

Kiosks for Tourists: Heterogeneous Distributed Database Access for Multimedia Information Presentation

Michael Wilson
Rutherford Appleton Laboratory

ABSTRACT

When accessing on-line electronic information from databases it is necessary to query each independently, retrieve the information and integrate the answers. This requires a knowledge of each information source, its access protocols and representation of information. The MIPS system stores information about these aspects of each information source and permits the user to issue a single query which is issued to each information source, and the answers are automatically integrated into a hypermedia presentation. The system has been demonstrated by providing information for tourists about the island of Corfu, its history, museums, entertainment and hotels, and could be exploited for tourist or other applications elsewhere.

When tourists arrange their holidays today they use brochures produced by tour operators, travel guides, maps, the advice of travel agents and the information they supply about the latest bargains and what is actually available. Most of these resources are now available in some computer accessible form from airlines, tour operators, regional tourist offices or local hotels and entertainment centers. However, if tourists try to access each of these information sources they have to know a series of access protocols, and understand many formats for querying databases and interpreting the answers.

This is an example of a general problem of accessing heterogeneous distributed electronic information sources. If a new office is being designed which uses about 10 databases, then they will all be created to use the same formats and access protocols so that a single query can be issued to them together as a distributed information source which returns a single reply. Alternatively, if one is trying to access information spread around the World Wide Web, it is stored in a set of common formats which can be searched and browsed from a single tool, but users must spend their time grinding through the sites suggested by a search engine as the result of an initial query. However, if the number of data sources ranges from ten to one hundred and they are used regularly by a person or community it is possible to apply a different technology which allows a single query to be issued that will be expanded and sent to each of the information sources, then the returned information will be integrated into a single presentation to the user.

Modern relational databases hold information in tables where each column contains an attribute for a record, or row. For example an airline flight may be represented as a table with columns for attributes such as flight number, departure location, arrival location, departure time, arrival time, total number of seats, number of vacant seats, and price. However, such databases are normally constructed to be optimally efficient

at storing and retrieving the information which they contain. To do this, only the minimum information is stored. There is no explanation of the currency in which the price is stored, or the format in which times and dates are represented. Not only may the date be represented in text or numbers, but it may use the American or British ordering (1/12/96 or 12/1/96 for the first day of January in 1996). Representations vary from one country, and indeed one operator, to another. If the user wishes to enter a single query that can access many databases which have different internal representations it is necessary for the program to know the representation used in each target database, and to translate the query into the appropriate terminology, and then translate all the answers back into a common format for presentation to the user. This problem is well known, and the theoretical problems have been considered for many years (1). Indeed the variations considered become bizarre beyond the different classes of seat and pricing structures, to cover the inclusion of sales tax in the price, and whether this has already been added (as in the United Kingdom) or not (as in the United States) and whether this should be added at the rate applicable at the site of the information or the site of the user.

With modern information systems the problem of heterogeneity is increased with the inclusion not only of textual and numeric data, but of images in many formats, and even sound and video which may need to be synchronised together after the sound has been chosen in the appropriate language for the user. As well as including database representation and multimedia format variations, there are also variations in the programs chosen to store the information. These may be databases from different manufacturers which use a common language for querying the information such as the Standard Query Language (SQL) or they may use proprietary languages for both querying and structuring the returned data as an answer. Furthermore, the storage system may not be a database, but a hierarchical file structure or a text storage system which each have their own access languages and representational formats.

It can be appreciated that the range of possible storage formats, query languages and returned data representations is considerable, and any programme which will translate a user's query into the appropriate one for each information source requires considerable information about that information source, and about the user who is asking the questions in order to present the information in the best form. Since the information to be retrieved extends beyond numbers and text which can be easily displayed in tables, a hypermedia presentation is required which will include text, images, video and sound structured on a series of pages which are linked together so that the user can navigate between them to browse the answer to the information in a manner which will clearly answer the question posed. This automatic generation of a hypermedia presentation from the returned information poses a second and equally complex research problem to that of the access of heterogeneous distributed information.

The European Union partly funded the MIPS project to address these two problems through its Esprit III research programme. The three year project was undertaken by a consortium of industrial and academic organisations from across Europe: Cartermill International, the Rutherford Appleton Laboratory, and Herriot-Watt University from the UK, Sema from Belgium, Trinity College Dublin from Ireland, DTI from Denmark, STI from Spain, and Epsilon Software from Greece. The resulting information retrieval

system was demonstrated as an automated tourist brochure in a kiosk in the Greek island of Corfu for the Corfu Initiative, a consortium of tourism related organisations in Corfu.

The structure of the MIPS information retrieval interface is to support the eight stages of :

- 1) to expand the query provided in order to enrich the information description;
- 2) to map this description to the largest possible set of stored descriptions of available information sources;
- 3) to reduce the number information sources according to processing constraints of redundancy, time, cost and reliability;
- 4) to retrieve the currently available information;
- 5) to constrain the amount of returned data if there is too much;
- 6) to convert the returned data into common formats;
- 7) to resolve any conflicts within the data set;
- 8) finally to design the presentation of the retrieved information.

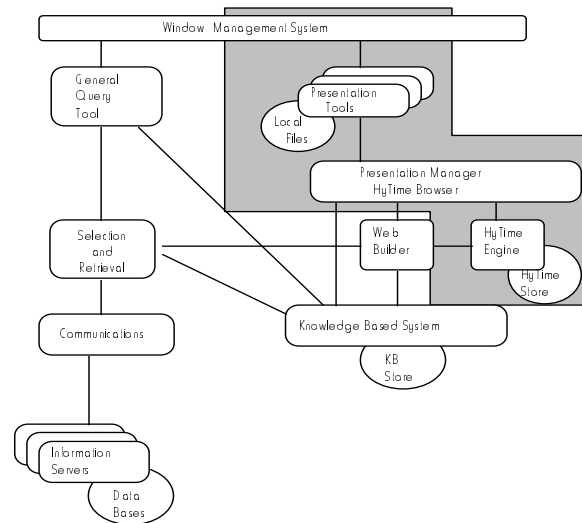


Figure 1 - The architecture of the MIPS system.

The architecture developed for the MIPS application is shown in Figure 1, consisting of a General Query Tool through which the user enters information about themselves such as their preferred language, and into which the user enters a query by either selecting from a series of menus, or by completing a series of illustrated forms. The menu option permits more complex queries to be constructed, while the forms interface is easier for users unfamiliar with the system. The query once constructed here is passed to the Selection and Retrieval tool which performs the first seven stages of the above sequence in conjunction with the Knowledge Based System (KBS). This

interaction addresses the heterogeneity problem and will be returned to later. The output of the Selection and Retrieval Tool is a nested relational table containing the answers to the query which are passed to the Web Builder which interacts with the KBS to produce a hypermedia web representation of the answer in the HyTime (2) language. HyTime is an ISO/IEC standard for hypermedia time based information develop on top of SGML which is richer than the HTML language used in the World Wide Web, and whose functionality is slowly being adopted into it as versions progress. The web is stored locally in a HyTime store, from where it is presented to the user by the Presentation Manager which reads the HyTime web document and calls Presentation Tools to present the retrieved information to the user.

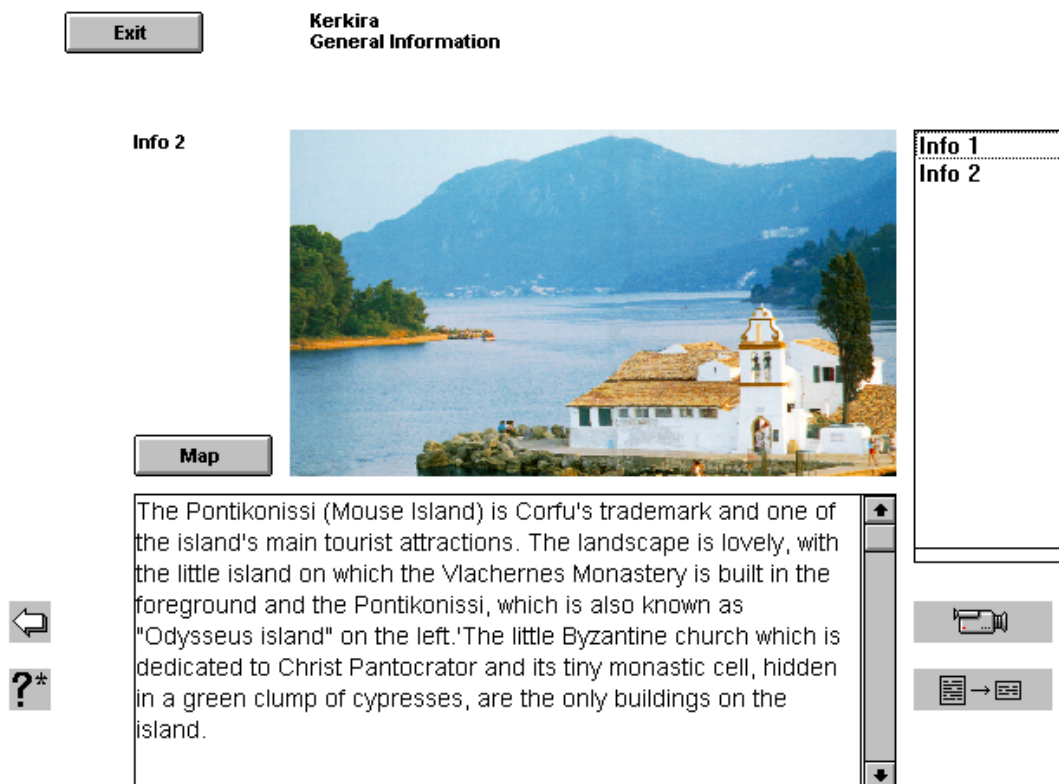


Figure 2: The MIPS user interface to the tourist information system showing the top page created in response to a general query about Corfu.

The approach taken to the problems of heterogeneity of the information sources is to store a large amount of information in the KBS about the domain (in the demonstrator tourism), the query language of the information source, the database or document structure of that information source, and the formats of returned data, as well as about the user themselves. This information is called upon by the Selection and Retrieval tool in the first of the eight stages to expand the simple queries (such as 'tell me about Corfu') into the set of all possible domain information which may be asked for (e.g., a map of the island, and city; text about the history; lists of hotels and details about rooms and facilities available in each hotel). Secondly, this lengthy domain description is mapped to the information available in any of the information sources which are known about, so that only that which is available is asked for. Thirdly, the number of information sources from which information is requested is reduced by constraints set by the user on the amount of time and money they will spend locating the information,

and the recency of the required information (this is more important in business applications where alternative sources of share prices may be available at different costs inversely proportional to the recency of the share price information). Fourthly, the queries are converted to the query language and representational terminology of each target information source, and dispatched to them to retrieve the information. Fifthly, some of the returned information may be discarded if there were too much of it according to further rules in the KBS. Sixthly, the returned information is converted into common currencies, date formats etc. on the basis of the KBS knowledge of the source database and a set of conversion algorithms which are chained together as required. Once the data is all in the same format and notation, it may be that conflicts exist which have to be resolved on the basis of the accuracy or recency of the source of the information, or on the reliability of the format conversion. This leaves a single set of multimedia information in answer to the expanded query which is stored in nested relational tables and passed to the Web Builder.

The Web Builder uses answer templates that are selected to match the structure of the expanded query and which are instantiated with the information from the relational tables. Where templates do not exist for the information requested and returned, the KBS generates them using design rules through a hierarchical constraint based planning mechanism (3) from simple building blocks for the presentation of images, video, text etc. or widgets which can be used to present menus of items or tables. The web structure must not only include the returned information but must also include navigational links from one page to the next which are generated to reflect the internal structure of the expanded query.



Figure 3: A Kiosk in Corfu employing the MIPS system to provide tourists with information.

The demonstration application of the MIPS system included databases with information about 28 hotels on Corfu, the town, museums etc. including text, images, sound and video. A single general query can be asked by a tourist, and a presentation of the island, city or individual hotels can be generated. The demonstration was used (maybe still) in several kiosks on the island to provide information to tourists. The application could also be used by specialist tour operators or even travel agents in northern Europe who wishes to generate personalised travel brochure for customers. If the databases connected include not only those of regional tourist centers, but also of individual hotels, such brochures can be accurate up to the minute about which hotel vacancies exist, which excursions are available, what local entertainment is available and the prices of each. The demonstrator application was developed for the Greek tourism market in Corfu, but this technology can obviously be applied to anywhere which wishes to attract more select tourists who wish to plan on the basis of accurate knowledge about local facilities in order to construct their own personal package holiday.

The technology that solves the problems of heterogeneous distributed information access and the automatic creation of hypermedia presentations has been demonstrated for tourism, but it can be applied to any organisation or industrial situation where users regularly access between ten and one hundred information sources and need to integrate the answers into a single presentation.

References

- 1) Sheth, A.P., and Larsen, J. Federated Database Systems for managing distributed, heterogeneous and autonomous databases, *ACM Computing Surveys*, 22(3) (1990), 183-236.
- 2) Newcomb, S.R., Kipp, N.A., and Newcomb, V.T. 'The HyTime hypermedia/time-based document structuring language' *Communications of the ACM*, 34, (1991) 67-83.
- 3) Borning, A., Freeman-Benson, B., Wilson, M. Constraint Hierarchies. *Lisp and Symbolic Computation*, 5 (1992), 223-270.