Information Translation, Mediation, and Mosaic-Based Browsing in the TSIMMIS System*

SIGMOD Demo Proposal (final version)

Joachim Hammer, Hector Garcia-Molina, Kelly Ireland,
Yannis Papakonstantinou, Jeffrey Ullman, Jennifer Widom

Department of Computer Science
Stanford University
Stanford, CA 94305-2140
Tel.: (415)723-3118 Fax: (415)725-2588
E-mail: `joachim@cs.stanford.edu`

1 Overview

The TSIMMIS project [2] provides an architecture and tools for accessing multiple heterogeneous information sources by translating source information into a common self-describing object model, called the *Object Exchange Model* (OEM) [3]. TSIMMIS provides integrated access to heterogeneous sources through a layer of source specific translators as well as "intelligent" modules, called mediators. Translators (wrappers) convert queries over information in the common model (OEM) into requests the source can execute. The data returned by the source is converted back into the common model. Mediators are programs that collect information from one or more sources, process and combine it, and export the resulting information to the end user or an application program. Users or applications can choose to interact either directly with the translators or indirectly via one or more mediators.

Figure 1 shows an overview of the TSIMMIS testbed as it is currently operating in our lab. We distinguish three different layers: the source layer, the server layer, and the client layer. Starting at the bottom (source layer), we can see three (disk-shaped) heterogeneous information sources, containing bibliographic data in three heterogeneous formats:

1. A University-owned bibliographic library called FOLIO which is accessed through an interactive front-end (called INSPEC).

2. A Sybase relational database containing bibliographic information.

3. A UNIX flat file containing a personal bibliography similar to the one used by TEX.

Above the sources, in the server layer, are translators (one for each source) that convert the underlying data objects into OEM. To do this logical conversion, the translator converts queries

*This work was supported by ARPA Contract F33615-93-1-1339, by the Anderson Faculty Scholar Fund, by the Center for Integrated Systems at Stanford University, and by equipment grants from Digital Equipment Corporation and IBM Corporation. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of the US Government.

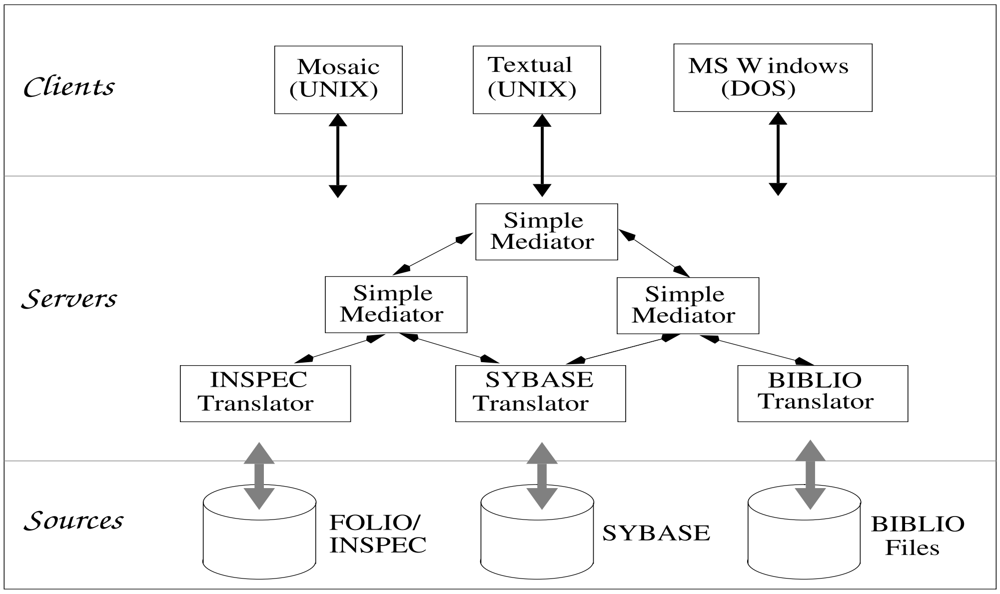

Figure 1: TSIMMIS testbed

over information in the common model into requests that the source can execute, and it converts the data returned by the source into the common model. Above the translators lie the mediators. A mediator embodies the knowledge that is necessary for processing a specific type of information. For example, a mediator for "publications on database systems" might know that the relevant information sources are FOLIO and BIBLIO (bottom of Figure 1). When a mediator receives a query, say for information on "deductive databases," it will know to forward the query to those sources. The mediator may also process answers before forwarding them to the user, say by converting the references into a standard format, or by eliminating duplicates.

In the top-most layer we can see three different clients that are currently implemented: a graphical user interface based on DOS and Microsoft Windows, a UNIX-based textual interface, and a graphical interface based on Mosaic and the HTTP protocol [1,4] called MOBIE (MOsaic-Based Information Explorer). Our demo uses MOBIE since it is the most interesting of the three. MOBIE is a platform-independent tool for displaying and exploring OEM objects that are returned as a result of queries. MOBIE provides a mechanism for navigating through objects, zooming in on their nested substructures as necessary. It is not required to know in advance the structure or "schema" of the data being explored. MOBIE uses HTML commands to format the portion of the object space being displayed, using indentation and links to represent the data structure and relationships. An important advantage of using Mosaic as the basis for our user interface is its widespread use and popularity.[1] Ultimately anyone on the internet will be able to use TSIMMIS and MOBIE to explore any information source on the net, provided there is an appropriate OEM translator available for it.

We have implemented a complete version of the testbed shown in Figure 1, consisting of translators for the three sources, several simple mediators for integrating bibliographic data, and

[1]Mosaic currently operates on Unix workstations, on Macintosh computers, and on many PC's.

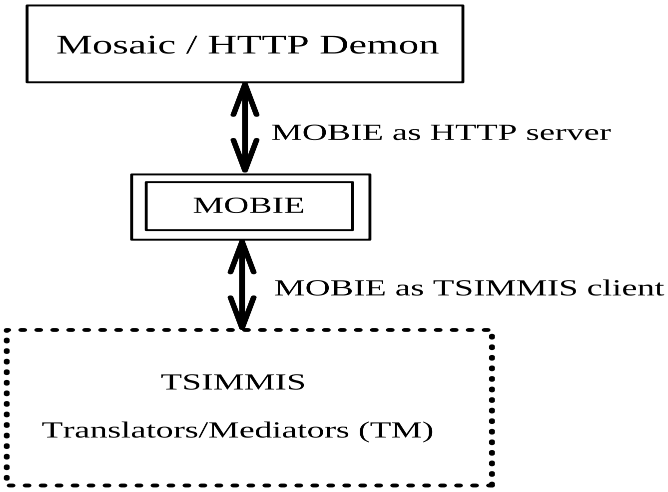

Figure 2: MOBIE acting as a client and server

three clients. Based on this functionality we are able to access and browse heterogeneous objects from different sources in a uniform way. Although there is considerable substructure, since the SIGMOD demo will illustrate the browsing interface primarily, we will focus on the interface in the next section.

2 The Browser

MOBIE lets end users connect to mediators or translators and specify queries using the TSIMMIS query language OEM-QL[2]. MOBIE is a collection of programs that exhibits two kinds of behavior: On one side it acts as a *client* to a TSIMMIS mediator or translator (henceforth referred to as TM) and on the other side it acts as a *server* to the HTTP demon and Mosaic. If we start our examination of MOBIE at the Mosaic side (MOBIE acting as a server), we notice that MOBIE receives requests that are submitted to the Mosaic interface and translates them into queries that can be understood by the underlying TM. The interaction between Mosaic and MOBIE is based on Mosaic's version 2.4 "fill-out-form" feature. User input is collected in the input fields on the form, packaged into an environment variable, and transferred to MOBIE (using HTTP). Here the query string is parsed, translated into OEM-QL syntax, and sent to the appropriate TM. On the TM side, MOBIE (acting as a client) processes all information that is returned from the TM and translates it into dynamic HTML result pages that are displayed by Mosaic. The initial fill-out form page is the only static HTML page in our system (by static we mean stored in a file on disk). All other pages are created dynamically (from within a C program) since they depend entirely on the information (size, structure, ...) that is being returned from the TM. The interaction between MOBIE and Mosaic and between MOBIE and TSIMMIS is shown in Figure 2.

It is important to note that besides providing a user-friendly interface and submitting and receiving information to and from TM, MOBIE serves an additional important purpose. It "buffers" the results of a query until the data is ready to be displayed by Mosaic, which is by itself stateless (i.e. there is no notion of memory in Mosaic other than the ability to return to a certain number of previously visited URLs). The user has the option to decide (1) which information is to be

[2]OEM-QL is similar to SQL and exhibits many of its features such as the familiar `SELECT-FROM-WHERE` structure.

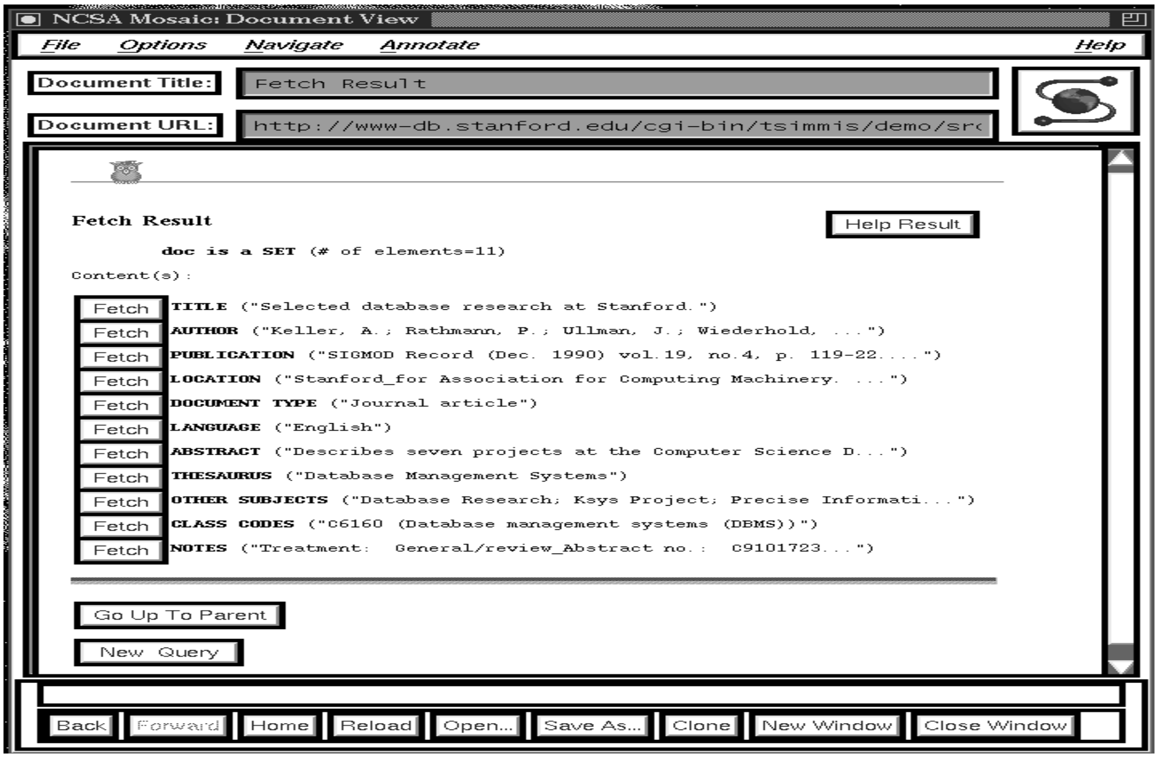

Figure 3: *Fetch Result* page displaying a selected document

displayed, (2) how much of the chosen information s/he wants to see, and (3) when. Information is presented one screen at a time, allowing the user to browse complex objects, which may be too large to view all at once, in a "cafeteria-style" (pick-and-choose) fashion.

We illustrate MOBIE's operation by walking through a particular interaction. The first step in accessing information through MOBIE is to select a translator or mediator on MOBIE's home page[3] and to connect to it. After the connection is established, a *Query Request* page is displayed and the system is ready to accept an OEM-QL query. If a submitted query is valid and successfully executed by the TM, the answer object is returned to MOBIE and displayed on a *Query Result* page. Except for very small objects, to see the complete result the user will move through the structure of the answer object using MOBIE's *navigational capabilities.* In OEM, answer objects are structured like trees (or graphs, in the more general case) [3]. The user can move from the top level object to a lower level by traversing HTTP hyperlinks (`Fetch` buttons) which invoke subsequent result pages. The result of clicking `Fetch` for one document in the initial collection is shown in Figure 3. The subobjects of this document are labeled `Title`, `Author`, etc., and their values can again be fetched by clicking on the `Fetch` buttons. These "lower level" result pages either display the complete subobjects (in case they are atoms) or links to further subobjects on additional result pages.

[3] Mosaic displays information through a series of text screens or *pages*, the first of which is always called the *home page*.

3 Status and Demo

At SIGMOD we propose to demonstrate at least the first and currently completed version of our TSIMMIS testbed as seen in Figure 1. We are currently working on the following improvements and extensions which we expect to incorporate into the demo as well.

- Additional sources (e.g., object-oriented database) and more translators
- Additional more sophisticated mediators
- Improvements to the user interface (e.g., "smart" layouts that maximize the information that can be conveyed to the user on each screen, source profiles that allow users to customize the way in which information is displayed on the screen)

References

[1] T. J. Berners-Lee, R. Cailliau, and J.-F. Groff. The World Wide Web. *Computer Networks and ISDN Systems*, 25:454–459, 1992. North-Holland.

[2] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. The TSIMMIS project: Integration of heterogeneous information sources. In *Proceedings of the 100th IPSJ*. Computer Society of the IEEE, Tokyo, Japan, October 1994.

[3] Y. Papakonstantinou, H. Garcia-Molina, and J. Widom. Object exchange across heterogeneous information sources. In *Proceedings of the Data Engineering Conference*. Computer Society of the IEEE, Taipei, Taiwan, March 1995.

[4] S. J. Vaughan-Nichols. How to glue together Mosaic (internet browser). *Government Computer News*, 13(15):33, July 1994. Tutorial.