HAGIS Project

Draft Functional Requirements Specification

Please return to
Nick Bonvoisin

INSTITUTE of HYDROLOGY

INSTITUTE of HYDROLOGY

The **Institute of Hydrology** is a component establishment of the UK Natural Environment Research Council, grant-aided from Government by the Department of Education and Science. For over 20 years the Institute has been at the forefront of research exploration of hydrological systems within complete catchment areas and into the physical processes by which rain or snow is transformed into flow in rivers. Applied studies, undertaken both in the UK and overseas, ensures that research activities are closely related to practical needs and that newly developed methods and instruments are tested for a wide range of environmental conditions.

The Institute, based at Wallingford, employs 140 staff, some 100 of whom are graduates. Staff structure is multidisciplinary involving physicists, geographers, geologists, computer scientists, mathematicians, chemists, environmental scientists, soil scientists and botanists. Research departments include catchment research, remote sensing, instrumentation, data processing, mathematical modelling, hydrogeology, hydrochemistry, soil hydrology, evaporation flux studies, vegetation-atmospheric interactions, flood and low-flow predictions, catchment response and engineering hydrology.

The budget of the Institute comprises £4.5 million per year. About 50 percent relates to research programmes funded directly by the Natural Environment Research Council. Extensive commissioned research is also carried out on behalf of government departments (both UK and overseas), various international agencies, environmental organisations and private sector clients. The Institute is also responsible for nationally archived hydrological data and for publishing annually HYDROLOGICAL DATA: UNITED KINGDOM.

A Geographical Information System for the Institute of Hydrology

HAGIS Project (T12004D2)

Functional Requirements Specification

Draft document June 7, 1988

Nick Bonvoisin, Roger Moore and David Morris

Preface

Over the past few years at the Institute of Hydrology (IH), a large and expanding quantity of spatially referenced data has been amassed. Further, a variety of techniques for the handling of such data have been developed. It would now appear appropriate to bring together these elements to produce a unified system, a Geographical Information System (GIS), and a project has been initiated to bring this about.

Initially the GIS will be used as a tool within IH, but the design will allow for its transfer to the water industry, to other Natural Environment Research Council (NERC) institutes, and to the environmental sector as a whole.

This document describes the essential features, performances, design constraints, attributes and external interfaces of the proposed GIS, provisionally named HAGIS (a Hydrologically Appropriate GIS). Once this document has been approved, the more detailed internal description (the System Requirements Specification) can be written, describing how this design is to be implemented.

Table of Contents

Appendices

1. Introduction

1.1. Background

Maps are a prime source of data in hydrological studies, yet until recently it has only been possible to extract a tiny fraction of the potential information they contain. Consequently, although great progress has been made in the analysis of time-series data, the spatial dimension of hydrological problems is still treated in an extremely simplistic way; this situation is being transformed by the conversion of maps to digital form.

Digital cartography allows the user to produce maps showing only the area and detail required, at a scale and in a style appropriate to a particular application. However, geographical information systems offer far more than just map production; they allow the numerical analysis of all the information within the map whether it be implicitly or explicitly shown.

Consider the various examples of hydrological and environmental problems listed below.

hydrological modelling

vulnerability of groundwater to pollution

production of river quality maps

determination of consent conditions for effluent discharges

asset management

pollution control

fisheries   production of fish maps

recreation and amenity

sea defences

erosion potential

biological applications

ecological applications

It can be seen that they have much in common; all involve both spatial and time-series information, with a need to present and manipulate both types of data.

As an example, consider a river pollution control officer who needs to rapidly identify the cause of pollution so as to prevent or restrict further discharges. At present this can be a slow manual process or one which relies on the officer knowing likely sources; tracing up-stream of a sampling point to all possible sources in a complex river network is both slow and error-prone. Given a database of the potential sources of pollution, combined with river network and terrain information, the officer would be able to apply his or her expertise more efficiently. Not only would point discharges be identified more quickly and reliably, but also dispersed pollution over a land surface could be identified using the terrain information. A GIS has the potential to provide such facilities.

However, existing GIS are not able to meet these requirements because:

generally they are derived from automated cartographical and drawing office systems and are designed to cope with static patterns in maps. Only very limited facilities for attribute data are available, and systems are not designed to cater for time-series data;

there is rarely an off-the-shelf system which will meet all of a user's requirements, and there is certainly no GIS which will satisfy all the environmental problems listed above. It is extremely expensive to customise existing systems; and

when confronted with real datasets of the size involved in tackling many environmental problems, existing GIS are very slow.

The lack of an appropriate system represents a serious impediment to research, therefore it is intended to design and implement a GIS at IH. This document defines the user's view of the proposed system, illustrated in figure 1, the objective of which is to realise the potential of spatially referenced data. This will provide a basis for myriad applications: the system will meet some standard user needs directly, and will include a subroutine library interface to help programmers satisfy other individual requirements.

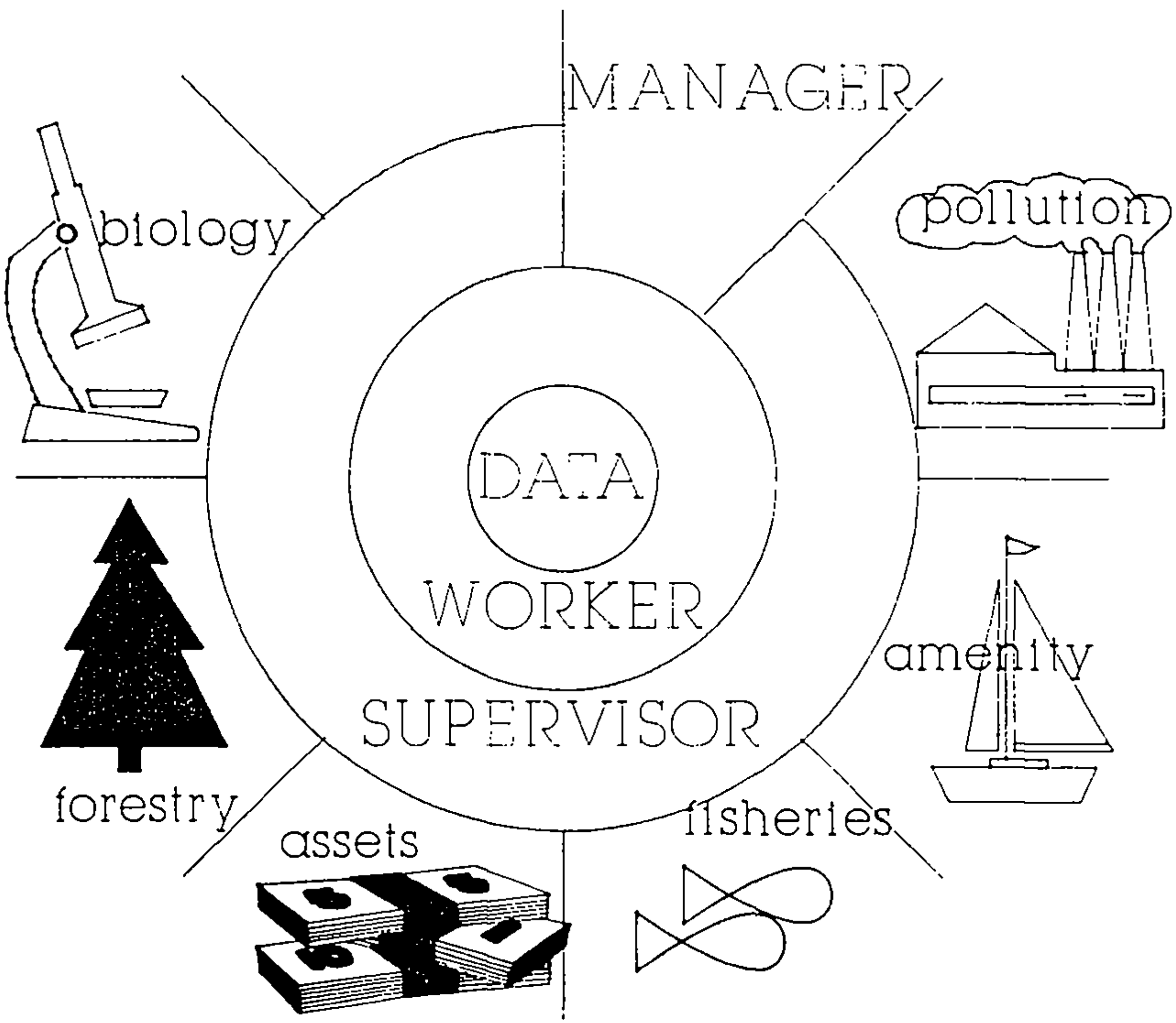
The system will handle a wide variety of data, both spatial and time-series. The main aims in the system design are generality, simplicity, adaptability, efficiency and, very importantly, speed.

1.2. Procedure

This document describes the proposed system's environment, function, performance, logistics, cost and schedule of development. It is not a description of how it works, but of its behaviour.

Throughout the specification various levels of implementation will be addressed, these being the requirements of:

Figure 1. HAGIS system



IH,
the water industry,
NERC, and
the environmental sector.

Further, three groupings of hardware are to be considered to take account of the limitations of some computer system types. These implementation groupings are:

microcomputers,
minicomputers, and
mainframes.

2. System description

The objective of the system is to permit the capture, update, validation, storage, retrieval and analysis of spatial and time-series data. The general design is modular with each module being a collection of procedures performing particular functions. At the core will be the data and its attendant software. When a user accesses the system it will be through the 'Supervisor Module' which simply invokes whichever application the user requires (figure 2). The individual applications access the data via the 'Worker Module', the function of which is to handle update, retrieval and display of data, and to protect the database from corruption. There will be one further module necessary for all implementations, the 'Manager Module', which will provide a variety of 'housekeeping' functions. Once the three central modules have been implemented, additional applications modules will be added to meet specific requirements as they arise. The aim is a friendly, interactive, multi-user system. The complexity of some spatial operations requires the greatest efficiency in handling huge volumes of data, so certain flexibilities may be sacrificed in the interest of speed. Figure 3 illustrates the equipment which might be used in a stand-alone microcomputer implementation.

2.1. Environment

2.1.1. Users

The most important aspect of the system's environment is the people who use it. There are essentially four types of user to be considered:

a naive user who requires an easy to use, friendly and robust system. Such a user will not be assumed to have any knowledge of the mechanisms involved;

a user adding data to the database who wants a similarly friendly interface, but who will be expected to answer questions on the basis of some understanding of the system, for example supplying field widths for attributes;

a programmer, using the subroutine libraries to write new applications modules;

the system administrator (SA), responsible for initiating the functions of the Manager module, for example granting user identifiers and backing up data.

There is no reason why a single person would not fulfil all these roles, for example the user of a stand-alone microcomputer implementation.

All text displayed or printed by the system will be held in

Figure 2. Proposed system design

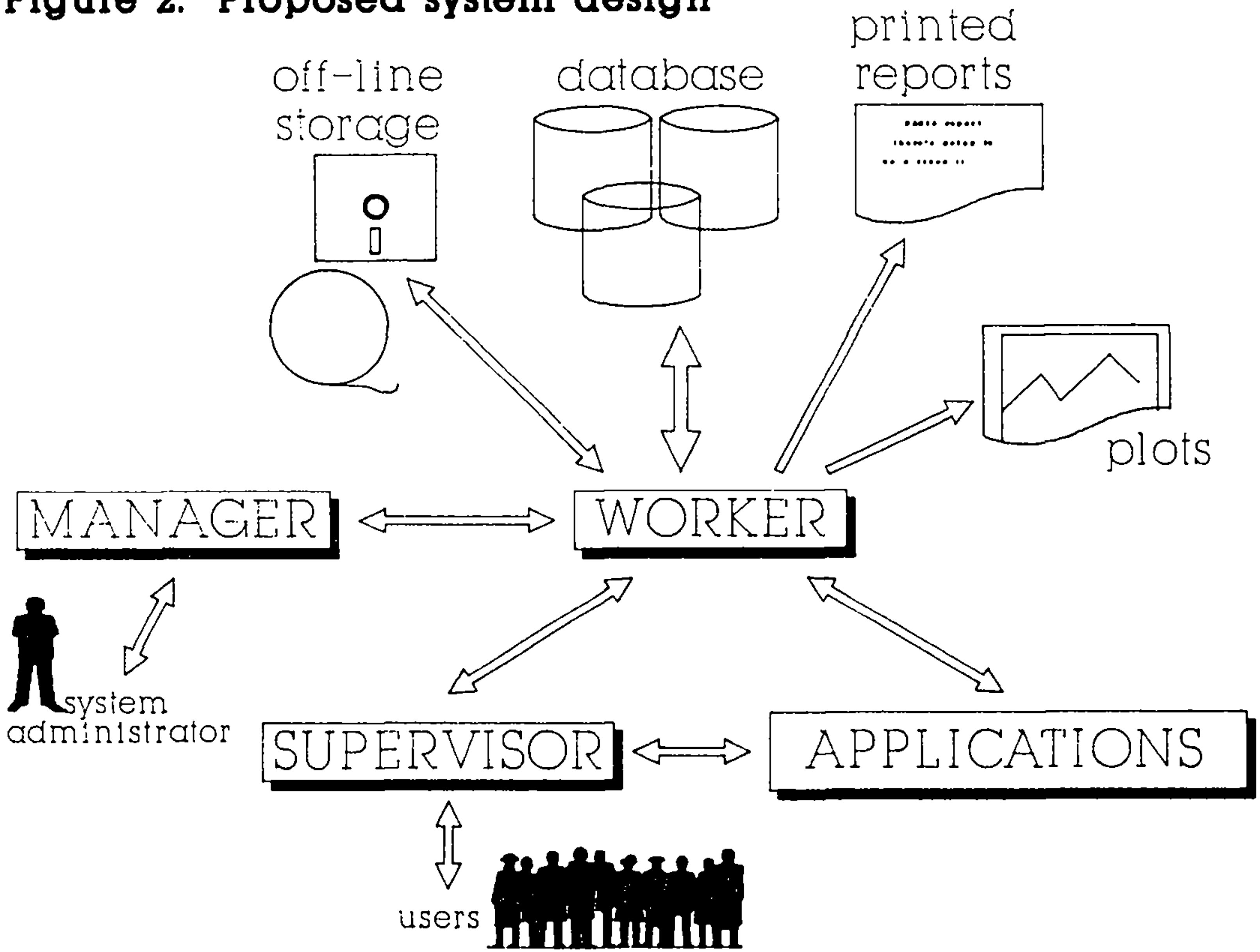
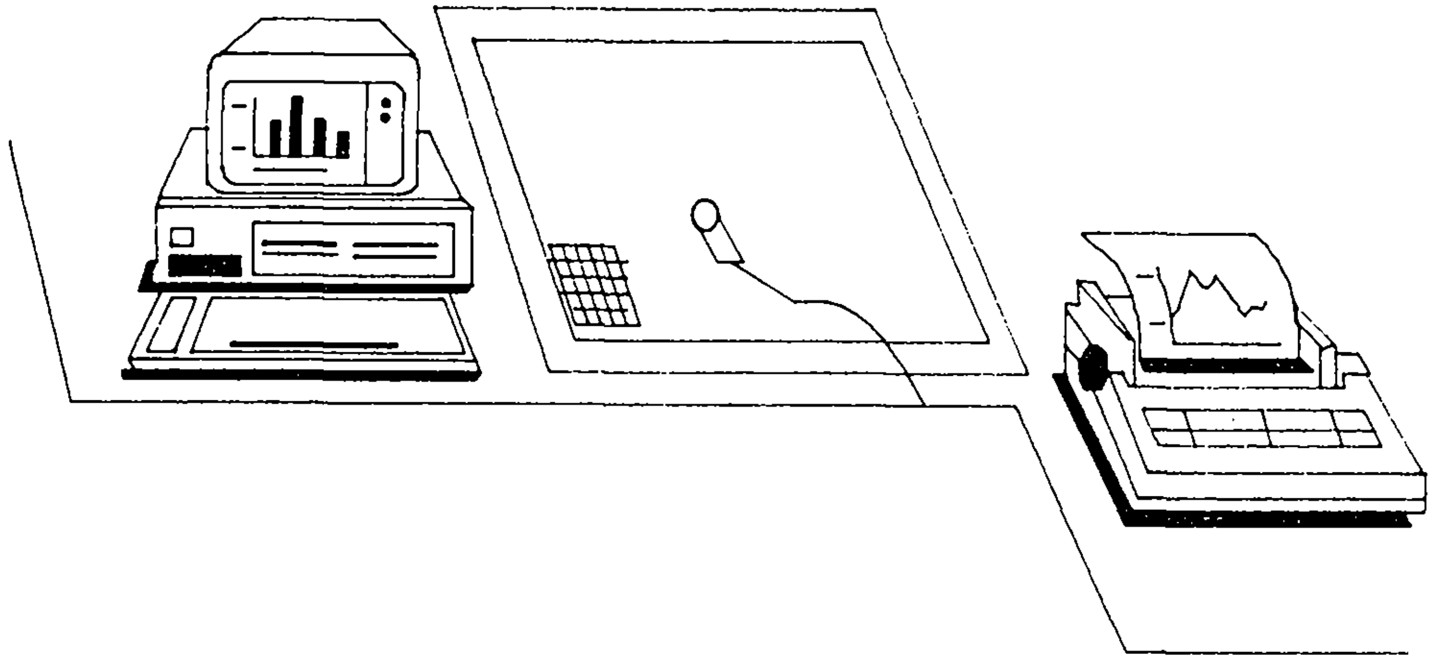


Figure 3. HAGIS hardware



readable files so that conversion to a foreign language is straightforward.

2.1.2. Hardware

Different users have different needs and resources. To cater for this the system is intended to run on microcomputers, minicomputers and mainframes. To make the system 'portable' in this way, the hardware and operating system dependent aspects of the software will be minimised and isolated, so that transfer between machines is not too burdensome.

2.1.3. Software

Writing new software is expensive so the package will be a combination of existing libraries together with purpose-written software, coded in C and/or FORTRAN. The software components required include a graphics library, a database package, and spatial and statistical manipulation routines. Full details of the software options are given in Appendix A.

2.1.4. Data

The data environment, i.e. the data held in the system, will vary according to the organisation using the system. For example, at IH datasets will include DTMs, river networks, soil maps (and tables of map units and associations), geology maps, gauging station attributes (including site photographs), catchment boundaries, a variety of climatological maps (particularly rainfall), a gazetteer and a map library. Other users will require different data to be available; the system will be written in general terms and the user will describe, in a dictionary, the data to be stored.

2.1.5. Other systems

Interchange of information with other existing systems will be handled by a module which will have facilities for reading and writing recognised interchange formats.

2.2. System functions

The central module of the system is the Worker, containing a graphics interface and high- and low-level system libraries; applications modules will not be allowed to use the low-level routines, so as to protect the data. (The only module using the low-level routines will be the Manager which will have unrestricted access to the data.) The Worker, Supervisor and Manager modules will be mandatory; all other modules will be optional, the user being able to select from the standard set or commission further modules. Further, programmers will be able to write their own applications modules using the high-level and graphics libraries. The initial set of modules envisaged is as follows:

- Worker
- Supervisor
- Manager
- Bulk Input/Output
- Interactive Editor
- Mapping
- Manipulation

These modules are described in section 3. Two further applications modules, Water Quality Statistics and River Quality Mapping, will also be written but are not described in this document.

A functional requirements outline is shown in Appendix B.

2.3. User interface

The user interface will be similar on all implementations and will cater for both the expert and inexpert user; the system will be menu-driven, but with the facility for entering more complex instructions in a command line. The principles for development of the user interface are set out in Appendix C. comprehensive help system will be available at all times. Accompanying the system will be User's and Programmer's Manuals, though it is intended that the ordinary user will rarely need to refer to any such documentation. The full range of documentation is described in section 5.

Instructions entered in the command line will be carefully checked, and clear and informative error messages displayed if necessary. The command lines will be stored during a user's session, so that previous instructions can be reviewed, edited and re-used, and so that the resultant collection of commands can be used as input in batch mode (described later). Menu choices will also be interpreted and added to the list of commands. The effect of a stored sequence of commands will often be reversible using an 'undo' command.

When data are to be entered into the system as text, typed at a terminal or microcomputer, the format for entry will be simple, robust and rapid. Data already held in a digital form will be loaded easily and quickly, but with much checking and many optional demands on the supplier for details of accuracy, precision, coverage, documentation, references, copyright, method and date of collection, definitions, ownership, restrictions on access and other measures of quality. The data supplier will be permitted to avoid supplying these details (e.g. for temporary datasets), but the quality of the data held must be known to the system if misuse of data is to be minimised.

The hardcopy output will be graphical and/or textual. Reports will contain as much information as possible to help the user make reasoned judgements on any results. A report will be readable, making use of set expressions (easily translatable into different languages), and the information presented may include:

a description of the quality of data used as input;

details of any documentation, references and copyright;

error analysis: estimates of the accuracy of any values obtained by the combination of different datasets; and

statistical analysis of datasets.

A number of operations for manipulating the data will be made available to the user, along with some other techniques which may be of use such as sorting routines. These will be available to programmers wishing to develop new applications, via the high-level subroutine library. (See Appendix D.)

It is further intended that many operations may be initiated in a batch mode, whereby a command file will be supplied to the system which will then work serially through it executing instructions. In this way the user will not be tied to the terminal or microcomputer for the duration of large, time-consuming or repetitive operations.

2.4. Performance

Most data retrieval and display operations will be able to be performed interactively, though selections on very large areas or involving complex criteria may take some time to respond. The data volumes under consideration are very large, in Northern Ireland for example the 50 m x 50 m digital terrain model (DTM) contains about nine million points (each with four values including elevation and slope direction) and the 1:50,000 river network comprises half a million co-ordinates. It is intended that all testing will involve very large volumes of data so as to check performance under load.

2.5. Security

Some software security on microcomputer implementations may be provided by use of a key disc protection system. In any case an implementation will clearly identify the purchaser of the system so as to aid the tracing of pirate copies to source. Software will be provided with only limited protection against unauthorised access, and this will be in the form of a password and granting permission to use modules and data.

The system will provide facilities for the automatic, or system administrator initiated, restructuring, rationalisation and compression of the database. In the event of accidental corruption of data by users or system failure, the consequent recovery will rescue as much information as possible, while maintaining the integrity of the database.

It is not intended to encrypt the database, and if implemented on a microcomputer the data files will be totally unprotected. The solution for existing microcomputer systems will be the storage of data on removable storage media, use of the physical lock which exists on many processors, or implementation of available software which requires a user to give a password before using the microcomputer. The exact form of security is therefore highly hardware and software dependent, and will vary with different implementations. Ultimately, the protection of data will be the responsibility of the organisation using it, and it is not intended to give full protection against users who have been allowed access to the computer system.

3. Requirements of individual modules

3.1. Supervisor Module

The Supervisor will provide a user-friendly 'front-end' to the system, and will offer the user a menu for choosing which module to invoke. It has the following functions:

Logging on will be achieved by supplying a four character user identifier and an eight character password. When a user logs on a session identifier is allocated so that, if the same user identifier is used concurrently, the different sessions can be distinguished.

Password changes will be made by the user having logged on to the system; a user will only be able to alter his/her own password. A password will be of up to eight characters of which at least six must be non-blank and not include the user identifier.

Module selection will be made by the user choosing which module to run next, having been offered a menu of those available.

3.2. Manager Module

System housekeeping will be provided by the Manager, a separate program which only the system administrator (SA) will be allowed to use; the SA will not have normal GIS access to the database. Note that the SA may well be the sole user of the system, and will therefore have two identifiers, one for system management and one for system use.

Certain of the Manager's functions will be very time-consuming and some will exclude users from the system. The Manager will perform the following functions:

System installation will involve creation of a number of database files, input of the SA's passwords (the SA has two as explained below), setting of a number of variables such as the number of log on attempts allowed, and identification of the hardware being used.

Addition of a module will be achieved by changing the Supervisor so that it is able to offer a new module as an option.

Database creation will be initiated by the SA on behalf of a user. A database will have an owner who will choose who will be able to read from and write to it. The SA may also set up databases owned by the system which will be freely accessible to all with access to the system.

Major error recovery will involve re-creation of indices or retrieving the whole database from a back up copy. Any transactions current at the time of the crash will be 'rolled back' so as to leave no trace.

User identifier allocation will be provided by the SA, who will enter the user's unique identifier and a password. The SA will be able to change any user's password, thus allowing for the occasions when a user forgets the password. Little allowance will be made for the SA's bad memory; IH will keep a routine for recovering the system if the SA's passwords are lost. If more than a set number (chosen by the SA) of unsuccessful attempts are made to log on, then the user identifier will be suspended until released by the SA. A suspended identifier will be listed along with a time stamp and any other information of help. Only the SA's identifier cannot be suspended, but if too many attempts are made then subsequent attempts will require two passwords.

Database integrity checking will be initiated by the SA and will include checks on all pointers within the data.

Garbage collection will be initiated by the SA, either at a convenient time, or when the system has noticed a large number (where this number will be defined by the SA) of deleted records. This data compaction may well exclude use of the system by others.

Performance monitoring will be carried out as a matter of course by the system and reports will be generated when requested by the SA.

Backing up of data will be assumed the responsibility of computer operations staff for mainframe and minicomputer implementations. For microcomputers the SA will be advised to make a back up copy after a pre-selected number of writes, or number of hours of editing. The SA will choose these numbers to suit the user needs. The method used for backing up will be either a time-consuming floppy disc back up (which will certainly prove more costly in the long run), or a tape streamer back-up.

3.3. Bulk Input/Output Module

The Bulk Input/Output Module will provide facilities for the transfer of large volumes of data from external files to the system database, or vice-versa.

Users of this module will be expected to have a higher level of expertise than those just using some module which, for example, plots river quality maps.

Format: The module will be required to cope with one particular transfer format, the National Transfer Format, NTF (Ordnance Survey, 1987), though not all four levels of the standard will be supported initially. The module will also read, and provide as output, a fairly simple format for in-house transfers. This simple format will read in free format where possible, asking the user to identify what is being supplied and how it is to be read.

Quality information: A data supplier will be given the option of entering a wide variety of information to the system before it will accept new data. When data is removed, a report will be generated which will contain this information.

Checking: The data supplier will be able to specify ranges within which values must lie, and to specify what action is to be taken on finding a value out of range.

Conversion: The supplier will be able to supply conversion tables for changing codes from one set of values to another. Further, conversion factors may be applied to data to allow for a change in units.

It is accepted that the loading of data may well be time-consuming if this means that retrievals are made more rapid. The performance of the Bulk Input/Output Module is very much less critical than that of routines retrieving the data for manipulation. Further, the operation of this module may well be carried out in batch mode so slower operations are less relevant.

3.4. Interactive Editor Module

The Interactive Editor Module will provide facilities for the initial input of small amounts of data, and for the correcting of data already resident on the system. Facilities will be made available for both textual and graphical input. Textual input and editing will be provided by easy-to-use forms involving the minimum of typing, and the maximum flexibility. Graphical editing will be implemented using a digitising table, an A0 (1.2 m x 0.8 m) size one being the most suitable for conventional maps.

The interactive operations will include the following:

Insert will allow for the insertion of new features and will contain provision for digitising their position, and typing in attributes and new samples.

Amend will allow for the alteration of a feature's position and attributes (including insertion, deletion and amendment of attributes).

Delete will allow for the removal of positional data, attributes, samples or all the data for a feature.

Select will provide limited criteria for selecting data about a set of features or samples, or for selecting specified items of data for a set of features or samples.

Display will plot selected data as a map on the screen if positional data have been selected.

List will list the selected data on the screen.

Plot will do the same as *display* but provide output to a plotter instead.

Print will do the same as *list* but provide output to a printer.

Note that if a feature has relationships with other features, the user may be required to clarify a command. For example, if a field boundary coincides with a parish boundary and the line is only recorded once, the removal of the parish boundary will not alter the shape of the field.

The retrieval of information for editing will be rapid; update of data will not be as quick, provided the person editing the data is not kept waiting.

3.5. Mapping Module

At present, a mapping package is being developed to meet IH mainframe requirements, by accessing existing data structures. The HAGIS Mapping Module will be able to exploit fully the effort that is going into this package.

This Module will produce a wide range of output, from simple on screen display of data to complex and polished map generation. The Module will include a symbol library to which users may add new symbols.

Maps produced may include some or all of the following features:

map title,
date of production,
credit to producer,
credit to HAGIS system,
details of copyright,
description of data used,
scale (chosen by user) information,
grid lines,
full colour,
vector and raster datasets,
scale related symbols,
symbols proportional to data,
orientated symbols, and
symbol key.

Maps displayed on a monitor will be readily manipulated, enhanced and interrogated; only the display will be edited, not the corresponding data in the system. Having established the format using the interactive display, a user will be able to request either a simple hardcopy or a full map of the type described above.

The Mapping Module will be rapid, within the speed restrictions of the hardcopy device in use, as data is only being retrieved in common with the applications packages.

3.6. Manipulation Module

This Module will provide the user with tools for transforming the database. As research progresses further tools will be added, but the initial list will include:

- node matching,
- network creation,
- polygon formation,
- scan-line conversion of polygons,
- co-ordinate system transformation,
- rubber-sheeting,
- gridding, and
- catchment derivation.

Because many of the operations of this Module will update the information in the database, its operation will not be as fast as a 'retrieval only' module, such as Mapping. Therefore, some procedures will be biased towards use in batch.

3.7. Worker Module

The Worker module will perform all access and update of the database and provide graphical facilities. Most of the graphical facilities will be provided by the chosen graphics library. (Note also that some of the manipulation functions are provided by some graphics packages.) The functions required of the Worker are listed in Appendix D.

The Worker will comprise three main subroutine libraries: graphics, high-level system and low-level system; system libraries may include existing libraries as described in Appendix A. The high-level system library will operate on objects of interest to the user, whereas the low-level one will deal with records, files and indexes.

Applications will communicate with the database via the high-level system library, the Worker being the only module with direct access to the database itself.

Before an application can make use of the database, and the functions of the Worker, the user must have logged on to the system successfully through the Supervisor.

High-level database functions will invoke low-level routines which will not only perform the required operations, but also be responsible for:

Consistency will be achieved by changes to the data being seen immediately they are complete.

Avoidance of contention and deadlock will be provided by maintaining a 'dependency matrix' showing which user sessions are waiting for other user sessions to release locks on files. Deadlock will be detected if a lock is requested that would generate a cycle; a 'victim' user session is chosen and thrown off the system.

Transaction processing will be supported by allowing transactions to be committed or rolled back. A transaction log will record each update, insertion and deletion (showing old and new values), along with a session identifier.

Performance monitoring will be done in a number of functions, thus assisting the Manager in monitoring database usage, though this will not be so excessive as to affect performance.

Off-lining of data will be assisted by the system. Consider, for example, the situation where soil polygons have been converted to raster format. It would not be sensible to clutter on-line disc space with this data; instead the data will be held off-line. However, it is useful if the data can be found and re-loaded easily, so the system will monitor the archiving of data. Similarly, on a microcomputer system, different datasets may be held on different floppy or optical discs, so the system will tell the user which disc to insert next.

Data security will be provided by the user who originally input the data being the only one allowed use of the data, unless explicit permission is given by that user for others to access it. The user will maintain two lists of users (or a code meaning all users) allowed access to each dataset owned. The first list will identify users allowed to read the data (which implicitly includes the user), and the second identifies users with write permission, from which a user may choose to exclude him or herself.

4. Data model

The exploitation of spatial data is critically dependent on the way in which they are stored. The expected data volumes are vast and the databases will therefore require good indexing systems to achieve the very fast access times necessary; a range of database designs have been explored at IH.

A data model describes how the real world is to be generalised into a set of objects, each described by a set of items of data, and how these objects relate. The model being used here details at four levels how the data are to be stored:

The *conceptual level* is an abstraction of the real world the user's view. It identifies entities, relationships between entities, and attributes of entities.

The *information level* describes the data structures in terms of lists of data items, i.e. it identifies what is to be recorded for each entity.

The *access level* describes access paths and index structures. (In designing algorithms for efficient retrieval, there is a need to balance strategies for data independence, integrity and security, with tactics for rapid access, ease of update and economy of space.)

The *encoding level* shows how the model is to be implemented, taking account of the specific computer configuration: programming language, machine design and device characteristics.

The access and encoding levels are not covered in this document, but in the System Requirements Specification.

In designing the data model, the objective has been to achieve a system that is:

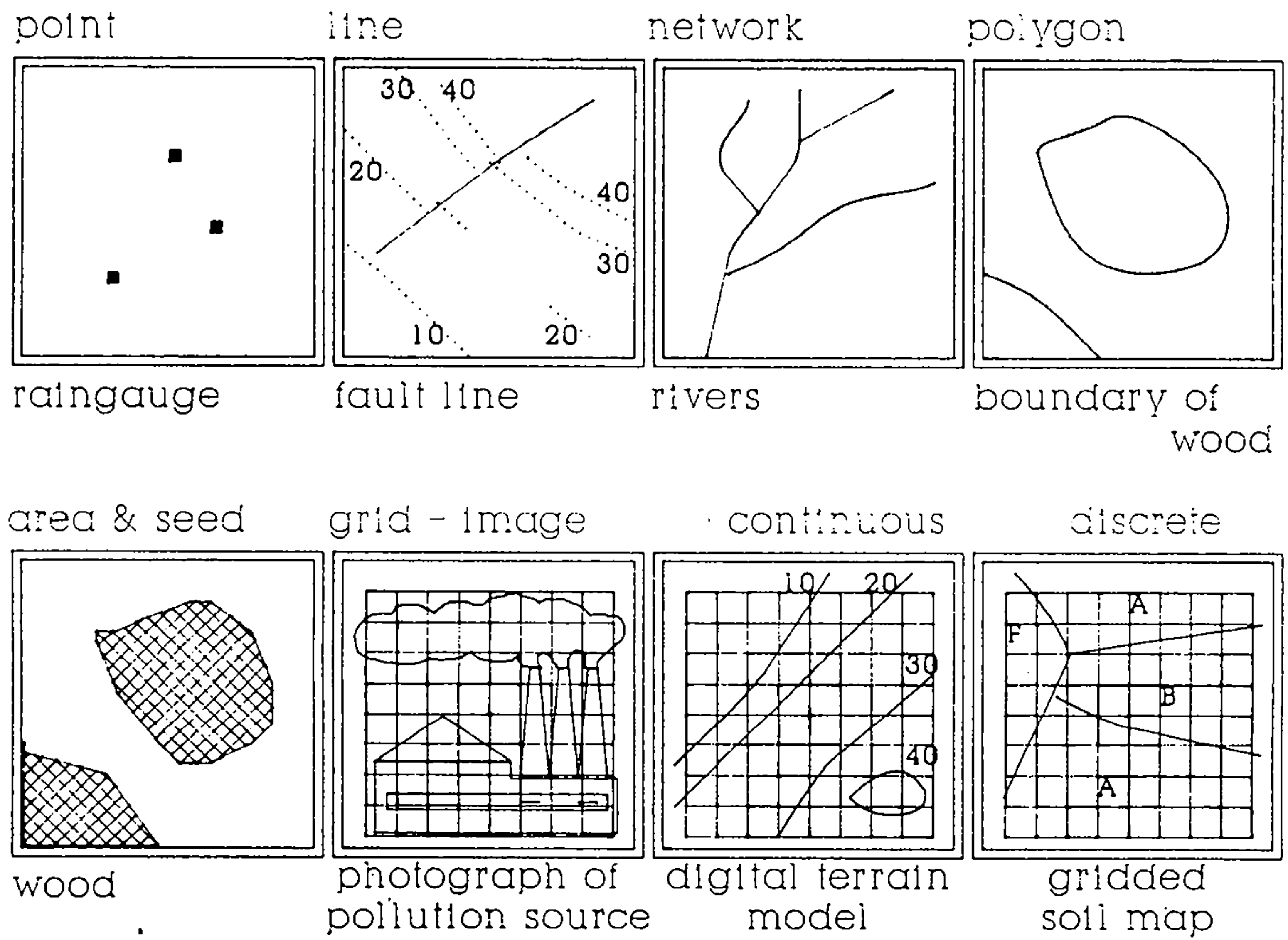
general
simple
adaptable
robust
efficient
fast

4.1. Conceptual level

The data model will handle the data types listed and illustrated in figure 4.

A *feature* is something about which information is to be stored in a database. Features can be related, for example a national park feature comprising three other features: a woodland area, a parkland area and the point location of the park warden's

Figure 4. Data types



office. Another example might be a gauging station which is described as a point feature, and has associated a photograph (an image feature) and a catchment boundary (a polygon). These associations are referred to as the feature's *relationships*. A feature does not have to have spatial data associated with it, for example a licence or consent, and may or may not have a relationship with a spatially described feature.

Features may also have *attributes*, which will generally be considered to be part of a time-series. Thus, a gauging station number may be considered to be an attribute of a gauging station, and would be held along with a date. If the gauging station number is then changed, as sometimes happens, this new number is recorded with the new date.

Thus a feature may have associated with it relationships, and spatial and attribute data, as illustrated by figure 5; the spatial data to which it may refer can be vector or grid (figure 6). The model will allow both grid reference and latitude/longitude co-ordinate systems.

Further, attributes can be associated with a feature at a number of levels in a hierarchical manner. As an example, consider a stream sampling point for which attributes may be recorded at four levels:

(a) the sampling point itself - e.g. grid reference and the name of the owner of the land;
(b) a visit to the sampling point - e.g. date and time, water level and degree of shading from branches overhead;
(c) a sample taken on the visit e.g. time, volume of sample, counts of various species, and fractions of various chemicals; and
(d) a sub-sample from a particular sample - e.g. volume of sub-sample and count of a particularly numerous species.

This hierarchy is supported and extended by the data model as shown in figure 7.

4.2. Information level

The next level down from the conceptual design is the information level, where a collection of tables describing the relationships and spatial and time-series attributes of features (figure 8) are considered. The structure of these tables is detailed in Appendix E which has sections on non-spatial data, spatial data, the data dictionary and implementation. The feature table contains a single entry for every feature on the system, defining a hook on to which spatial and attribute data can be hung.

The *non-spatial* data tables are:

the relationships table stating the association of two features, for example, a gauging station and its photograph;

Figure 5. Overall data structure

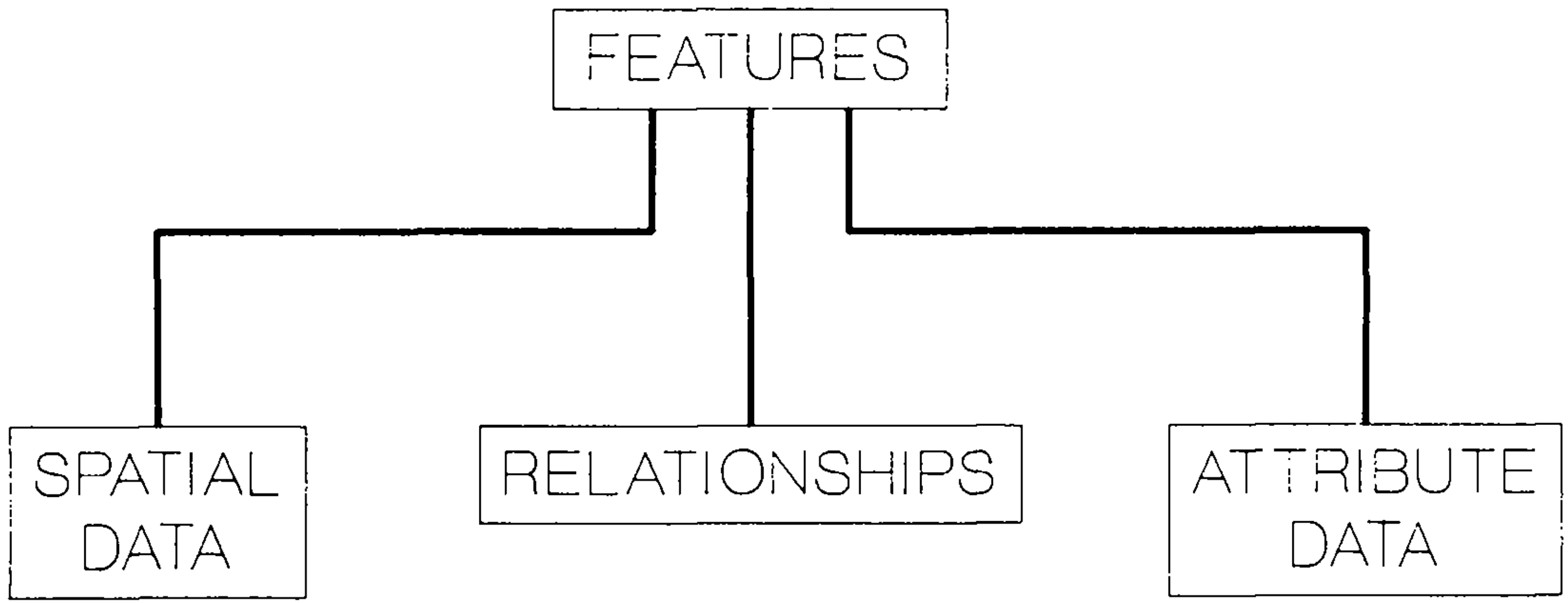


Figure 6. Spatial data structure

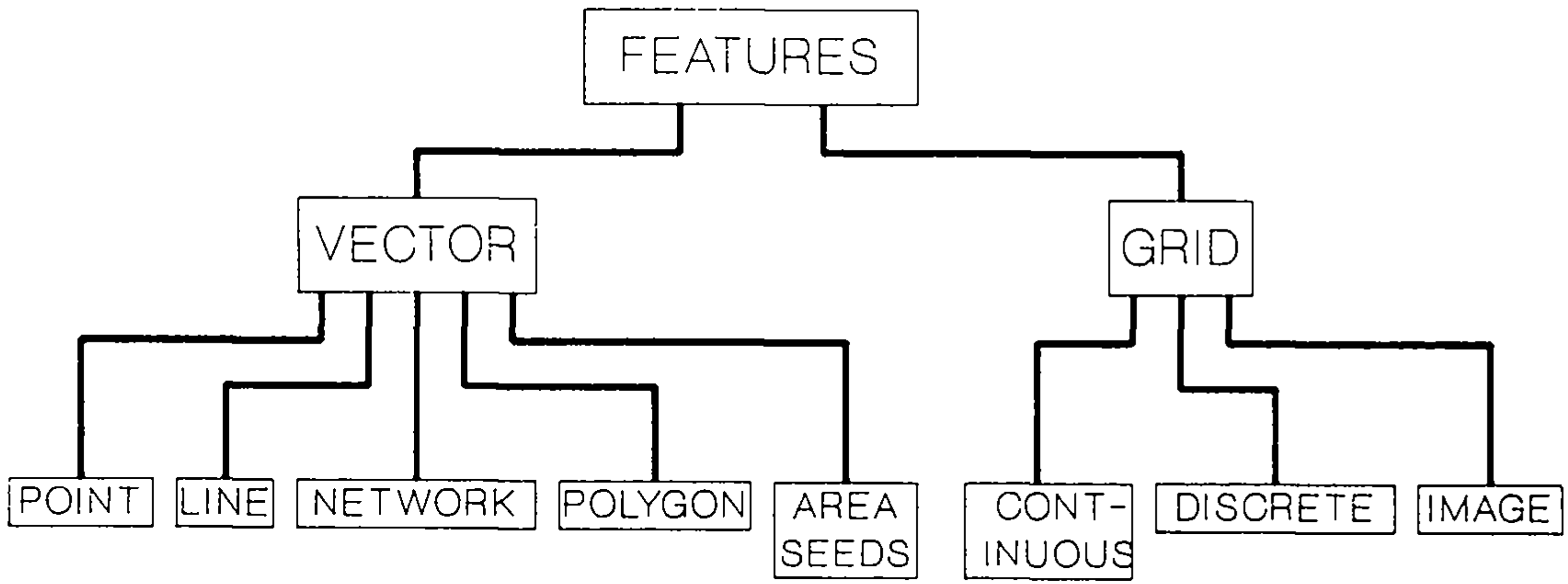


Figure 7. Attribute data structure

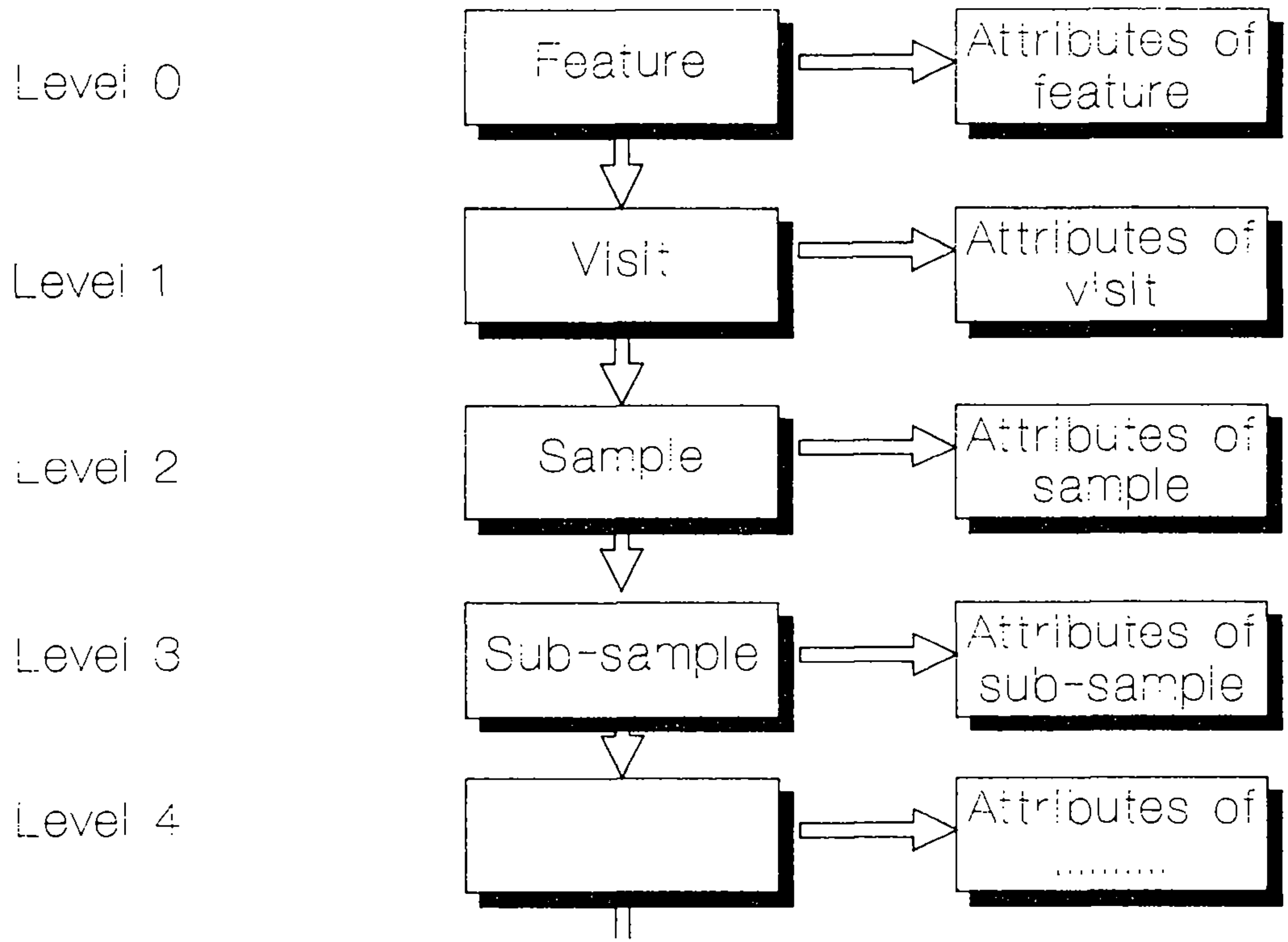
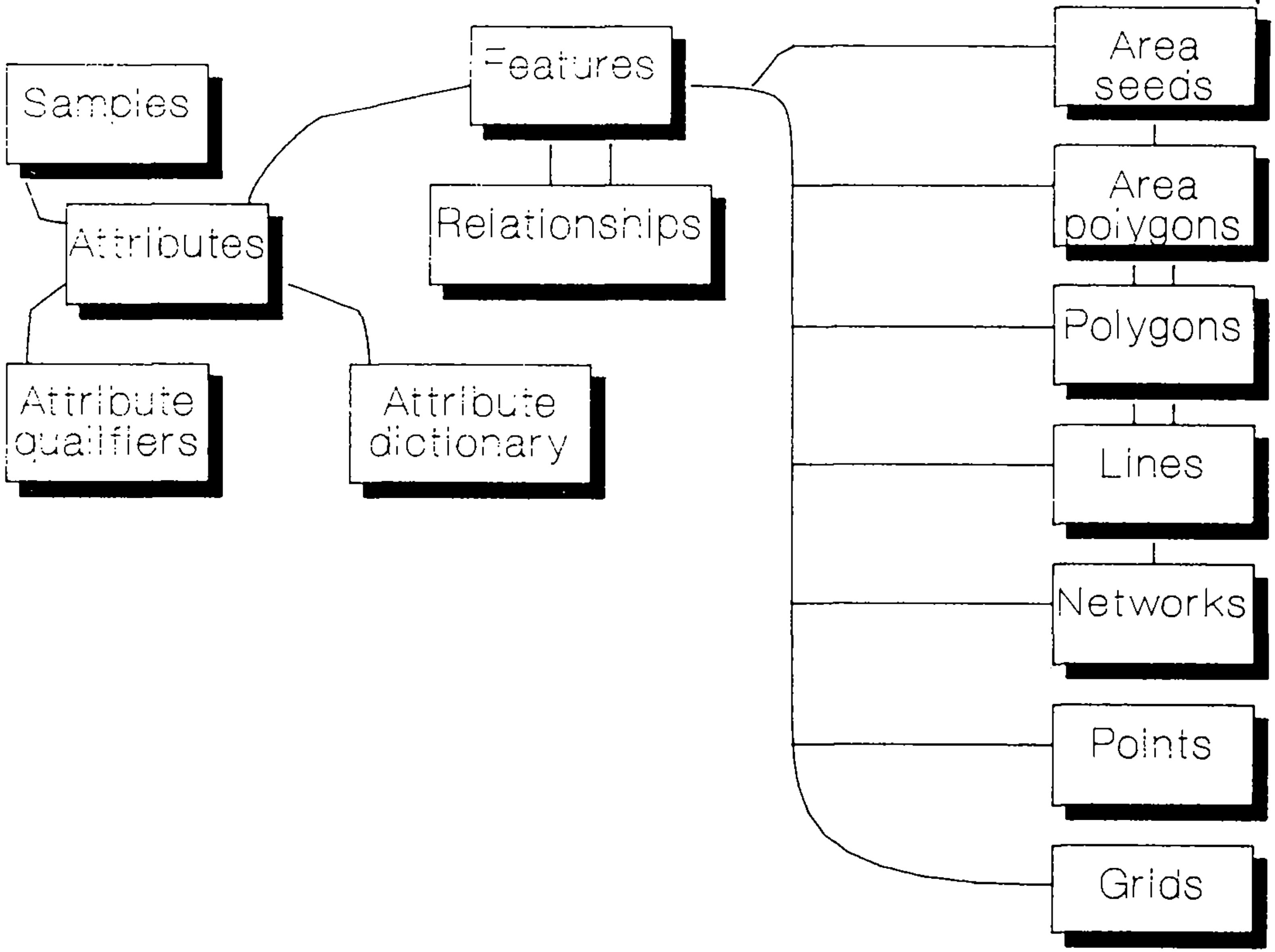


Figure 8. Data tables



the sample table which allows a hierarchy of attribute samples taken at a feature to be given a unique identifier;

the attribute table which contains attribute data for a feature or sample identifier. Values can be stored for a number of attributes types and codes, and many values can be stored for a single attribute code. Further, a qualifying code can be associated with the data at a number of different levels;

the attribute qualifier table which defines the meaning of qualifying codes entered with attribute values; and

the attribute dictionary which defines the meaning of attribute codes and values.

The *spatial data* tables are:

the point table defining the location of a point;

the line table giving the co-ordinates of a line (in either relative or absolute terms), along with an indicator of direction of 'flow' with respect to direction of digitising;

the network table which gives the end two co-ordinates from both ends of a line;

the polygon table giving the sequence of line features comprising a polygon feature;

the area polygon table defining the collection of polygons which give an area feature;

the area seed table which relates a seed to an area; and

the grid table which stores gridded values as a feature.

The *data dictionary* lists the different logical data types to be handled, and the range of values each will be able to take.

In the *implementation* section the true structure of the database tables is revealed, but this need not concern us here.

5. Documentation

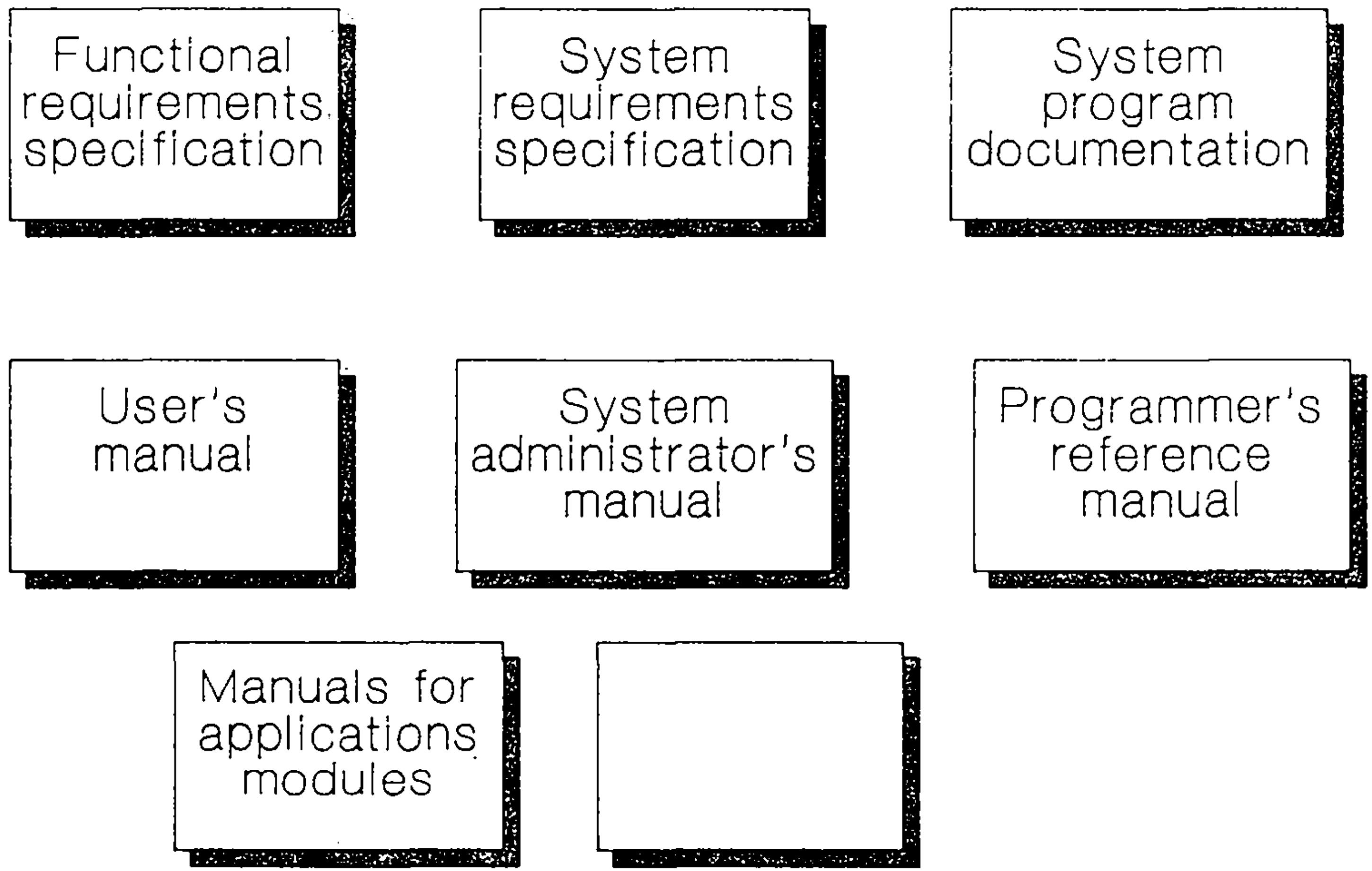
There will be a number of different manuals with the system: one for users, one for the system administrator, one for programmers and further manuals accompanying applications packages. The full range of documentation is illustrated in figure 9. All of the manuals will be structured similarly to the User's Manual as described below.

It is intended that users will look at the User's Manual before using the system for the first time, so it will help the user learn how to operate and exploit the system. It will include the following:

- an introduction which should be read by everyone at least once;
- a principles and procedures section describing the functions and facilities of the system;
- a tutorial which will take the user step by step through examples, and will encourage the user to experiment without risk; and
- a reference section, including explanation of error messages and their avoidance.

All inadequacies, problems, revisions and enhancements will be fully documented and made available as appropriate.

Figure 9. Range of documentation



6. Constraints

6.1. Design and implementation constraints

A variety of restrictions are imposed on the system design by the hardware and software available at IH, and outside. Because the first implementation will be on the IH mainframe, the software will have to be written in Fortran or C. Similarly, the graphics library on the mainframe will have to be an NCS supported one. The system will be transferred on to other computer hardware so code must be portable and will rarely be able to exploit locally available, non-standard features.

Likewise, the hardware facilities for plotting and interactive use on the IH mainframe are fairly restricted, and allowance must be made for equipment with even fewer useful features. (For example, the only interactive input method on the mainframe is the keyboard with cursor keys.) A microcomputer implementation may be restricted by the size of random access memory and by disc storage capacity. Further, hardware is likely to be replaced or enhanced, so the system must be able to survive and exploit such changes.

Another constraint, but one which will enhance this system, i the writing of code in a formal manner, well commented and documented. The standard subroutine style and documentation requirement will be described in detail in the System Requirements Specification.

The life-expectancy of the system is the final design constraint to be considered. The data model and general design should allow the movement, development and enhancement of the system so as to extend its life beyond a five year minimum from the date of implementation.

6.2. Resource constraints

The resources available in terms of time and money must be balanced against the performance required, in respect of the capability and quality of the system.

If the system is to find a market outside IH then it must be available within 18 months, and work must begin in earnest within 6 months to ensure continued interest. To produce a system of the required functionality, a bare minimum of two man years of effort will be needed. The cost to IH may be in the region of:

7. Future development and enhancement

The modular system design will allow the addition of modules with further functions. The libraries of the Worker Module will be expanded as further procedures are developed. Also, if changes in code are identified which improve performance, but do not alter argument lists, then these will be introduced. Obviously, such enhancements will be made immediately within IH, but existing outside users will be expected to purchase enhanced versions at reduced cost.

The data model may be extended to handle true three-dimensional data (and the initial data structures will allow for this enhancement), as graphical and spatial data handling techniques are developed.

Other future improvements may include a spatial data query language, a better relational database structure and a full implementation of NTF input and output.

8. Project implementation plan

The project development plan is illustrated by figure 10, and a task list is given in Appendix F. On completion and approval of this Functional Requirements Specification, the System Requirements Specification will be written, covering:

introduction;

system design - selection of software and hardware
- full database design with details of data types and sizes and indexing system;

process design - functions and options
- input/output and interfaces with other processes
- algorithms and data structures;

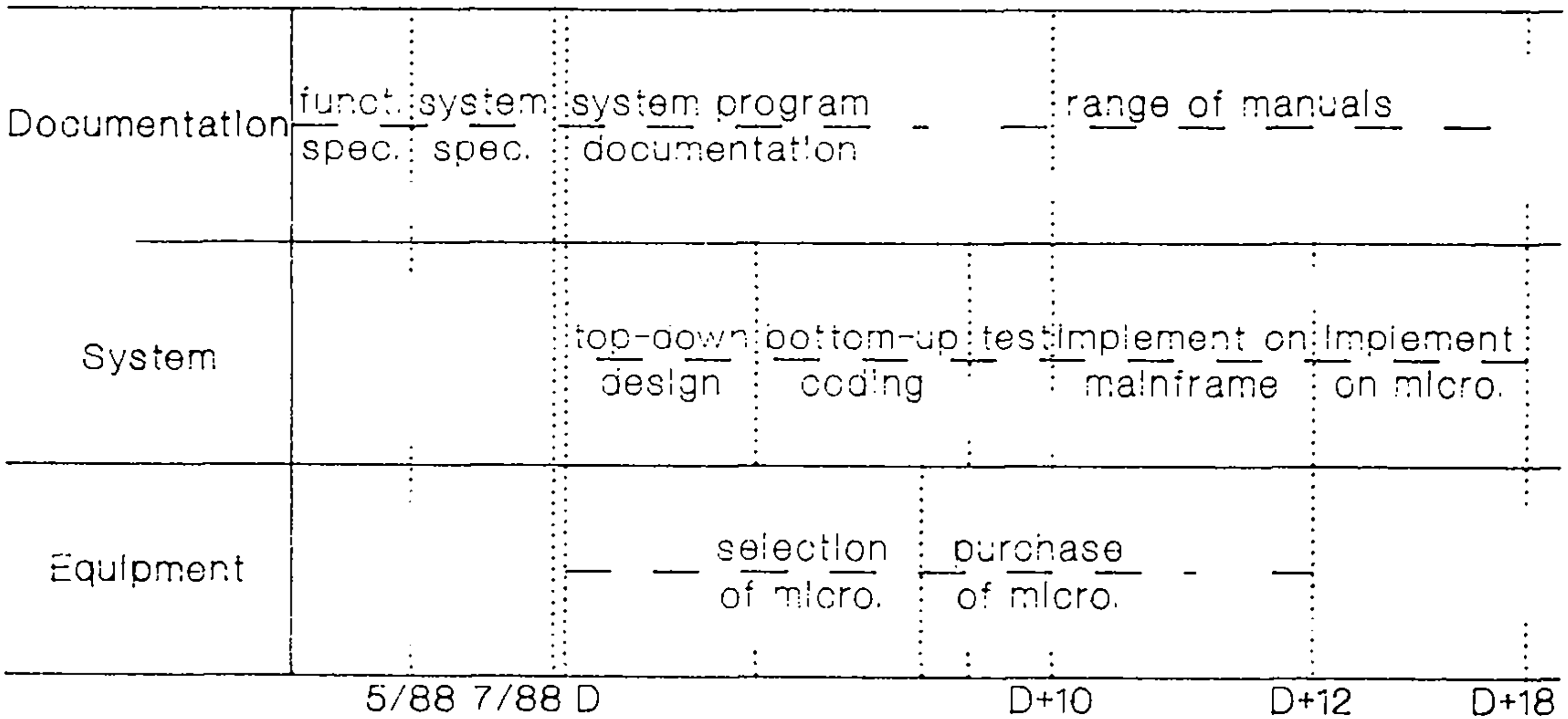
revised and more detailed cost and time estimates; and

revised project implementation plan.

When these two specification documents have been examined and approval given, the program development will commence: first, the Worker and Manager Modules including database implementation will be coded and, second, other central modules as described in section 3. In parallel with this development, the system program documentation will be written.

The system will then be installed on the IH mainframe and the manual writing will begin. At the same time, microcomputer hardware and software will be selected and purchased to facilitate the transfer onto this equipment. Both implementations will involve extensive and thorough testing with substantial quantities of data. The pilot area will be Northern Ireland, for which we already have a DTM, complete river network and a variety of other data. Further, these data have been used extensively so comparisons of the results of analyses can be made.

Figure 10. Project implementation plan



Date (end of month)

D is date of approval of request
(assuming two people working full time D onwards)

References

Chorley, 1987. Report to the Secretary of State for the Environment of the Committee of Enquiry into the Handling of Geographic Information. HMSO, London.

Digital Research Ltd, 1985. GEM Graphics Environment Manager software: programmer's guide. Digital Research, California.

Institute of Hydrology (IH), 1988. Proposal for a Great Ouse database. IH internal document.

ISO, 1985a. Information processing systems - Computer graphics - Graphical Kernel System (GKS) functional description. ISO 7942, Geneva.

ISO, 1985b. Information processing systems - Computer graphics - Graphical Kernel System (GKS) for three dimensions (GKS-3D) functional description. ISO/DP 8805, Geneva.

ISO, 1985c. Information processing systems - Computer graphics - Programmer's Hierarchical Interactive Graphics System. Working Draft ISO/TC97/SC21 N819, Geneva.

Kernighan, B.W. and Ritchie, D.M., 1978. The C Programming Language. Prentice-Hall.

Moore, R.V., Morris, D.G. and Bonvoisin, N.J., 1987. Digital mapping as an aid to hydrological analysis. British Hydrological Society National Hydrology Symposium, University of Hull.

Morris, D.G. and Heerdegen, R.G., 1988. Automatically derived catchment boundaries and channel networks and their hydrological applications. Geomorpholgy, 1(2): 131-141.

Numerical Alogorithms Group (NAG), 1987. The NAG Fortran mini manual - mark 12. NAG Ltd, Oxford.

Oracle, 1985. Product Summary and Technical Documentation. Oracle Corp. UK Ltd., Richmond.

Ordnance Survey, 1987. National Transfer Format. Produced by OS on behalf of the Working Party to Produce National Standards for the Transfer of Digital Map Data.

Sekulin, A.E., 1986. Cache-Cache Manual. IH internal document.

Steward, D.V., 1987. Software Engineering with systems analysis and design.

American National Standards Institute (ANSI), 1978. ANSI standard X3.9-1978 Fortran 77. ANSI, New York.

Abbreviations and acronyms

AONB	Areas of Outstanding Natural Beauty
BGS	British Geological Survey
DBMS	Database Management System
DOE	Department of the Environment
DTM	Digital Terrain Model
GIS	Geographical Information System
GKS	Graphical Kernel System
HAGIS	Hydrologically Appropriate Geographical Information System
IBM	International Business Machines
IH	Institute of Hydrology
ITE	Institute of Terrestial Ecology
JANET	Joint Academic Network
MOD	Ministry of Defence
MSS	Multi-Spectral Scanner
NCS	NERC Computing Service
NERC	Natural Environment Research Council
NTF	National Transfer Format
OS	Ordnance Survey
OSNI	Ordnance Survey of Northern Ireland
SA	System Administrator
SPOT	Systeme Probatoire pour l'Observation de la Terre
SSSI	Site of Special Scientific Interest
TM	Thematic Mapper
WDDES	World Digital Database for Environmental Sciences

Glossary of terms
Largely taken from Lord Chorley's report (1987).

Accuracy	The degree of conformity with a standard, whether absolute or relative. Accuracy relates to the quality of the result, whereas *precision* relates to the quality of the operation by which the result is obtained.
Attribute	A property of an *entity*.
Continuous grid data	A grid of values derived from a continually varying surface, e.g. a *digital terrain model*. Contrast with *discrete grid data*.
Database	An organised, integrated collection of data stored so as to be capable of use by relevant applications with the data being accessed by different logical paths.
Data capture	The encoding of data, e.g. *digitising*.
Data model	How the real world is to be represented in the database.
Dataset	A named collection of logically related *features* arranged in a prescribed manner.
Deadlock	Situation in which a *database* is being accessed by two processes, each waiting for the other to finish using some data. For example, process 1 is reading some data A and wanting to have sole access to data B so as to write to it; it is stopped from accessing B by process 2 which is reading B and waiting to write to A; the resulting *deadlock* is known as a 'deadly embrace'.
Digital terrain model	A digital representation of relief. Often enhanced by the addition of planimetric information.
Digitising	The process of converting analogue maps and other sources to a computer readable form.
Discrete grid data	*Grid data* in which adjacent squares frequently have the same value, e.g. a soil map. Contrast with *continuous grid data*.

Entity	Something about which information is stored in a *database*. The information may consist of relationships, *attributes*, positional and shape information etc.
Feature	Another word for *entity*.
Feature code	An alphanumeric code which is an *attribute* of a *feature* and describes and/or classifies that *feature*.
Grid data	Spatially referenced data held in a rectangular array of cells referenced to a grid.
Garbage collection	Removal of records marked as deleted but still physically resident in a database.
Key disc	Similar in effect to a *dongle*, but instead a special disc must be inserted into the microcomputer before the software will work.
Pixel	One of a regular array of cells in a collection of *raster data*, a picture element.
Precision	The exactness with which a value is expressed, whether it is right or wrong.
Quadtree	A data structure for thematic information in a raster database that seeks to minimise storage.
Resolution	A measure of the ability to detect quantities. High resolution implies a high degree of discrimination, but has no implication as to *accuracy*.
Run-length encoding	A method of compacting *raster data* by recording counts of multiple, consecutive occurrences of a value.
Raster data	Data expressed as an array of *pixels*, with spatial position implicit in the ordering of the pixels.
System administrator	The individual responsible for the original installation of the system and for initiation of certain functions of the Manager Module.
Vector data	Positional data in the form of co-ordinates of the ends of line segments, points, text position etc..

Appendix A. Software options

Purpose-written software

For the software to be written at IH, C (Kernighan and Ritchie, 1978) and FORTRAN are being considered. C's flexibility and generality gives it advantages in database and user-interface design, but there is a great deal of inertia favouring FORTRAN. If FORTRAN were used, then it would be strictly standard FORTRAN 77 (American National Standards Institute, 1978).

Graphics library

The main contenders for a graphics library are:

Graphics Environment Manager, GEM (Digital Research, 1985), a popular two-dimensional graphics library for microcomputers, purchased by IH, and having drivers for a wide variety of hardware;

Graphical Kernel System, GKS (ISO, 1985a), an existing two-dimensional graphics international standard; NERC Computing Service (NCS) have the UNIRAS implementation on the IBM mainframe at IH. GKS is also available for microcomputers, but has not proved popular;

GKS-3D (ISO, 1985b), the three-dimensional implementation of GKS;

Programmer's Hierarchical Interactive Graphics System, PHIGS (ISO, 1985c), an alternative to GKS-3D; and

UNIRAS, a collection of raster graphics libraries. The research councils, universities and polytechnics recently made a mass purchase agreement, and NCS has an implementation on the IH mainframe. Purchase of the IBM microcomputer implementation is now being considered.

Further, the graphics interface will be replaceable, in case more appropriate software become available.

Database software

There exists a variety of options regarding the database software, including:

writing all our own software;

use of a database management system (DBMS) already available at IH, such as the commercially-available Oracle (1985) or home-grown Cache-Cache (Sekulin, 1986) systems; and a system exploiting some existing software to handle the lower level aspects such as searching datasets, for example an indexed-sequential files system.

Other software

The market will be examined to see if there exist libraries o routines to ease the task of writing the data manipulation software, as well as improving it.

Statistical operations may be provided by a further library, and the popular Numerical Algorithms Group software (NAG, 1987) would be the prime candidate. The NAG library is widely available in the academic sector and a large subset of the routines has been implemented on microcomputers.

Most of the time-series functional requirements at IH will be met by existing software.

Appendix B. Functional requirements outline

invoke Supervisor
1. logon
   a) supply user identifier
   b) supply password
   c) session identifier allocated
2. select option
   a) change password
      1> supply old password
      2> supply new password
      3> re-enter new password as check
   b) Bulk Input-Output
      1> select transfer direction
      2> select transfer format
      3> input extra information
      4> specify range checking and action on error
      5> supply conversion tables and factors
      6> print report
   c) Interactive Editor
      1> insert
      2> amend
      3> delete
      4> select
      5> display
      6> list
      7> print
      8> plot
   d) Mapping
      1> select data items
      2> design format
         a: enter map title
         b: choose colour scheme
         c: choose type of grid lines
      3> display on screen
      4> plot
         a: quick dump
         b: quality map
   e) Manipulation
      1> select data items
      2> manipulate
         a: node match
         b: create network
         c: form polygons
         d: scan-line convert polygons
         e: transform co-ordinate systems
         f: rubber sheet
         g: grid
         h: derive catchments
         i: create DTM
   logoff

B. invoke Manager

1. logon
    a) supply user identifier
    b) supply password
    c) supply second password if too many tries at first
    d) session identifier allocated
2. SA is shown state of system
    a) whether a back up should be made
    b) whether data should be compacted
    c) whether there has been a system crash any incomplete transactions
    d) whether any users have been suspended
3. select option
    a) install system
        1> database files created
        2> enter two SA passwords
        3> set number of log on attempts allowed
        4> set no. hours editing before back up warning
        5> set no. database changes before back up warning
        6> set no. deleted records before compaction warning
        7> identify hardware
    b) introduce new module to Supervisor
        1> identify module to be added
        2> choose position in list of Supervisor options
        3> Supervisor altered to offer new module
    c) create new database
        1> identify owner
    d) delete database
    e) recover from major error
    f) allocate user identifier
        1> enter user identifier
        2> enter password
        3> re-enter password as check
    g) remove user identifier from system
        1> list existing users
        2> select user identifier to be removed
        3> confirm removal
    h) release suspended user identifier
        1> list suspended user identifiers
        2> release user from suspension
    i) check database integrity
        1> check pointers in data
    j) compact - garbage collection
        1> check on use of the system
        2> issue warning
        3> compact
    k) monitor performance
        1> display system variables
    l) back up data
        1> identify data to be backed up
        2> identify device to back up to
        3> back up
4. logoff

Appendix C. Principles for developing user friendly interfaces Taken from Steward (1987).

Think of yourself as the program and how you would talk to the people who are using it.

Be polite. Don't put down the user.

Constantly sell yourself to the user.

Picture the user as one who has just returned from vacation and has not used the program in a month.

Think of your user as a partner.

Don't punish.

Check quickly whether you understand the user's response.

Make it easy for the user to change his or her mind.

Allow users to explore without risk and anxiety. Allow them to do and undo without harm. Help them cope.

Do unto users as they would have you do unto to them.

Try to learn the user's habits and adapt to them.

Organise the structure of the interaction to flatter the organisation of the user, not follow the organisation of the program.

Keep it simple.

Be consistent. Don't do the unexpected. Avoid anything that might be confusing or leave your user not knowing how to respond.

Make sure what you ask users to do has meaning to them. Avoid the arbitrary.

If you do the same type of thing again, do it the same way. Keep the rules users must remember and understand to an absolute minimum.

Make simple things simple, then more complicated things more complicated only if they need to be.

Develop a simple model of how the program will behave in all instances.

Provide feedback.

Keep a neat screen.

Keep user's attention.

Given a choice, have them point rather than say.

Appeal to what is already understood.

Don't be boring.

Be concerned about the amount of information the user needs from one screen while looking at another.

Have someone else run the program and observe the problems he or she has.

Appendix D. Worker Module library functions

The following list is neither exhaustive nor definite, but illustrates the range of routines and concepts required of the system.

Spatial data manipulation

polygon area and digitising direction
line intersection
left-right identification
line reduction
network traversal
interpolation
contouring
gridding (in particular DTM creation)
projection (co-ordinate transformation)
rubber-sheeting
node-matching (with split and join)
vector-to-raster conversion (polygon)
raster-to-vector conversion (polygon)
quadtrees
searching:
- point-in-polygon
- line-in-polygon
- radial
- box

buffers
overlay
conversion of line to raster representation and vice versa
gridded polygon boundary identification
catchment derivation from DTM

Time-series

Statistical

regression
standard deviation
mean
median
extremes

Database

insert
delete
update
select

Graphics

mandatory output functions:
- clear screen
- colour
- text
- point
- line
- marker
- area fill
- window
- viewport
- user-specified co-ordinate space
- image storage or exclusive-or plotting

optional output functions:
- multiple viewports
- clipping
- cell array
- line thickness and type
- multiple fonts
- update of colour table
- picture segment transformation
- 2D graph
- bar chart
- circle
- rectangle
- 2D contour map
- choice of viewpoint and perspective for 3D graphics
- 3D graph
- 3D contour map
- generalisation
- non-overlapping labels

mandatory input functions:
- string input
- use of mouse or cursor keys to input co-ordinates

optional input functions:
- data tablet/digitiser input
- event mode input

Other

sorting:
- Heap sort
- Shell sort
- Quick sort

searching:
- binary chop

data encryption

Appendix E. Logical data structure

Within each table is shown a description of the field, the field name and the field data type (as defined in the data dictionary).

Items stored in the feature table

Feature type	FTYPE	FTYPE
Feature identifier	FID	FID
User code (optional)	UCODE	CODE
Feature name (optional)	FNAME	TEXT

A complete list of all features of all types and classes. FTYPE and FID are the minimum information to define the existence of a feature to the system.

Non-spatial data

Items stored in the relationships table

Parent feature type	MTYPE	FTYPE
Parent feature identifier	MID	FID
Child feature type	CTYPE	FTYPE
Child feature identifier	CID	FID
Direction	DIR	DIR
Sequence	SEQ	COUNT

Records non-spatial relationships between features, but see below.

Items stored in the sample table

Feature type	FTYPE	FTYPE
Feature identifier	FID	FID
Sample level	SLEVEL	SLEVEL
Sample number	SN	SN
Parent sample identifier	MSID	SID
Child sample identifier	CSID	SID

Items stored in the attribute table

Feature type	FTYPE	FTYPE
Feature identifier	FID	FID
Time units	TUNIT	TUNIT
Start date/time	SDT	DT
End date/time	EDT	DT
Sample level	SLEVEL	SLEVEL
Sample identifier	SID	SID

Number of repeats	REPEAT	COUNT
Level of repeat	LREP	ALEVEL
Qualifier position	QPOS	ALEVEL
Attribute type	ATTYPE	ATTYPE
Attribute code	ATCODE	ATCODE
Number of values	NVAL	COUNT
Attribute value	ATVAL	ATVAL
Attribute qualifier	ATQUAL	ATQUAL

Items stored in the attribute qualifier table

Attribute type	ATTYPE	ATTYPE
Attribute code	ATCODE	ATCODE
Attribute qualifier	ATQUAL	ATQUAL
Meaning	MEANS	TEXT

Items stored in the attribute dictionary

Attribute type	ATTYPE	ATTYPE
Attribute code	ATCODE	ATCODE
Attribute value	ATVAL	ATVAL
Synonym flag	SYN	SYN
Short description	SDESC	SDESC
Long description	LDESC	LDESC
Definition	DEFN	TEXT
Bibliography	BIBLIO	TEXT
Update date	UDATE	DT
Format	FORMAT	
Specification	SPEC	
Length	LENGTH	
No data flag	NODATA	
Minimum value	MINV	
Maximum value	MAXV	

Spatial data

Items stored in the point table

Point type	XTYPE	FTYPE
Point identifier	XID	FID
Zone	ZONE	ZONE
Unit	UNIT	UNIT
2-D/3-D marker	D2D3	D2D3
X-coord	X	COORD
Y-coord	Y	COORD
Z-coord	Z	COORD

Records the position of point features.

Items stored in the line table

Line type	LTYPE	FTYPE
Line identifier	LID	FID
Zone	ZONE	ZONE
Unit	UNIT	UNIT
2-D/3-D marker	D2D3	D2D3
Direction of flow	FLOW	DIR
Relative/absolute marker	RELABS	RELABS
No. of points	NPTS	COUNT
First X-coord	XF	COORD
First Y-coord	YF	COORD
First Z-coord	ZF	COORD

repeated (NPTS-1 times):

Absolute X-coord	XABS	COORD
Absolute Y-coord	YABS	COORD
Absolute Z-coord	ZABS	COORD

or

Relative X-coord	XREL	RCOORD
Relative Y-coord	YREL	RCOORD
Relative Z-coord	ZREL	RCOORD

Records the position of lines.

Items stored in the network table

Line type	LTYPE	FTYPE
Line identifier	LID	FID
Zone	ZONE	ZONE
Unit	UNIT	UNIT
2-D/3-D marker	D2D3	D2D3
First X-coord	XF	COORD
First Y-coord	YF	COORD
First Z-coord	ZF	COORD
Second X-coord	XFB1	COORD
Second Y-coord	YFB1	COORD
Second Z-coord	ZFB1	COORD
Last but one X-coord	XLB1	COORD
Last but one Y-coord	YLB1	COORD
Last but one Z-coord	ZLB1	COORD
Last X-coord	XL	COORD
Last Y-coord	YL	COORD
Last Z-coord	ZL	COORD
Direction w.r.t. digitising	DIR	DIR

Records the connectivity of a network.

N.B. Every line in the network has two entries in the table, one with the co-ordinates in order with DIR='F' and one with the co-ordinates reversed and DIR='R'.

Items stored in the polygon table

Polygon type	PTYPE	FTYPE
Polygon identifier	PID	FID
Sequence number	SEQ	COUNT
Line type	LTYPE	FTYPE
Line identifier	LID	FID
Direction	DIR	DIR

Records the relationship between a polygon and the lines that make it up together with the order and direction in which they would be joined together.

Items stored in the area polygon table

Area type	ATYPE	FTYPE
Area identifier	AID	FID
Polygon type	PTYPE	FTYPE
Polygon identifier	PID	FID
Inner/outer marker	IOM	DIR

Relates an area to the inner and outer polygons that define its extent. There will be one entry for the outer polygon and as many entries for inner polygons as there are islands within the area.

Items stored in the area seed table

Area type	ATYPE	FTYPE
Area identifier	AID	FID
Zone	ZONE	ZONE
Unit	UNIT	UNIT
2-D/3-D marker	D2D3	D2D3
X-coord	X	COORD
Y-coord	Y	COORD
Z-coord	Z	COORD

Allows attributes to be attached to an area.

Items stored in the grid table

Grid type	GTYPE	FTYPE
Grid identifier	GID	FID
Zone	ZONE	ZONE
Unit	UNIT	UNIT
2-D/3-D marker	D2D3	D2D3
Encoding method	ENCODE	ENCODE
Storage square size	STSIZE	STSIZE
Cell size	CLSIZE	CLSIZE

Minimum X-coord	XMIN	COORD
Minimum Y-coord	YMIN	COORD
Minimum Z-coord	ZMIN	COORD
Raw data	RAW	RAW

Spatial indices

Note that the network table is used as an index the starts and ends table.

Items stored in the squares/cubes table

Zone	ZONE	ZONE
Minimum X	XMIN	COORD
Minimum Y	YMIN	COORD
Minimum Z	ZMIN	COORD
Feature type	FTYPE	FTYPE
Feature identifier	FID	FID

A rapid indexing device for retrieving features over any rectangular area, or finding the feature near or nearest to a given co-ordinate.

Header contains units, square size and 2-D/3-D marker.

Items stored in the bounding rectangle/cuboid table

Feature type	FTYPE	FTYPE
Feature identifier	FID	FID
Zone	ZONE	ZONE
Unit	UNIT	UNIT
Minimum X	XMIN	COORD
Minimum Y	YMIN	COORD
Minimum Z	ZMIN	COORD
Maximum X	XMAX	COORD
Maximum Y	YMAX	COORD
Maximum Z	ZMAX	COORD

Contains the bounding rectangles of non-point features. Used to speed up overlay operations.

Contains one logical entry per feature.

Header contains units and 2-D/3-D marker.

Data dictionary

Data type	Type	Range
ALEVEL	integer	0..[illegible]
ATCODE	character	
ATQUAL	character	single character
ATTYPE	character	
ATVAL	character	
CLSIZE	integer	$1..(2^{15}-1)$
CODE	character	any eight characters
COORD	integer	$-(2^{31}-1)...+(2^{31}-1)$
COUNT	integer	$1..(2^{15}-1)$
D2D3	integer	2..3
DIR	character	F,R,B,I,O
DT	integer	date/time code
ENCODE	character	R,C,I
FID	integer	$-(2^{31}-1)..+(2^{31}-1)$
FTYPE	character	any four characters
LDESC	character	
RAW	character	up to $(2^{15}-1)$ characters
RCOORD	integer	$-(2^{15}-1)..+(2^{15}-1)$
RELABS	character	R,A
SDESC	character	
SID	integer	$-(2^{31}-1)..+(2^{31}-1)$
SLEVEL	integer	0..4
SN	integer	$1..(2^{31}-1)4$
STSIZE	integer	$1..(2^{31}-1)4$
TEXT	character	up to $(2^{15}-1)$ characters
TUNIT	character	Y,M,W,D,H,M,S
UNIT	integer	$1..(2^{15}-1)$
ZONE	integer	$-(2^{7}-1)..+(2^{7}-1)$

Implementation

As noted under spatial indices, the network table is in fact an index, aiding access to line data.

Further note that the feature relationships table is used for spatial relationships too. Thus the polygon-line relationship (polygon table) is easily transferred to this table. An area seed (area seed table) is defined as a point feature which owns a number of polygon features (area table) the direction field being used to store the inner/outer flag. Therefore, the only spatial data tables will be the point, line and grid ones.

Appendix F. Task list

TASK 0 Write functional requirements specification (this document)

TASK 1 Write system requirements specification

TASK 2 Write system documentation

TASK 3 Design modules top down
- a) Worker
- b) Supervisor
- c) Manager
- d) Bulk I/O
- e) Interactive editor
- f) Mapping
- g) Manipulation

TASK 4 Code modules bottom up
- a) Worker
- b) Supervisor
- c) Manager
- d) Bulk I/O
- e) Interactive editor
- f) Mapping
- g) Manipulation

TASK 5 Testing of modules
- a) Worker
- b) Supervisor
- c) Manager
- d) Bulk I/O
- e) Interactive editor
- f) Mapping
- g) Manipulation

TASK 6 Selection of microcomputer equipment

TASK 7 Purchase of microcomputer equipment

TASK 8 Write manuals

TASK 9 Implement on mainframe

TASK 10 Implement on microcomputer

Index

The demand for long-term scientific capabilities concerning the resources of the land and its freshwaters is rising sharply as the power of man to change his environment is growing, and with it the scale of his impact. Comprehensive research facilities (laboratories, field studies, computer modelling, instrumentation, remote sensing) are needed to provide solutions to the challenging problems of the modern world in its concern for appropriate and sympathetic management of the fragile systems of the land's surface.

The **Terrestrial and Freshwater Sciences** Directorate of the Natural Environment Research Council brings together an exceptionally wide range of appropriate disciplines (chemistry, biology, engineering, physics, geology, geography, mathematics and computer sciences) comprising one of the world's largest bodies of established environmental expertise. A staff of 550, largely graduate and professional, from four Institutes at eleven laboratories and field stations and two University units provide the specialised knowledge and experience to meet national and international needs in three major areas:

★

Land Use and Natural Resources

★

Environmental Quality and Pollution

★

Ecology and Conservation

● **FRESHWATER BIOLOGICAL ASSOCIATION**
The Ferry House, Far Sawrey
Ambleside, Cumbria LA22 0LP
Tel: 09662 2468 Fax: 6914
Telex: 8950511 ONEONE G
REF 16173001

○ **The River Laboratory**
East Stoke, Wareham
Dorset BH20 6BB
Tel: 0929 462314 Fax: 462180
Telex: 8950511 ONEONE G
REF 16174001

■ **INSTITUTE OF HYDROLOGY**
Wallingford, Oxon OX10 8BB
Tel: 0491 38800 Fax: 32256 Telex: 849365

□ **Plynlimon Office**
Staylittle, Llanbrynmair
Powys SY19 7DB
Tel: 05516 652

INSTITUTE OF TERRESTRIAL ECOLOGY

▲ **Edinburgh Research Station**
Bush Estate, Penicuik, Midlothian EH26 0QB
Tel: 031-445 4343 Fax: 3943 Telex: 72579

△ **Banchory Research Station**
Hill of Brathens, Glassel
Banchory, Kincardineshire AB3 4BY
Tel: 03302 3434 Fax: 3303 Telex: 739396

△ **Merlewood Research Station**
Grange-over-Sands, Cumbria LA11 6JU
Tel: 04484 2264 Fax: 4705 Telex: 65102

▲ **Monks Wood Experimental Station**
Abbots Ripton, Huntingdon, Cambs PE17 2LS
Tel: 04873 381 Fax: 467 Telex: 32416

△ **Bangor Research Station**
Penrhos Road, Bangor, Gwynedd LL57 2LQ
Tel: 0248 364001 Fax: 355365 Telex: 61224

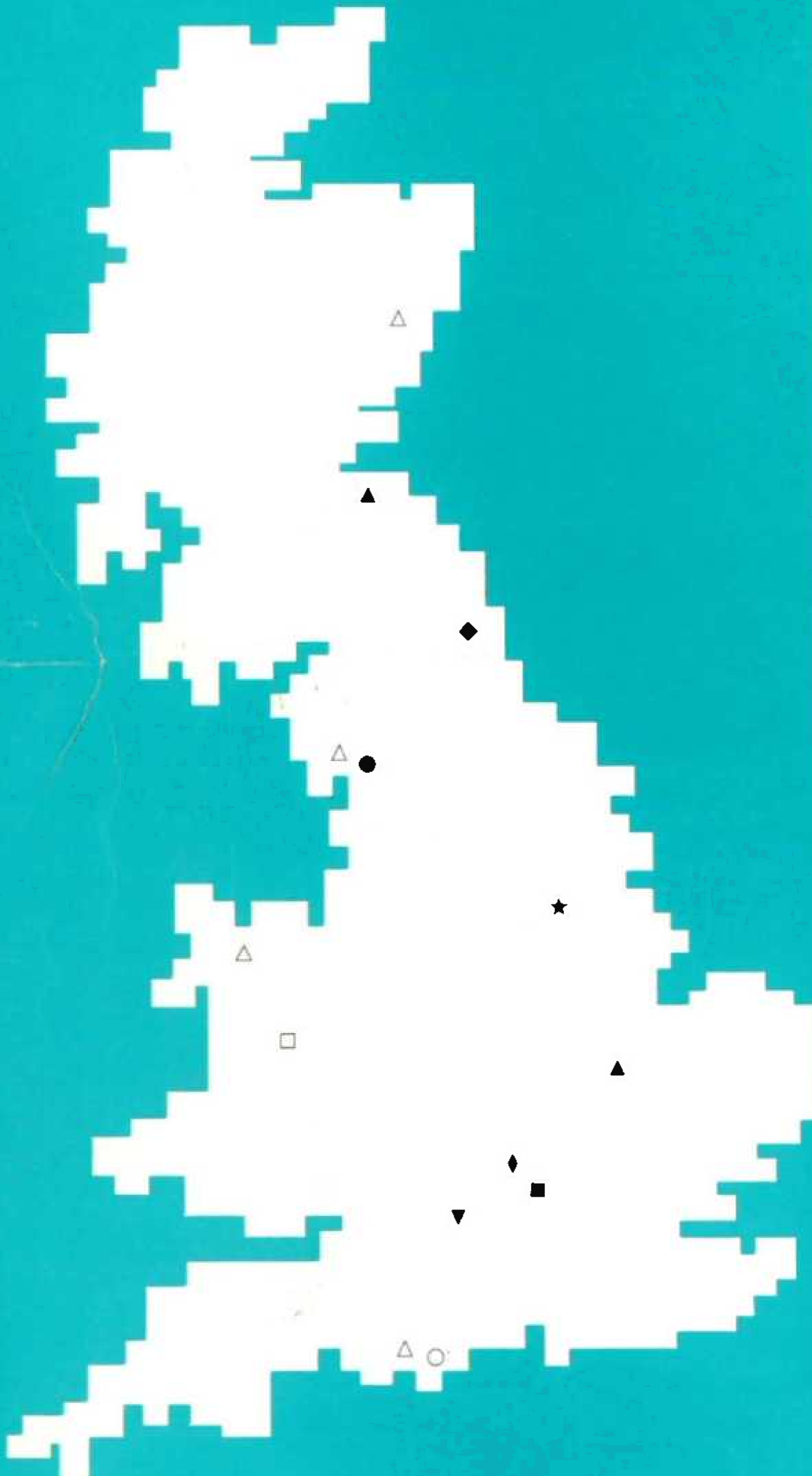
△ **Furzebrook Research Station**
Wareham, Dorset BH20 5AS
Tel: 0929 51518 Fax: 51087

♦ **INSTITUTE OF VIROLOGY**
Mansfield Road, Oxford OX1 3SR
Tel: 0865 512361 Fax: 59962 Telex: 83147

★ **UNIT OF COMPARATIVE PLANT ECOLOGY**
Dept of Plant Sciences, Sheffield University, Sheffield S10 2TN
Tel: 0742 768555 Fax: 760159 Telex: 547216

◆ **UNIT OF WATER RESOURCES SYSTEMS RESEARCH**
Dept of Civil Engineering
Newcastle University
Newcastle upon Tyne NE1 7RU
Tel: 091-232 8511 Fax: 261 0191 Telex: 53654

▼ **DIRECTORATE OF TERRESTRIAL & FRESHWATER SCIENCES**
Natural Environment Research Council
Polaris House, North Star Avenue
Swindon SN2 1EU
Tel: 0793 40101 Fax: 511117 Telex: 444293



DIRECTORATE OF TERRESTRIAL & FRESHWATER SCIENCES

Natural Environment Research Council