an extension of *Formulate* [1, 2, 3, 4, 5, 6, 7], *WebFormulate* is an equation-based visual programming language with a user interface specifically designed for users with little or no programming knowledge. The *WebFormulate* user interface is very similar to the *Formulate* user interface with the exception that it runs in a web browser environment. *WebFormulate* also extends the functionality of *Formulate*, allowing the user to perform continual queries of distributed databases and to use the results of those queries for subsequent computations and visualizations.

The main distinction between the *WebFormulate* system and existing Internet facilities to retrieve information and assimilate it into computations is that *WebFormulate* provides the necessary facilities to develop and maintain dynamic links such that computations and reports automatically maintain themselves. This will empower researchers by facilitating automated retrieval and analysis functions through a network interface.

2.Understanding the Biodiversity Community Analysis Problem

As part of an NSF Division of Biological Infrastructure grant, *WebFormulate* will be tested by researchers in the biological collections community, users who hitherto have been unable to fully utilize their specimen collections data; a consequence of an abundance of independent database systems and an absence of software tools to track and analyze changes in those databases. Time series and regional analysis of biodiversity information, based on actual collections data within a distributed computing environment on the Internet, has represented an enormous challenge for the biodiversity research community. The distributed computing framework of *WebFormulate* will provide a powerful infrastructure for the integration of data sources and a means of achieving these data analysis objectives.

As an example of the type of problem *WebFormulate* could be used to solve, suppose that a biologist in California wants to graph population changes of a species based on data from a biological database that is maintained at a University in Kansas. Typically, this person would need to do the following to address this problem: (1) connect to the remote database, (2) construct a SQL statement, (3) download the query results, (4) import the data into a spreadsheet, and (5) construct a graph from the spreadsheet data. In order to monitor changes in the species population over time, the biologist would need to perform these various tasks repeatedly as the database is updated. In reality, it is unlikely that a biologist would have all the requisite technical skills (and perseverance) to perform these tasks. This example clearly demonstrates that an obstacle to the synthesis of networked data is the lack of a programming interface which permits end-users to easily formulate and execute scientifically meaningful and often complex queries of distributed

data sources and to be able to link the results of those queries to analysis applications.

WebFormulate allows users to develop "programs" by creating web pages (forms), dragging various types of objects from a palette in the user interface onto the form(s), and assigning values and/or equations to attributes of those objects. The population change problem described above could be addressed in *WebFormulate* by doing the following:

(1) Create a form with a 'database' object, specifying its value attribute to be the URL of the web-accessible biological database. The database schema will then be displayed within this object, shown hierarchically as a tree of the names of tables, related tables, and fields within the tables.
(2) Add a 'database query' object to the form with an SQL-like equation specifying the fields to be selected, the selection criteria, and the continual query specifications (i.e., a notification condition, a trigger condition, and a termination condition). Database fields referenced in this equation can be selected simply by clicking on nodes in the database schema representation of the 'database' object. The user is not required to type in table and field names, or construct a complex SQL statement involving joins between tables.
(3) Add a 'graph' object that references the fields of interest for this analysis (e.g., latitude and longitude).

Each time this continual query returns updated results, the 'database' object and the 'graph' object (which references the 'database' object) will automatically be updated on this form. If the form is closed, *WebFormulate* will automatically reconnect to the associated continual query process when the form is re-opened.

WebFormulate integrates three functions into a single application: (1) it provides a simple interface for querying complex databases, (2) it provides utilities for directly analyzing query results, and (3) it provides a mechanism that automatically maintains and updates query results and analyses as the data set changes through time. The integration of these components will make this system extremely beneficial for research scientists, dramatically reducing the amount of redundant work they must perform and the technical skills they must have in order to query and track changing information in databases.

3.Summary

It is important not only that information be accessible to the public, but that the same public be able to combine this information into effective analyses. The imperfect, dynamic, and "unknowable" structure of databases, and the inability of non-programming scientists to formulate queries against multiple Internet databases has been a significant impediment to research. The technology developed for the *WebFormulate* system will address these problems and will contribute to the general advancement of the Internet and its impact on industry, government, and education.

References

[1] A. Ambler and A. Broman, "Formulate Solution to the Visual Programming Challenge," in *Journal of Visual Languages and Computing*, 9(2), April, 1998, pp. 171-209.

[2] J. Leopold and A. Ambler, "A User Interface for the Visualization and Manipulation of Arrays", in *Proceedings of IEEE 12th Symposium on Visual Languages*, 1996, pp. 54-55.

[3] J. Leopold, "A Multimodal User Interface for a Visual Programming Language", Ph.D. Thesis, University of Kansas, Lawrence, Kansas, 1999.

[4] J. Leopold, A. Ambler, M. Heimovics, and T. Palmer, "A Visual Query System for the Specification and Scientific Analysis of Continual Queries", in *Proceedings of IEEE Symposium on Visual Languages*, Stressa, Italy, September 2001.

[5] G. Wang and A. Ambler, "Invocation Polymorphism", Proceedings of IEEE Symposium on Visual Languages, Darmstadt, Germany, September 1995, pp. 83-90.

[6] G. Wang and A. Ambler, "Solving Display-Based Problems", in *Proceedings of IEEE 12th Symposium on Visual Languages*, 1996, pp. 122-129.

[7] G. Viehstaedt and A. Ambler, "Visual Representation and Manipulation of Matrices", in *Journal of Visual Languages and Computing*, Volume 3, 1992, pp. 273-298.