



Distributing Geographical Information Systems and Data Using Java and the Internet

David R Hill
 Institute of Hydrology, Crowmarsh Gifford,
 Wallingford, Oxfordshire, OX10 8BB. UK
 + 44 (0) 1491 692461
 E-mail. drh@ua.nwl.ac.uk

Presented at the second annual conference of GeoComputation '97 & SIRC '97, University of Otago, New Zealand, 26-29 August 1997

Keywords

query	environmental	time series
spatial	4-dimensional	data model
database	SQL	GIS
Java	CGI	WWW
RDBMS	Internet	distributed

1.0 Introduction

Developing efficient and effective storage and access methods for large environmental databases is one of the main research aims of the Data and Software Systems group based at the Institute of Hydrology (IH).

The Institute of Hydrology investigates the effects of land-use, climate, topography and geology on the volume and character of water resources. It focuses on understanding water and energy fluxes arising from processes such as evaporation, interception and infiltration and modelling the hydrological cycle and chemical processes above and below ground. It is the aim of the Data and Software Systems group in conjunction with the scientists to design and implement software products for the dissemination of IH science. Many of these products involve the design and use of databases which are also used to manage IH's own environmental datasets. Much of the fundamental research behind IH's database designs took place in the period 1974 to 1990 during which time many of the commercial GIS packages which are in use today were not available or unable to deal with many of the problems presented by environmental datasets. As commercial GIS pack-

ages developed throughout the 1990's, research and development at IH moved to concentrate on environmental database design. There have been two key problems that IH has sought to ameliorate. One is that at present different data types are held in different systems making it difficult to explore relationships that span the different data types. The other is that the demand for data exceeds the IH Data Centre capacity to supply them. This paper will elaborate the problems and describe the underlying concepts involved in their solutions. It will then propose some suggestions for providing a simple query interface for environmental databases, that can be made available to remote users anywhere. The points made will be illustrated by reference to the IH's work on the Land Ocean Interaction Study (LOIS) (NERC, 1992).

2.0 Environmental Database Management

To improve understanding of coastal zone processes, NERC has invested over £25M in the LOIS programme (NERC, 1994). LOIS is a multi-disciplinary programme to study the movements of chemicals and then fluxes from the land into the rivers, out through estuaries and finally to the continental shelf and beyond. Information is vital to such a programme; the effective collation and manipulation of data from a wide variety of sources and subject areas being one of the keys to attaining the programme's scientific objectives.



2.1 Data Requirements for large thematic programmes

In order to manage effectively such a large data operation, the LOIS programme managers set up a data infrastructure. This infrastructure consisted of a Data Steering Committee and five Data Centres responsible for the acquisition and distribution of data from and to the researchers. The Rivers Data Centre, based at IH, is one of the five Data Centres and is responsible for acquiring, storing and distributing river based datasets. It is here that the systems described in this paper are being developed.

The diversity of the datasets that the database must accommodate creates a major challenge in terms of database design. These datasets include data that vary both in space and time ranging from river flow, water chemistry, species distributions, digital elevation data and river networks through to satellite images. It is the collective aim of the Data Centres to design and implement a unified database or environmental information system which is capable of bringing together these diverse datasets within one holistic database system. The hope is that by grouping all of these datasets within one integrated system, the task of researchers developing complex environmental models which cross component boundaries will be eased. This philosophy is supported by T.J.Browne (1995) who suggests that for an information system to be successful it must be holistic and interdisciplinary in approach.

2.3 Data acquisition and dissemination

Supplying and managing data for such a large thematic programme presents numerous problems for Data Centre managers, whose objectives are:

- To acquire major datasets from within and without NERC and make them available to the LOIS community.
 - To establish standards for data definition and exchange formats.
 - To provide data management services for LOIS data.
 - To ensure long term security of the LOIS data and their availability to future science projects.
- Traditionally, researchers obtain data from a Data Centre by writing, telephoning, E-mailing or completing a form on

the Internet detailing the data that they require. The Data Centre then processes the data request and retrieves the data from the database. This process often incurs a delay as data requests may not be serviced immediately, however the formulation and execution of the query in the current system must be performed by Data Centre staff. Once the data have been retrieved they can be supplied to the user either by E-mail, FTP or the postal system. What both the Data Centres and scientists would like is the ability to browse and retrieve data remotely via the Internet. The acquisition of data suffers from similar problems. Presently, data arrive on many different forms of media and in many different formats. At each stage in the movement and translation process there are opportunities for data loss, corruption and delay. Advances in databases, networking and computer technology are now enabling these processes to be undertaken on the Internet and this will be discussed in the second part of the paper.

Before such an Internet solution can be designed, a clear view is required as to how the user can browse such a diverse array of datasets. It is a fair assumption that many users browsing the database will not have a detailed understanding of either the system or the types of data held within it. It is also likely that support will be minimal and that they will not want to master different methods of interrogation for each data type. Therefore, it would be an advantage if the user can perceive all data, whatever their type, to be held in one simple logical structure. Such a solution has been explored in the Water Information System (WIS) as described below.

3.0 Environmental Information Systems

To achieve data integration for the LOIS programme the Rivers Data Centre at IH is using the Water Information System (Tindall and Moore 1997; Moore 1997). WIS is an environmental information system which was designed and developed at the Institute with the backing of International Computers Ltd. Essentially WIS is a conceptually simple data model capable of storing generic types of data (Hill and Bellamy, 1996). It is implemented in a Relational Database Management System (RDBMS) on top of which sits a

UNIX based user interface. This provides an interactive geographical front end enabling users to visualise their data. The current implementation of WIS requires the data model to be implemented in ORACLE and the user interface software operates on a Sun workstation running SunOS 4.1.X. Both the hardware and operating system are now elderly and present various problems in terms of continued hardware and software maintenance. Much has been learnt since the initial development of WIS and what follows describes both the existing system and the improvements currently being implemented. However, the core of the system, the database design, has survived the test of time with only minimal modifications.

4.0 Database design

The WIS database design to which the system owes its immense flexibility, is best described in two parts, firstly the logical database design and secondly the physical database design.

4.1 Logical Database Design

4.1.1 Conceptual view of the data model

The logical database design provides a simple conceptual model which helps users to visualise how their data are stored. It allows the user to record the history of any object, or feature as it moves through space and time (Moore and Tindall, 1992). Descriptions of features and the events observed at them are recorded in terms of variables, parameters or determinands, known collectively in WIS as attributes. Thus, to store river water quality data, an individual monitoring site might be classified as a feature and the variables which describe or are observed at the site, such as its position, the site name, a unique reference number, river flow, pH values and so on, would be its attributes. Other examples of features could include roads, urban areas, maps, sewage works, licences and satellite images. WIS supports a wide range of spatial and non-spatial data types allowing the user to record most types of attributes. Examples of LOIS attributes could include names, reference codes, colours, centre lines, boundaries, soil types, the concentration of mercury and tem-

perature. Both features and attributes are decided and defined by the users and their system and user definitions are stored in data dictionaries.

All attributes are assumed to be potentially time variant so even positional attributes may form a time series. For example, although a land based river monitoring station has a grid reference that is unlikely to change, marine and airborne sampling campaigns are conducted from a base that is constantly moving.

4.1.2 The WIS Cube

The description above provides one view of the logical design of the database. An alternative view of the same data is to imagine a cube of individual cells, as shown in figure 1.

The three axes of the cube represent features (where observations are made), attributes (what has been observed) and times (when the observations are made). Each cell contains a value (or values depending on the attribute's data type) of an attribute describing a feature at some moment in time. For example one cell might contain a real value representing the rate of flow in the river Thames at Teddington on the 20th May 1997. There are no constraints on the number of features, attributes or occasions which can be stored by the cube other than that imposed by the physical limits of the hardware. Listed below are the key properties of the WIS cube:

- Any attribute may be observed at any feature;
- A feature may have any number of attributes;
- Any number of values may be recorded for an attribute over time at a feature;
- The values may be recorded at fixed or random time intervals;
- The data model does not distinguish between spatial and temporal data;
- The Cube is infinite in all directions;
- The significance of the cube is that it provides a completely generic data independent structure around which to build equally generic tools for data load, retrieval and analysis.

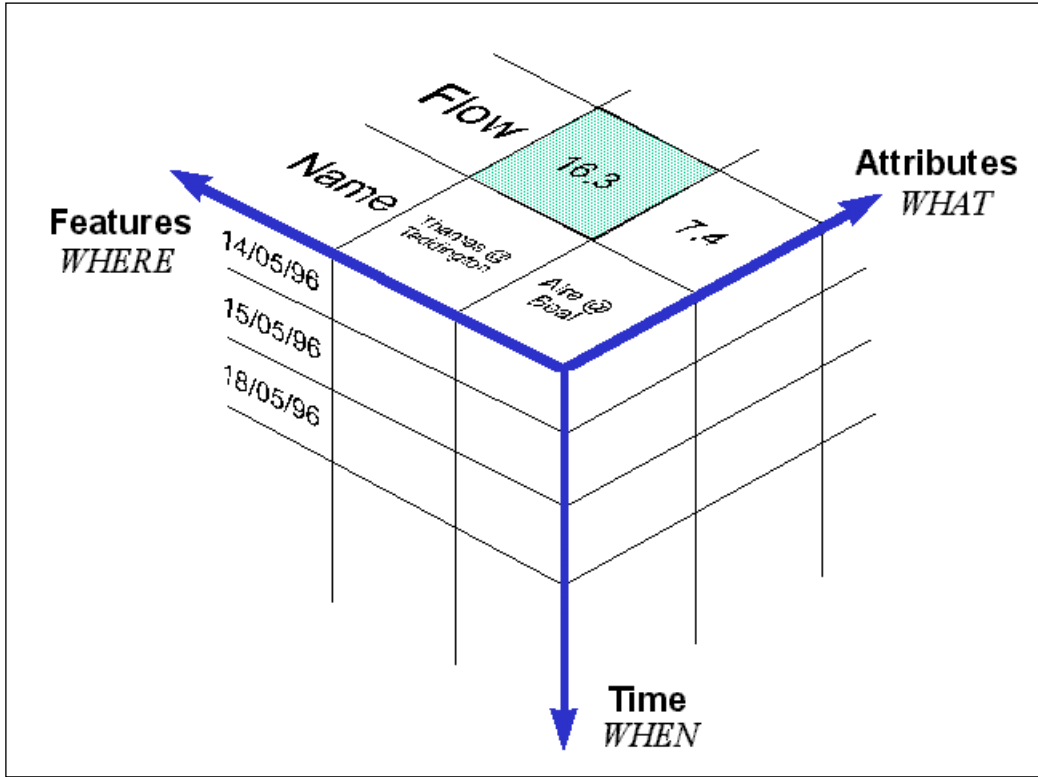


Figure 1. The WIS logical data model. The WIS cube.

4.2 Physical database design

The WIS cube can be implemented in any RDBMS. The underlying tables contain three different types of data;

Data Tables

The data tables share a common basic design and differ only in that different data types need different columns to hold the values. Each row in a data table contains a value from a cell in the cube. The first three columns of a data table contain the cube co-ordinates of the value, for example every value will have a feature ID (FID), an attribute ID (DID) and time ID (TID). For simple data types a fourth column holds the value. Thus, the DT_REAL table contains real values stored in a column called RVAL. Integer values are stored in the DT_INTEGER table in a column called IVAL. However, more complicated data types such as points are stored in the DT_POINT table and require three columns called X,Y and Z to represent their values. Other data types include names, character data, line data,

grid data and binary (OLE) objects. Binary objects can be used to store JPEG images or, for example, Microsoft Word documents within a cell of the cube.

List Tables

The WIS search and select model relies on the concept of lists. A list contains a set of feature identifiers, attribute identifiers or date/time ranges which pick out the data required from the cube.

For example a 'where' list contains a set of features of interest. A 'what' list contains a list of attributes of interest. A 'when' list would contain a subset of the time axis. Combinations of what, where and when lists are also possible as in a 'where/when' list. An individual list is created by constructing the equivalent of a 'where' clause in a Structured Query Language (SQL) query. The range of logical operators, however, is greater and includes spatial operators and a facility to exploit parent/child relationships between features. Complex queries are possible by using



set operators on lists, such as UNION, MINUS and INTERSECTION. Sophisticated facilities for time matching are included that allow for the fact that values relate to different periods; some referring to an instant, others a day and yet others to a month or year. For example, a problem in the past has been that rainfall data are attached to rain gauges and river flow data are attached to gauging stations. Selecting flow data for occasions when it was raining was difficult if not impossible on most systems. The list approach allows the construction of 'when' lists of occasions when it was raining, which may then be used to extract flow data.

Reference data

Reference data can be divided into;

Standard data. Examples of these include units of measurement, methods, periods, statistics, methods, qualifiers and validation status codes.

Field and structure definitions;

These are the definitions of the data types that the system supports.

Feature type definitions;

A feature type is the primary classification of a feature and is the only mandatory attribute in the WIS data model.

Attribute definitions;

Attribute definitions comprise both the system and user information. The user information comprises of an identifying code, name, definition and reference. The system data include its datatype (structure), period, statistic and internal identifier.

Most users are completely unaware of the physical implementation of the database. However, application writers, programmers and modellers often want to interface directly with the database at a low level. To make this possible an object orientated database Application Programming Interface (API) is being designed and is currently being implemented for the latest version of the data model. The database API will provide the main access route to the database at a programming level. It has two roles: to make access easy and to protect the database from corruption. Generic data models are nearly always more difficult to query than specific ones. The API allows the user

to express requests in a convenient way and then generates the SQL to answer them. The aim behind the API is to allow the programmer to think of the 'cube' as though it is a 3D array in memory. Assigning and using values in the database will be achieved by simple arithmetic statements. For example, a programmer could retrieve a value or update a value using the following statements;

value = mydatabase.cell(FID,DID,TID)

or

mydatabase.cell(FID,DID,TID) = value

Self evidently, rigorous validation checks for data that might corrupt the database will be included.

5.0 Distributing GIS and Data via the Internet

5.1 A changing computing paradigm

The original purpose of the generic data model was to facilitate the exploitation of relationships that span data types and to avoid the need to redesign the system whenever a new data type was introduced. However, a generic data model is also an important component in enabling remote data access to data. The WIS data model and database API as detailed in previous sections is used to form the core of a distributed GIS. As suggested in section 2.3, both Data Centres and scientists would like the ability to browse and retrieve data remotely via the Internet. Up until recently computing technology has not been able to allow the development of such systems. However, changes in this situation will soon mean connecting to the Internet will be as common as using the telephone, resulting in a web browser on virtually every desktop computer. It enables simple communications between millions of people throughout the world from a common user interface. The software which will operate on these browsers could be written in the Java programming language (Sun Microsystems, 1997). Java was designed to provide a platform and operating system independent programming environment. Although Java is relatively immature in terms of computer languages, it provides some fundamental advantages over its rivals which can be summarised as follows;



Applications or applets may be written once and executed on any platform, reducing development costs.

Applications or applets may be downloaded on demand from a centrally administered server.

Java provides the advantages associated with object oriented languages.

Java has been designed for communication across the Internet therefore security issues have been properly addressed.

Java removes the programmer from the complexity of pointers and memory management found in languages such as C/C++.

In many cases the actual Java language itself is not the most important development but rather the introduction of the Java Virtual Machine (JVM). Java computing operates in a client/server environment where applets are dynamically downloaded on demand from a server.

Figure 2 illustrates how this computing paradigm can provide a solution for distributing the means of querying a database and subsequently viewing and retrieving data. This methodology can of course be used for the reverse process of submitting data to Data Centres. Imagine the following situation; a user goes to his client terminal, it could be a PC, Unix Workstation or Network computer, and connects to the Data Centre Web page. From this page

the user indicates that they would like to browse the database. The Web server deals with this request by sending the appropriate Java applet to the client machine which loads and runs the applet in memory. For a LOIS scientist, the applet might enable the user to formulate questions to the database. The questions are first sent to the Web server where the Java applet makes a database API call. The database API calls manage the connection, the formulation and execution of any user queries. Internally the database API calls produce SQL queries which execute on the database server. Any result produced by an API call is then sent back to the user in either textual, data or graphical form or as a direct data input stream for a Java applet already running on the client machine. As far as the user is concerned, any connections are established directly between the user and the database server as indicated by the loop drawn of figure 2. However, communication between the user and the database is managed transparently by the Web server which supplies Java Applets or Web pages on demand.

Java applets can be as simple or a sophisticated as desired. For the LOIS programme it is hoped that they will enable the users to query the database with the aid of simple maps and have any results presented as reports or graphs. The results of queries should also be available as a simple

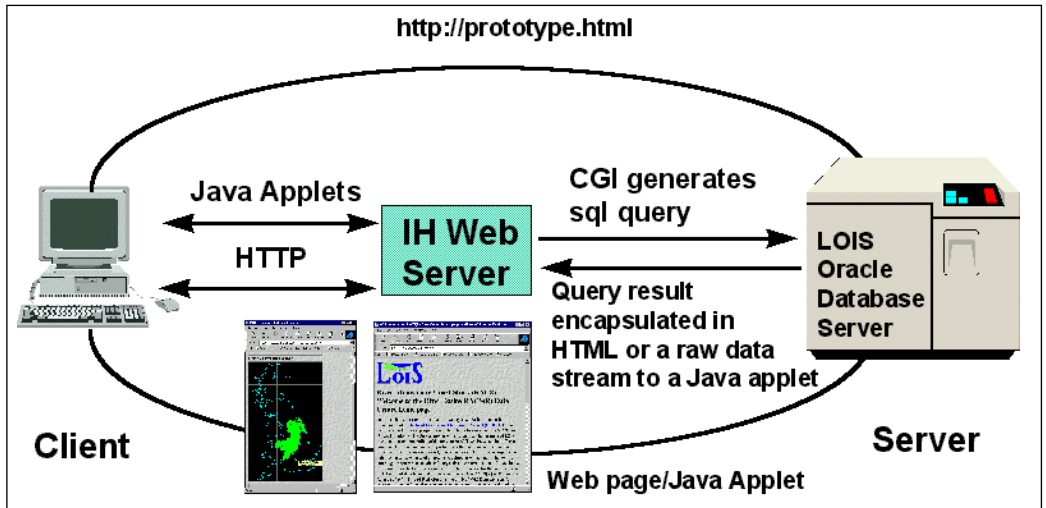


Figure 2. Distributing GIS and Data Using the Internet and Java.



export file that can then be imported, for example, into a spreadsheet for analysis. The Java applets would provide the basic GIS functions such as pan, zoom, scale and re-projections. More complex GIS functionality such a river climbing and catchment boundary derivation could be coded in Java or Common Gateway Interface scripts depending on the processing requirements. However, access by this means is currently only feasible if the number of different database systems that must be queried can be kept to a sensible minimum, ideally one. Hence, the desire for a single all purpose data model.

5.2 Advantages and disadvantages of distributing GIS and data

Distributing GIS capability and data has many advantages for both the user and the Data Centres. Obvious benefits for the user include a common single interface to a large comprehensive data source. Users would also have the ability to express spatial and time series queries from a map based user interface and be presented with the option of downloading the results for further analysis.

Java operates in a client/server environment enabling system developers to determine where processing is undertaken. For example, Data Centres do not want the computing overhead of executing and maintaining the display of the clients user interface. Java allows the applet to be downloaded onto the client machine and executed on their local processor. The only processing carried out by the Data Centre is the preparation and execution of the user's query.

Java is a relatively young language and many of the techniques described in this paper have yet to be tested. Much of the Java work, is however, in the initial stages of design and prototyping. Problems have occurred when attempting to establish large data stream connections with remote database servers. Many scientists have expressed concerns about the security of their intellectual property rights with regards to their datasets. Java does have a security model which has been designed for Internet use. However it is still unknown exactly how secure this model is and comprehensive tests will need to be undertaken

before releasing any system to the public.

6.0 Conclusions

The WIS data model described in this paper has illustrated that it is possible to combine many diverse spatial and temporal datasets within one physical database and thus facilitate the exploration of relationships that span different data types. However, what is now required is an API that allows modellers to interact easily with the database. To achieve this an object orientated approach is being adopted that represents the data as composing of three object types, a database, dataset and cube cells. The data values are the properties of these objects and associated methods allow their manipulation.

The paper has attempted to provide an insight into the future developments of environmental information systems and the way in which the Internet will influence the design of such systems. Java and other associated Internet technologies have provided system developers with a rich set of tools and protocols for developing distributed systems. However, the success of Java may not be entirely due to the language itself but the introduction of the JVM. There have, however, been suggestions from a leading hardware vendor of developing a Universal Virtual Machine which would be capable of producing byte code from any of the main stream computing languages such a C/C++ or Visual Basic. Should this come about then the need for Java could evaporate.

Distributing simple GIS capabilities via the Internet has many advantages for users and Data Centres. Firstly, the task of browsing and retrieving data from the database becomes the responsibility of the user. Users may also download the results of their queries at their convenience, for further analysis. By moving the onus for browsing and retrieving data onto the user, Data Centres then become free to investigate other problems such as quality control, quality assurance, security and visualisation techniques.

7.0 References

Browne, T. J. (1995) The Role of Geographical Information Systems in Hydrology. *Sediment and Water Quality*





in *River Catchments*, UK, 1995, John Wiley & Sons Ltd, Pages 33 - 48.

Hill, D. R. and Bellamy, S. P. (1996) Search mechanisms for querying the time dimension in 4-D GIS. *1st International Conference on GeoComputation*, Leeds, UK, 1996, University of Leeds, Vol 1, Pages 405 - 420.

Moore, R.V. (1997). The logical and physical design of The Land Ocean Interaction Study database. *The Science of the Total Environment*, UK, 1997, Elsevier Science, Pages 137 - 146.

Moore, R.V. and Tindall, C. I. (1992) What is WIS? *IH/ICL Report*, Wallingford, UK

Natural Environmental Research Council (1992) Land-Ocean Interaction Study (LOIS) Science Plan for Community Research Project. *NERC*, Swindon, UK.

Natural Environmental Research Council (1994) Land-Ocean Interaction Study (LOIS) Implementation Plan for a Community Research Project. *NERC*, Swindon, UK.

Sun Microsystems (1997) *The Source for Java*, 1997, Sun Microsystems Inc, <http://java.sun.com/>

Tindall, C. I. and Moore, R.V. (1997) The Rivers Database and the overall data management for the Land Ocean Interaction Study programme. *The Science of the Total Environment*, UK, 1997, Elsevier Science, Pages 129 - 135.

