Proposal for a Great Ouse database

Prepared by:

THE INSTITUTE OF HYDROLOGY

Crowmarsh Gifford
Wallingford
Oxon OX10 8BB

(0491-38800)

May 1988

Project No. T12/4c

Contents

Appendices

Summary

This report is the first part of a two-stage project to create a Great Ouse Database.

The objective of the system is to record, analyse and display data about catchments necessary for the study and management of water quality matters.

The report proposes a system design and identify the components required to meet AW's need.

The feasibility study concludes that such a system is a practical proposition, but that its development cost will exceed the available resources.

However, IH and others have a similar requirement. If the costs and conditions outlined in the report are acceptable to AW, then IH would be prepared to approach its management to fund the balance.

An initial development plan, to be completed over 18 months, is outlined.

1. Introduction

1.1 Background

At a meeting held in Cambridge on 26 November 1987 the Institute of Hydrology (IH) was asked to design, write and implement a database for the Cambridge Division of Anglian Water (AW). It was suggested that the work be carried out in two stages. First, there should be a brief assessment of the requirement and proposals put forward for the facilities that the system should provide. Then, if these were acceptable, there would be a second stage to create the system.

In an exchange of letters dated December 1987, the Institute agreed to undertake the first stage.

1.2 Terms of reference

The terms of reference for the first stage were that the Institute should prepare a short report identifying the requirement and proposing a solution together with costs and a development plant; the major headings of the report to be:

Introduction
Existing systems
Proposed system
Development plan
Costs

1.3 Objective

Briefly, the objective is to design, write and implement a database to be called the Great Ouse Database. Its primary function will be to hold data related to water quality. These will include the descriptions of places of interest such as sampling points, sewage works, trade effluent discharges, the storage locations of dangerous substances, details of consents and licences, and water quality time series data. Additionally, it should be able to store spatial data, in particular the rivers. In essence, the requirement is to be able to record a description of a catchment in time and space with respect to water quality. Access to the data is to be via specific application packages each addressing a particular problem, such as the creation of river quality maps or the provision of statistical analyses of sample data. Most data will be abstracted from existing archives, though there will be a need to be able to add and update data within the system. Ideally the system should be able to fit on a personal computer or small mini-computer. The resources available are in the region of £40,000.

2. Existing systems

At the time of writing, descriptions of the existing systems are unavailable. They will be added when the information is available.

3. Proposed system

3.1 Introduction

In this section a proposal is put forward that goes beyond the stated requirement. This has been done because, from experience on the Water Archive System and with geographic information systems, it is possible to foresee how the requirement is likely to evolve in the future. To design the system purely on the basis of the present requirement would undoubtedly impose severe restrictions on future development. In this proposal, therefore, every endeavour has been made to look as far into the future as possible. In Section 5, suggestions are put forward as to the parts of the proposal that should be developed now.

3.2 System objectives

The main design objectives are that the system should:

(a) provide for the input, update, storage and analysis of the location and extent of physical features together with records over time describing their inter-relationships, their attributes and other variables observed at them. Examples of features are sewage treatment works, discharges, sampling points, river flow measurement stations, factories, rivers, soil polygons and administrative boundaries. Examples of attributes and observations are river flows, concentrations of chemicals, lists of hazardous substances, names and dimensions;

(b) be simple in concept;

(c) be general so that new types of feature or attribute can be added without major system changes;

(d) be capable of development in stages;

(e) be suitable for use on a large personal computer (e.g. IBM PS/2 model 80) or mainframe (e.g. the A.W. Divisional VAX's).

(f) be suitable for interactive use;

(g) be geographically based (i) because most of the data are geographical in nature and (ii) to minimise the effect on the database of administrative change.

3.3 System outline

3.3.1 Data

The system will provide for two types of data:
features
feature attributes (spatial, relational, time-series)

A feature is any object of interest to the user. Normally, it will be a physical object such as a sewage treatment works, dam, river, sampling point, trade effluent discharge or bridge. It may also be a location, such as the site of a spillage, an area such as a site of scientific interest, or a limit of extent, such as the downstream end of a section of river in a particular pollution class. More abstract concepts such as a licence or a constant can also be classed as features. The system will be entirely general, and the types of feature to be stored will be defined by the user in a dictionary.

For the purpose of recording the position and extent of features, features are classed as being:

abstract, point, line, network, polygon, area or raster.

An important element in the description of a feature is its relationship with other features. While many of these relationships are spatial in nature and are therefore recorded by their spatial data, many are not, and a special facility will be provided to record relationships, such as the sampling points relating to a particular sewage treatment works.

The non-spatial aspects of features will be recorded at two levels. At the simple level. where either no, or only minimal information is required (beyond a feature's existence, type and possibly location), a name and a simple 8-character user defined code will be able to be associated with each feature. For some features, however, a more comprehensive description is required. This will be provided for by a generalised system for recording attributes (or determinands). The attributes to be recorded are decided by the user and defined in a dictionary, each attribute being assigned an identifying code. Attributes may be of different types, for example, feature type codes, chemical determinands, microbiological determinands and plants. Pre-existing coding systems may be used in the system.

The storage of attributable data may be visualised as a table of feature, date, time and attribute value. Values may be integer, real or character. Other value types such as date or coordinate pairs may be possible later.

3.3.2 System structure

To allow for staged development, the system will be designed as a series of separate components as shown in Figure 3.1..

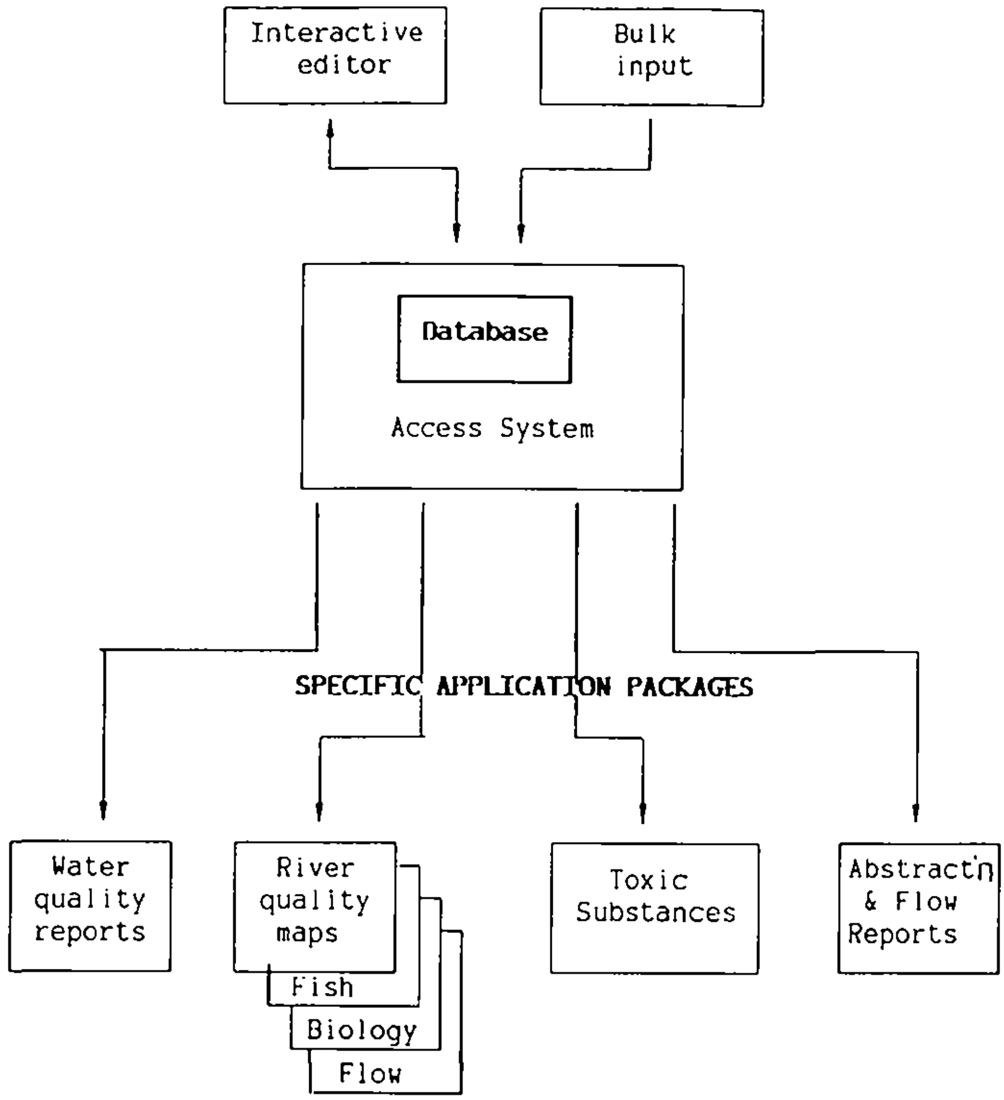


Fig. 3.1 The main components of Great Ouse Database

At the centre will be a database where all data and supporting dictionaries will be held. Data will reach the database either via the bulk data input system or via an interactive editor. The bulk input is conceived as being for data received from outside systems. The editor is for capturing digital map data, adding small amounts of attribute data and for correcting information in the database. It will include display facilities.

The access system will provide a route into the database via a subroutine library. However, it is envisaged that probably only users with a fairly sophisticated understanding of the system would use this facility. Normal access will be via a number of specific application packages, for example, water quality reports, river quality maps, effluent discharge reports, abstraction and flow reports and hazardous materials inventories. Other examples being considered are flood estimation, low flow estimation and the vulnerability of groundwater to pollution.

In addition to these facilities, there will be facilities for maintaining the database, for example, file reorganisation, integrity checks, archiving and recovery procedures.

3.3.3 Hardware

The system has been designed to take advantage of the new range of IBM micro computers. Figure 3.2 shows the configuration envisaged.

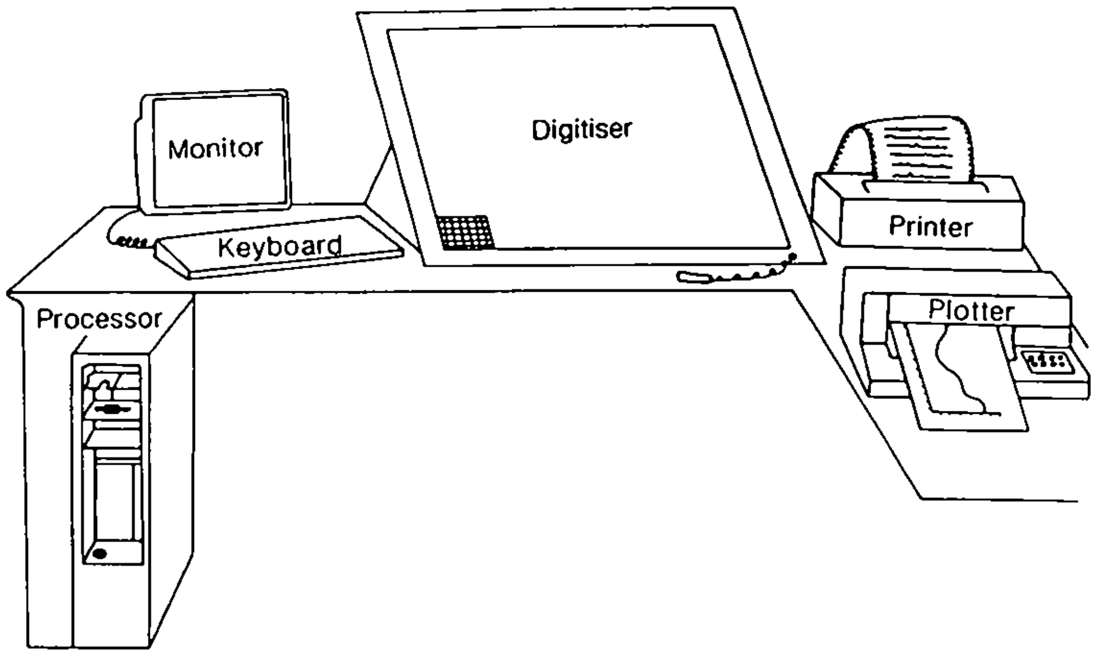


Figure 3.2 *Outline of the system hardware*

Essential items would be the computer, keyboard, high resolution screen, printer and digitiser. The quality of the printer and size of digitiser would be a matter of choice for the user. Desirable, but not essential at the outset, would be additional mass storage and a colour plotter.

It is planned that the system could also be used on existing IBM/PC's such as the AT. However, it is expected that users would probably require more space and higher quality graphics within a fairly short period.

3.4 Proposed system description

3.4.1 Introduction

In this section the outline given in Section 3.3 is expanded. The system is entirely general, and is described in general terms. Its application to specific problems is illustrated by example.

The description of the data and how it will be held is confined to its logical structure, i.e. the picture of the system as it will be seen by the user. Although a well developed plan exists of how the data will actually be held, the formalisation of that plan is seen as part of the next stage.

3.4.2 Data structure

3.4.2.1 Overall structure

A feature is any object for which the user wishes to record information about its location and extent, its relationship with other features and time-series of observations made at or about the feature. The user will decide the types of features to be recorded and will inform the system by recording them in a Feature Dictionary. Appendix 1 gives examples.

The information about a feature will be held in one or more of three linked logical tables which may be visualised as shown in Figure 3.3.

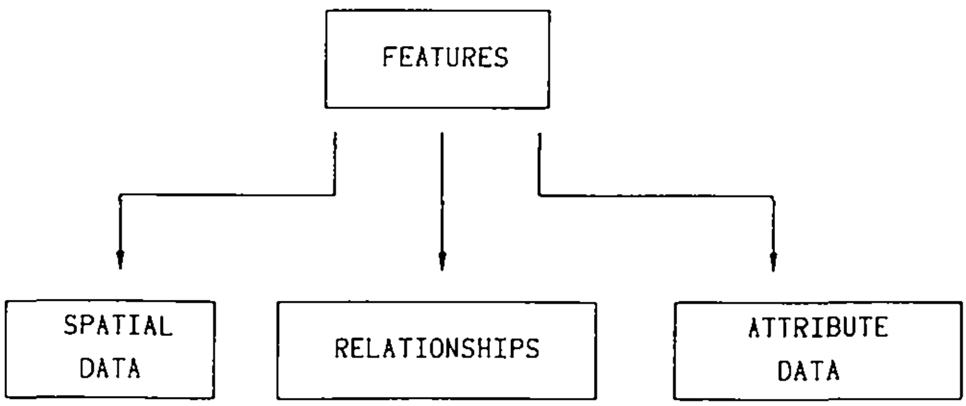


Figure 3.3 Overall data structure

At the centre of the data structure will be the feature table. This is simply a list of all features known to the system. It will contain two items of data, the feature's type code and its reference number. The entry may be regarded as a hook upon which all other data hang.

3.4.2.2. Spatial data. The position and extent of a feature will be recorded in the spatial data tables. Spatially, features may be classed as abstract, point, line, network, polygon, area, or raster. The spatial data structure may be visualised from Figure 3.4.

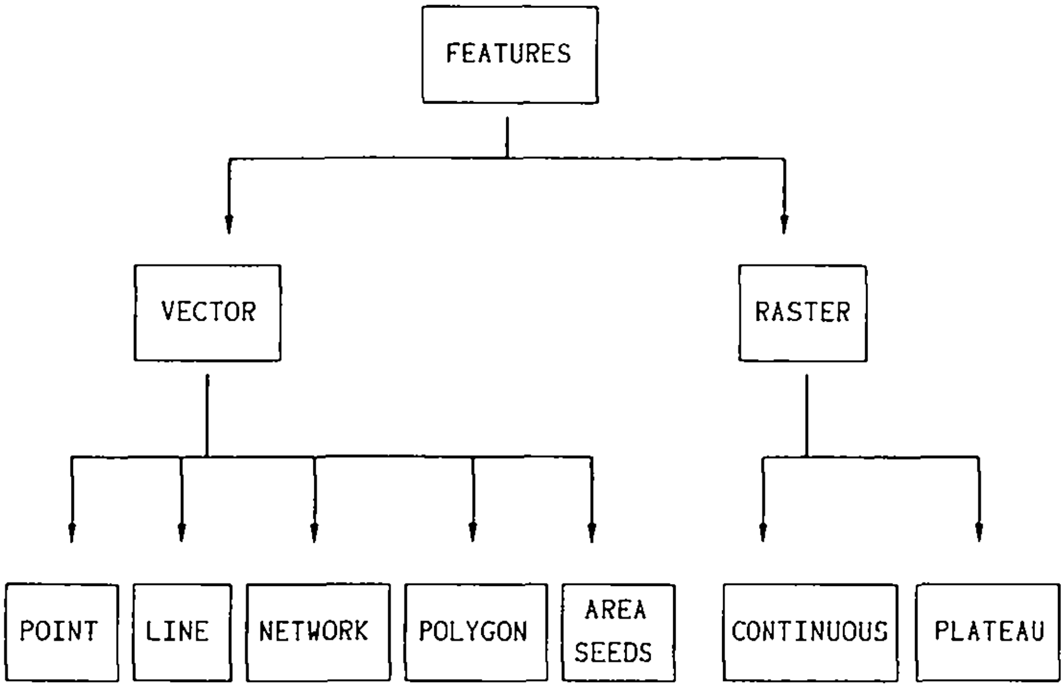


Figure 3.4 The spatial data structure

Examples of the different vector spatial types are (see Figure 3.5):

Point	raingauges, gauging stations and treatment works
Line	contours, isohytes and geological fault lines
Network	rivers, sewers
Polygon	national, water authority, divisional (SSI), areas of outstanding natural beauty (AONB) and catchment boundaries
Area	the surface within a polygon

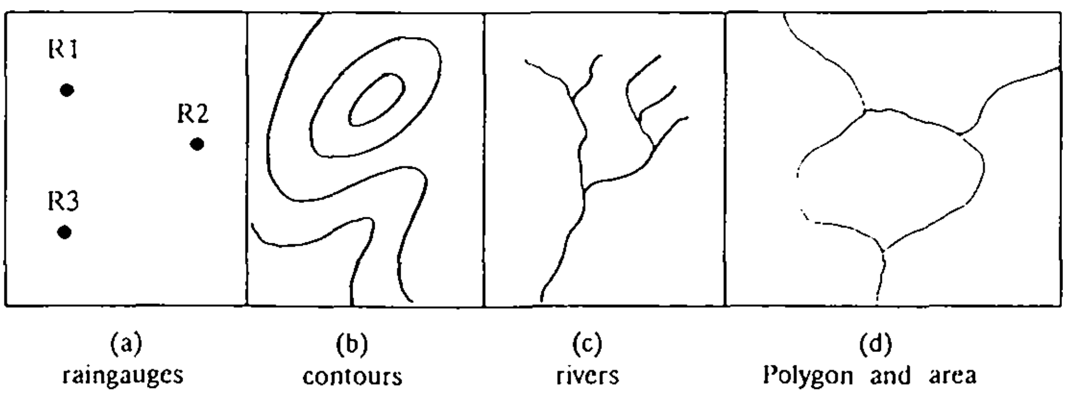


Figure 3.5 Examples of the different spatial classes of feature

Point features are any whose position and extent are adequately described by a single pair of coordinates. Line features are those where the position of the line and its attributes (e.g. a contour with its height) are a sufficient description. These may be contrasted with networks such as rivers, where direction and connectivity are vital. The distinction between polygon and area features is that a polygon is the line around the edge of an area. An area is bounded by an outer polygon and may have inner boundaries as well, for example, Snowdonia National Park surrounds Blaenau Ffestiniog. The exploitation of the topology of the spatial data is an important objective of the system.

Although it is frequently easier to capture contour, polygon and area data in vector form, it is generally very much easier to manipulate them in raster (or gridded) form. Two storage techniques are required to hold such data in gridded form; one for continuous data such as rainfall or elevation and one for plateau type data such as soils, areas of afforestation, administrative areas or catchment areas.

The vector spatial data will be held in the form of x, y or x, y, z coordinates, where the coordinate system and unit of measurement will be decided by the user. National grid coordinates would be a common choice for work in the UK. Gridded data will be held either as values at the grid intersections for continuous data, or as values for the grid squares for plateau type data. The grid size will be determined by the user.

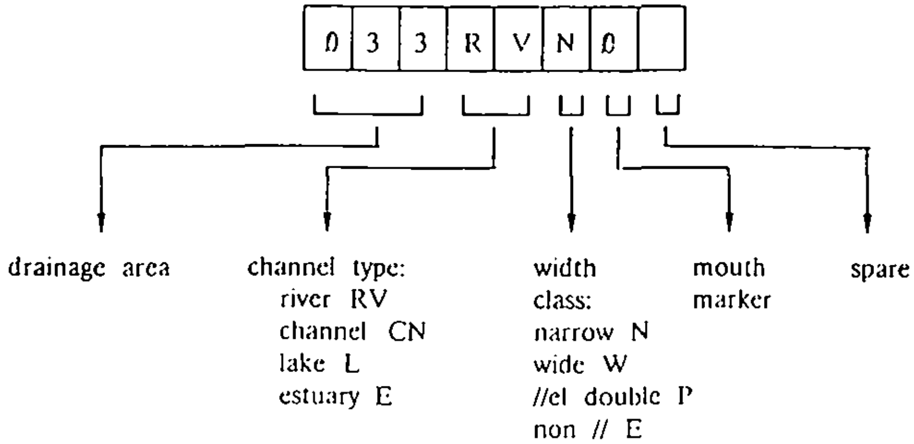
Features need not be spatial in nature, for example a licence or a consent, in which case they will be classed as abstract.

3.4.2.3 Feature relationships. Many features will be large complex places, where the same variables may be observed at different locations within the feature, for example, sampling the different processes within a treatment works. In such cases, it may be desirable to treat each sampling point as a separate feature, and the works as a whole as another feature. In such a case, it will be highly valuable to be able to cross reference the works to its sampling points. To achieve this a relationship file will be provided. It will simply provide a two-way cross-reference between features. Where relevant, it will also be able to record topological relationships where sequence and direction are important.

3.4.2.4. Feature attributes. Attributes record the observations made at or about features: chemical concentrations, rates of flow, power consumed, temperatures, dimensions, etc.. The attributes to be recorded will be decided and defined by the user. Frequently, only the simplest information will need to be associated with a feature. To provide for this every feature may have associated with it:

a user code
a name

Both are optional. The user code is a string of upto eight characters which the user may use in any way required. For example, for rivers it might be structured as follows:



The feature's name is self-explanatory, but an example for a gauging station might be 'The Cam at Bottisham'.

The facilities for attributes described so far, are only for recording the simpler time invariant items. For all other data a general system for recording attributes will be provided. Attributes may be associated with a feature at number of levels as shown in figure 3.6.

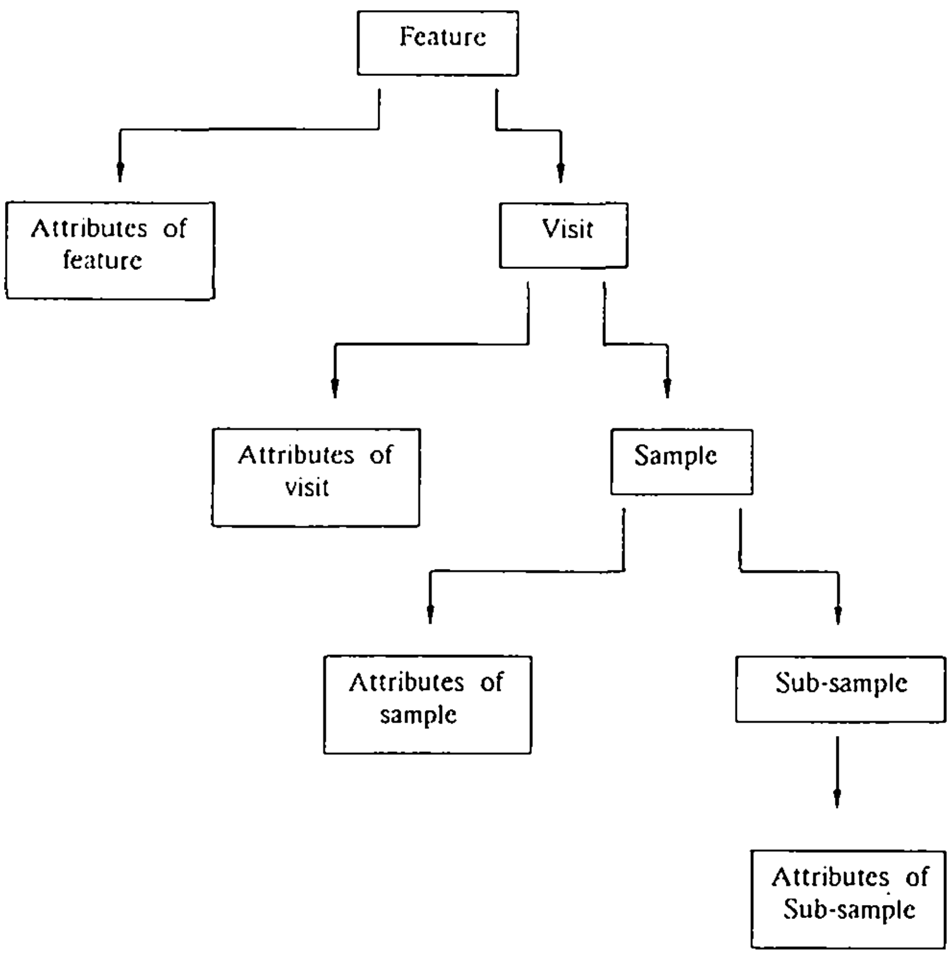


Figure 3.6 The outline data structure for attribute data.

Each feature may have associated with it any number of attributes whose values, for all practical purposes, do not change with time, e.g. method of construction for a dam. It is worth noting here that the number of things about a feature that are immune to change are surprisingly few*. To deal with things that change more rapidly, the system design envisages that features are 'visited' from time to time. During the visit, things will be noted (attributes) that relate to the visit as a whole, e.g. at a biological sampling point, the degree of shading, state of weed growth in the river, the water level and rate of flow. Then, one or more 'samples' may be taken. Were one of these to be a water quality sample, then its attributes would be the determinand results for the sample together with any administrative data, analyst's comments, purpose, etc. etc. Samples may be sub-sampled as is the practice in biological sampling, and the attributes recorded in a similar way.

Before data for a particular attribute can be entered into the system it must be defined to the system in the attribute dictionary. Pre-existing coding systems, such as the chemical, microbiological, sewage, hydrological and macrophyte determinand dictionaries can be used as they stand. To allow for this the dictionary is divided into attribute types, each identified by a 2-character attribute type code. To define a new attribute type to the system the following information will be required:

attribute type code, e.g. CD
name of type, e.g. chemical determinands.
maximum length of the codes of that type in characters, e.g. 4

To enter a particular attribute code within a type, the following will be required:

attribute type code
attribute code
attribute value type and storage format:
- integer, e.g. I*4
- real
- character, e.g. A20
- date
- etc.

unit of measurement for storage (if not stored explicitly)
short name
full name)
definition)
references) all optional
user data)

The reason for holding definition and references is that poor definition has caused significant problems in the past, for example, in water chemistry, confusion between common names, trade names and systematic names. As these problems were resolved, the dictionary became an important source of reference. The user data areas allows for recording conversion factors and other constants or notes.

---

* These data will actually be treated as time-series data, albeit very slow moving. Here, it will only be necessary to make an entry when something changes, the other attributes being still valid. At all other levels, the attribute values will be taken to be true only for the time they are entered against.

Where attribute values are themselves entered as codes, it is often desirable that these should be converted back to plain text on output. The allowed values of these codes may also be stored in the dictionary together with their interpretation.

The storage of attribute data may be visualised as the table in Figure ---

Feature type	Feature i.d.	Date	Time	Visit i.d.	Sample no.	Sub-sample no.	Attribute type	Attribute code	Value	Qualifier

Figure 3.7. *The logical data structure for storing attribute data*

[N.B. It is important to note that Figure 3.7 depicts the user's view of how data are stored, and NOT how they will actually be stored, which will be in a much more compact form.]

To what the attribute value refers depends upon how many of date, time, visit, sample number and sub-sample number have been completed. If date and time only are completed, the attribute describes the feature directly. If date, time, visit and sample number are entered, the attribute describes the sample, e.g. concentration of dissolved oxygen. If all are present, then the attribute describes a sub-sample, for example, a count of stone flies. The particular feature at which it was observed will be identified by the type of feature and its identifier. These two items will connect all the feature's data together. The attribute to which the value relates is implied by its type and code, e.g. CD180 would be orthophosphate in mg/l as P. Some users consider that a value alone is sometimes insufficient and wish to qualify it with 'less than' or 'greater than' or record the method of derivation. The qualifier meets this need.

4. Development and implementation plan

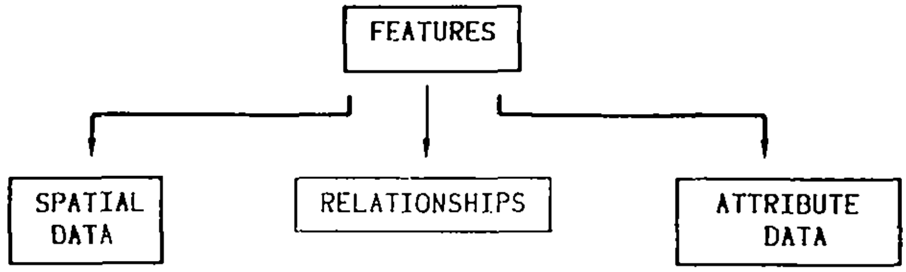
4.1 Introduction

Section 4.0 outlined a system design that looks as far into the future as possible. Plainly, the immediate development of all the ideas is beyond the available resources. This section identifies those aspects that will meet most of the immediate need and which can be developed within the resources.

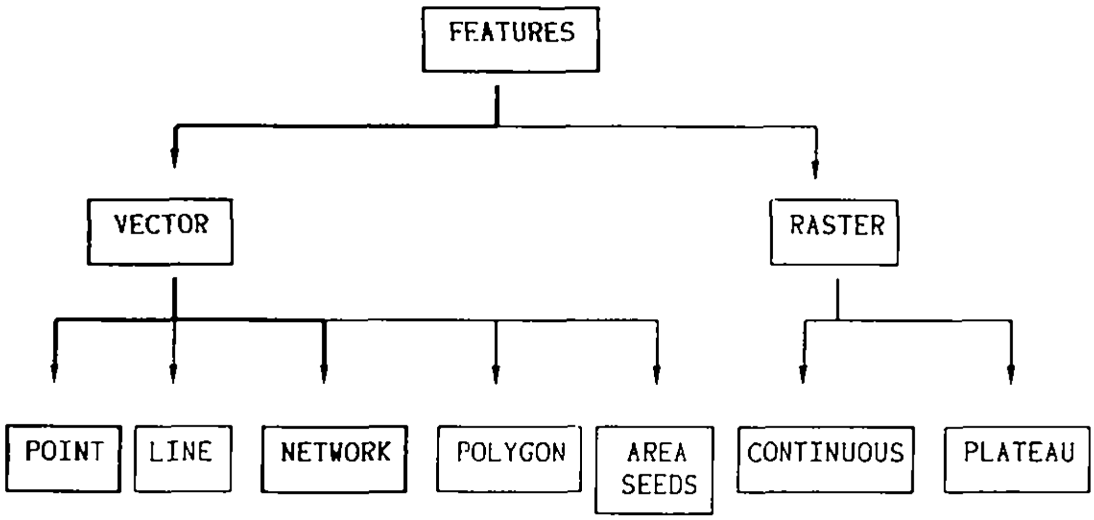
4.2 Elements of the proposal to be developed or purchased

4.2.1 The database

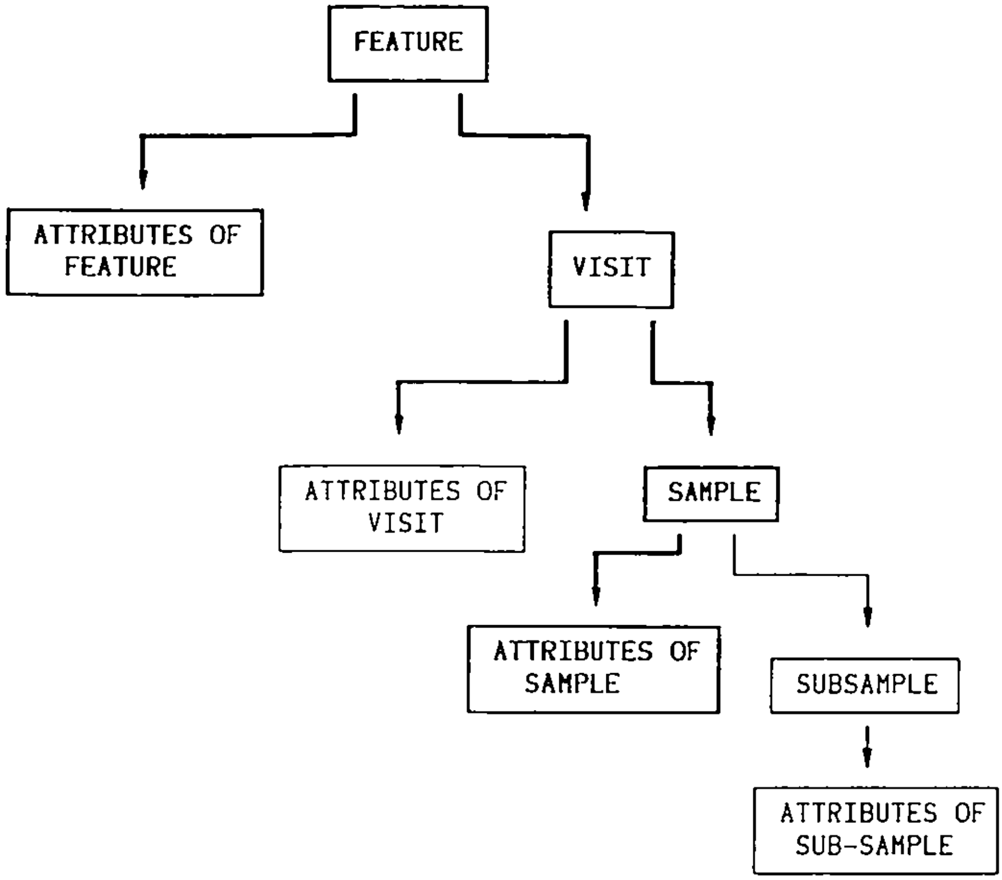
Figures 4.1a, 4.1b, and 4.1c show in bold lines those elements of the database that would be developed.


FEATURES
SPATIAL DATA
RELATIONSHIPS
ATTRIBUTE DATA

(a)


FEATURES
VECTOR
RASTER
POINT
LINE
NETWORK
POLYGON
AREA SEEDS
CONTINUOUS
PLATEAU

(b)



(c)

Figure 4.1 Elements of the data structure to be developed

This structure will meet the requirement to store point features such as treatment works, sampling points and the storage locations of toxic substances. It will also enable the features to be described and for sample data to be stored against any feature. In addition, the system will be able to store the digitised river network in a way that will allow the structure of the data to be exploited.

4.2.2 Data input and update

4.2.2.1 Bulk input. Initially, the requirement for data input, is that the system should be able to accept data from existing systems. Assuming these systems can output data to computer files that can be transferred between machines and read by the user's own programs, then it is proposed to develop three or four reformatting programs for reading in the data. The exact nature of these programs will be specified in the next stage of work, when the output formats of the existing systems are known.

4.2.2.2 Interactive input. Although, the data will be transferred from existing systems and will be be assumed to be correct, it is almost inconceivable that there will be no requirement to alter data. It is therefore proposed to develop the first stage of the editor, which will have the following facilities:

Insert
Amend
Delete
Select
Display
List
Plot
Print

INSERT will allow for the insertion of new features and will contain provision for digitising their position, adding feature attributes and new samples.

AMEND will allow for the alteration of a feature's position and the alteration of its attributes or sample data including the insertion, amendment and deletion of individual attributes (determinands).

DELETE will allow for the removal of positional data, whole 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 the same as 'display' but with output to a plotter.

PRINT the same as 'list' but with output to a printer.

These facilities of SELECT, DISPLAY, PLOT, LIST, PRINT should meet most of the initial requirements for retrieving data from the system.

4.2.3 Query language

It is not proposed to provide an 'ad hoc' query language at this stage. Each application will have purpose written routines to select the required data.

4.2.4 Applications

4.2.4.1 River quality mapping. The user will set up three types of feature: one to define the upstream limit of the survey on each river; one defining the flow at points throughout the river system; and one to denote the downstream limits of quality classifications. A program will then use these together with the digital river data to prepare a river quality map, coloured according to quality and with line thickness proportional to flow.

The user will be able to determine the area of the map by specifying the SW and NE corners of the area, or by specifying one or more points upstream of which a map is required.

4.2.4.2 Water quality report. This application will enable a user to prepare a water quality report for any sampling point for a user defined set of determinands and date/time range. The report will list the site details, name, reference number and location and tabulate the results for each sample, each column being a determinand and each row a sample. Optionally, summary statistics may be requested. These will include mean, minimum, maximum, standard deviation and any specific percentile.

The application will also allow the preparation of a summary report covering several stations. The user will identify the determinands and the required statistic for each determinand, the sampling points and a date range.

4.2.4.3. Effluent quality and compliance reports. This report is not yet defined in detail and is not included in the plan of work. It will be similar to the Water Quality Report, but specifically aimed at effluent data and the calculation of statistics required to determine compliance with consent conditions.

4.2.4.4. Abstraction and river flow reporting. Again this requirement is not yet defined in detail. However the objective will be to list details of abstractions and their related licence data together with amounts abstracted over time. The same facility should be able to handle daily river flow data.

4.2.5 Hardware

The system will be designed to run on an IBM PC/AT. Appendix 4 lists the necessary equipment. This, however, is seen very much as a starting position and it is expected that as the system becomes used more intensively, there will be a demand for better quality graphics, more storage space for data and increased speed. For these reasons, the system will be aimed primarily at the equipment in Appendix 3. It is strongly recommended that resources be allocated for its purchase within 12-24 months from beginning to use the system.

If the project goes ahead, immediate provision will need to be made for the purchase of any items in Appendix 4 not already held by Anglian Water, as they will be required during the development phase. Equipment in Appendix 3 will not need to be purchased until required by Anglian Water as the Institute will be purchasing such equipment for its own research work.

4.3 Implementation

The main problem in meeting the requirement is cost. From our experience with other systems and discussions with water authorities in the UK and abroad, there appears to be a growing demand for such a system. There is also an internal requirement within IH and NERC in general for such a system. We therefore consider that we can develop the system proposed in this section for the costs outlined in Section 6, if it is acceptable to AW that IH retains the rights to the

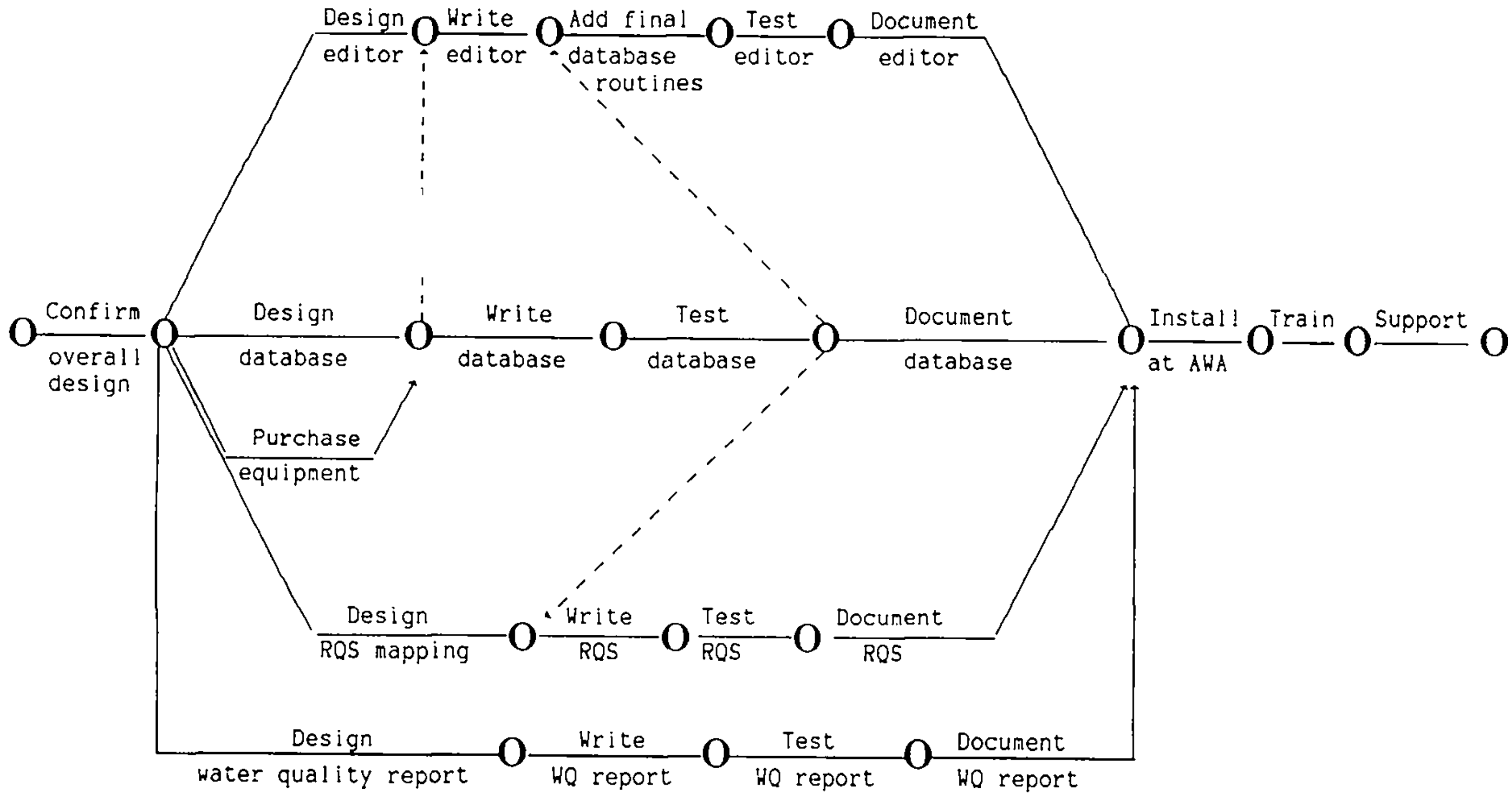


Figure 4.2 Outline plan of work

software.

The plan for implementing the system is outlined in Figure 4.2 and amplified in Appendix 5. At the outset, there would need to be a period of collaborative work with AW to clarify points of detail, e.g. calculations, data exchange formats, etc. This would probably amount to 1-man week's worth of AW effort. There would be a design phase at the end of which final details would be agreed. When the system was written and tested, it would be installed at AW. Training would be given to involved staff and practical help and advice given during the initial phase of use.

It is possible, though not certain, that an editor might be bought in. The bulk of the work, however, lies in the database design and implementation and this would definitely be carried out by IH as would the RQS mapping system and statistical analysis package.

It is considered that the database could be completed and installed over a period of 12-18 months from the start of work. Subject to detailed design work, it is considered that a preliminary version of the editor and database could be installed in about 12 months and that tidying up, completion of the River Quality Mapping system, statistical analysis system and documentation would take a further 6 months.

It is important to stress here that it will be a new system and that it is probable that there will be 'bugs' in it at the outset, though we will, of course, do our best to minimise these. Forbearance will be required while these are identified and removed.

5. Costs

In arriving at the costs below, we have separated the task list into items specific to AW, and items which either we have an interest in ourselves or whose development cost we may be able to recover in part from other sources. Examples of the former are the bulk data entry programs, the water quality report, travel and subsistence, training, installation and support. The latter are the database and access systems. The hardware costs assume that initial installation will be on AW's IBM/PC.

We estimate the cost of developing all of the central data base at about £250,000. To fulfil the proposal in Section 4 it will be necessary to complete about half of this work If the costs below are acceptable to AW then IH will seek permission within NERC to fund the balance from within its own resources over two financial years.

The costs to AW may be summarised as follows (see also Appendix 6):-

	£
Database design and production	25,000
Editor	3,000
ROS mapping system	5,675
WQ reporting system	5,675
Installation at AW	725
Training	1,000
Support	1,800
Hardware (to be arranged by AW)	(4,100)
TOTAL over 2 financial years	£42,875

Appendix 1

Examples of features and feature codes.

Point features

RNG	Raingauge
RFM	River flow measurement station
DAM	Dam
STW	Sewage treatment work
DIS	Discharge
ABS	Abstraction
SPT	Sampling point

Line features

FLT	Fault lines
CONT	Contours

Network features

WB	Water body, eg. rivers, lakes, etc.

Polygons

CAT	Catchment boundary

Area

SSI	Site of scientific interest
AONB	Area of outstanding natural beauty

Appendix 2

Logical data structure

Items stored in the feature table

Feature type	FTYPE	A4
Feature identifier	FID	I*4
Feature class	FC	A1
User code (optional)	FU	A8
Feature name (optional)	FB	A

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

Items stored in the point table

Point type	XTYPE	A4
Point identifier	XID	I*4
Zone	ZONE	I*1
Unit	UNIT	I*2
X-coordinate	XCOORD	I*4
Y-coordindate	YCOORD	I*4
Z-coordinate	ZCOORD	I*4

Records the position of point features including the nodes of networks

Items stored in the area seed table

Area type	ATYPE	A4
Area identifier	AID	I*4
Zone	ZONE	I*1
Unit	UNIT	I*2
X-coordinate	XCOORD	I*4
Y-coordinate	YCOORD	I*4

Allows attributes to be attached to an area.

Items stored in the network table

Line type	LTYPE	A4
Line identifier	LID	I*4
Zone	ZONE	I*1
Unit	UNIT	I*2
2-D/3-D marker	D2D3	I*1
End X-coordinate	XF	I*4
End Y-coordinate	YF	I *4
End Z-coordinate	ZF	I**
End but one X-coord	XFB1	1*4
End but one Y-coord	YFB1	1*4
End but one Y-coord	ZFB1	I**
Other end but one X-coord	XLB1	1*4
Other end but one Y-coord	YLB1	1*4
Other end but one Z-coord	ZLB1	I*4
Other end X-coord	XLB1	1*4
Other end Y-coord	YLB1	1*4
Other end Z-coord	ZL	I*4
First last marker	FL	A1

Records the connectivity of a network.

N.B. Every line in the network has two entries in the table. One with the coords in order with FL='F' and one with the coords reverse and FL='l'.
The 2 coordinate is not stored for 2-D data.

Items stored in the line table

Line type	LTYPE	A4
Line identifier	LID	I*4
Zone	ZONE	I*1
Unit	UNIT	I*2
2-D/3-D marker	D2D3	I*1
Relative/absolute marker	RELABS	A1
No. of points	NOPTS	I*2
absolute X-coord	XCOORD)	I*4
absolute Y-coord	YCOORD)	I*4
absolute Z-coord	ZCOORD)	I*4 repeated
OR		
relative X-coord	XCOORD)	I*2
relative Y-coord	YCOORD)	I*2
relative Z-coord	ZCOORD)	I*2 repeated

Records the position of lines

N.B. When the relative coords are used, the first coord of a line is absolute and held as I*4.

The Z coordinate is not stored for 2-D data.

Items stored in the area polygon table

Area type	ATYPE	A4
Area identifier	AID	I*4
Polygon type	P YPE	A4
Polygon identifier	PID	I*4
Inner/outer marker	IOM	A1

Relates an area to the inner and outer poylgons 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 polygon table

Polygon type	PTYPE	A4
Polygon identifier	PID	I*4
Sequence number	SEQ	I*2
Line type	LTYPE	A4
Line identifier	LID	I*4
Direction	DIR	A1

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

Items stored in the squares table

Zone		ZONE
Minimum X	of square	MINX
Minimum Y	of square	MINY
Feature class		FC
Feature type		FTYPE
Feature identifier		FID

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

Contains one logical entry per feature.

Items stored in the bounding rectangle table

Feature type	FTYPE	A4
Feature identifier	FID	I*4
Feature class	TC	A1
Zone	ZONE	I*1
Unit	UNIT	I*2
Minimum X	MINX	I*4
Minimum Y	MINY	I*4
Maximum X	MAXX	I*4
Maximum Y	MAXY	I*4

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

Items stored in the attribute values table

Feature type	FTYPE	A4
Feature identifier	FID	I*4
Date	FDATE	I*4
Time	FTIME	I*4
Visit identifier	VISID	I*4
Sample number	SNUM	I*2
Sub-sample number	SSNUM	I*1
Attribute type	AT	A2
Attribute code	AC	A
Attribute value	AVAL	A
Attribute qualifier	AQUAL	A

Appendix 3

Recommended hardware

Listed below are the items of hardware required to run the proposed system, together with a number of items which are optional (*), but which would enhance the performance of the system. The costs are at current list prices (January 1988), and it would be reasonable to expect to be able to negotiate a discount of at least 20%.

ITEM		List Price (ex VAT) £
IBM PS/2 Model 80, 2Mb memory, 115 Mb disc		7056
Enhanced keyboard UK (PS/2)		189
System board memory expansion kit 2 Mb		613
80386 2/6 Mb memory expansion		823
Second 115 Mb fixed disc drive		1991
80387 20 MHz maths co-processor		943
IBM PS/2 colour display 8513, 12"		583
OR		
PS/2 colour display 8524, 16"		1204
* PS/2 display adaptor 8514A		835
DOS 3.3		70
PS/2 model 80 technical reference		98
IBM Proprinter X24, 8" width		601
OR		
IBM Proprinter XL24, 13½" width		789
*IBM colour Jetprinter		754
* pen plotter		
IBM mouse option		60
Digitiser, AO	from approx.	4000
*Maintenance agreement	price not available but of the order 15%	
TOTAL PRICE	from	£15000

* Optional enhancements

Appendix 4

Minimum hardware

Listed below is the minimum set of hardware items to operate the system. The costs are current list prices (January 1988), it would be reasonable to expect to be able to negotiate a discount of at least 20%.

ITEM	LIST PRICE (ex VAT) £
IBM PC/AT	Already available
*IBM enhanced colour graphics adaptor and monitor	1000
DOS 3	Already available
Printer	500
Digitiser A3	1000
IBM plotter A3, 6 pen	1600
Maintenance agreement	Price not available but of the order of 15%
TOTAL PRICE	£4100

* Non-IBM equipment may cost less.

Appendix 5

Tasks to be completed for the intitial development of GOD

TASK 1 Confirm overall design

Description

Define scope of system:

a) data
b) functionality
c) hardware

Identify the major system components, e.g.:

editor
database
access system
applications

Outline the major design principles upon which each component will be based, e.g.:

data structures
menu/command driven
languages
interfaces
etc.

TASK 2 **Design the database**

Description

1) Design logical data structure

2) Design physical data structure

3) Database initialisation system design

4) Database initialisation program design

5) Database update system design

6) Database update program design

7) Database access system design

8) Database access program design

9) Database maintenance system design

10) Database maintenance program design

It is probable that there will be two levels of routines for accessing and maintaining the database, high and low. The low level routines will relate to the physical data structure and the high level to the logical data structure which is the view of the system that the user will see.

TASK 3 Write database

	Code	
	Compile	Database low level initialisation programs
	Test	
	Code	
	Compile	Database high level initialisation program
	Test	
	Code	
	Compile	Database low level update programs
9.	Test	
10.	Code	
11.	Compile	Database high level update programs
12.	Test	
13.	Code	
14.	Compile	Database high level access programs
15.	Test	
16.	Code	
17.	Compile	Database low level access programs
18.	Test	
19.	Code	
20.	Compile	Database low level maintenance programs
21.	Test	
22.	Code	
23.	Compile	Database high level maintenance programs
24.	Test	

TASK 4 Test the database

Description

) Systematic testing of the database functions.

TASK 5 Document the database

Description

) Document the low level routines. It is not proposed that this information would be released as it would not be required by a user.

!) Document the high level routines. This information would be available, but would only be required by users wishing to access the database with their own programs.

3) Maintenance documentation. This information would cover such topics as file management, archiving, recovery and fault finding.

TASK 6 Detailed design of the editor

Description

Detailed description of its functionality

Design the user interface

System design

Program design

TASK 7 Write editor

Description

1)	Code		
2)	Compile	}	Initialisation program
3)	Test		
4)	Code		
5)	Compile	}	Editor
6)	Test		

TASK 8 Add final version of database access routines to the editor

Description

1) The initialisation system

2) The editor

TASK 9 Test the editor

Description

) Systematic testing of all the editor functions

TASK 10 Document the editor

Description

Prepare a user manual for the editor

N.B. It is intended that the system should be largely self-documenting in that it will tell the user what to do at each stage.

TASK 11 Design the River Quality Mapping system

Description

1) System design (modification of existing system)

2) Program design

TASK 12 Write the River Quality Mapping system

Description

1) Code

2) Compile

3) Test

TASK 13 Test the River Quality Mapping system

Description

) Systematic testing of the River Quality Mapping system's functions

TASK 14 Document the River Quality Mapping System

Description

) Prepare a river user manual for the River Quality Mapping system.

N.B. It is intended that the system should be largely self-documenting.

TASK 15 Design the Water Quality Reporting system

Description

1) System design

2) Program design

TASK 16 Write the Water Quality Reporting system

Description

1) Code

2) Compile

3) Test

TASK 17 Test the Water Quality Reporting system

Description

) Systematic testing of the Water Quality Reporting system's functions

TASK 18 Document the Water Quality Reporting System

Description

) Prepare a river user manual for the Water Quality Reporting system.

N.B. It is intended that the system should be largely self-documenting.

TASK 19 Purchase of initial equipment

Description

1) Agree with AW who purchases the equipment

2) Purchase the equipment

3) Initial set-up at IH for development

TASK 20 Installation at AW

Description

1) Set-up hardware

2) Set-up software

3) Load dictionaries

4) Load demo data

5) Load AW rivers for area 033

6) Develop reformatting programs* for data transfer

* These programs should be simple and it will probably be easier for AW to write them than IH since AW will be familiar with the formats. They have not been costed.

TASK 21 Training

Description

1) System objectives

2) System description

3) Setting up hardware

4) Setting up software

5) Setting up dictionaries

6) Using the editor

7) Using the River Quality Mapping system

8) System maintenance

TASK 22 Initial support

Description

This would entail practical help and advice on how to apply the system to the problems of AW

Appendix 6

Costs

All aspects of database design writing, testing and documentation	25,000
Editor - course of action to be decided allow	3,000
RQS mapping system	
design	725
write	3,500
test	725
document	725
WQ reporting system	
design	725
write	3,500
test	725
document	725
Installation at AW	
set up hardwar set up software load - dictionaries - rivers - demonstration data — allow 1 day each	725
Training	
2 x 2 man days + travel allowance	1,000
Support	
10 man days + 3 visits	1,800