DESIGN AND IMPLEMENTATION OF MICROCOMPUTER BASED WATER RESOURCES DECISION SUPPORT SYSTEMS

Marc P. Armstrong
Departments of Geography and Computer Science
316 Jessup Hall
The University of Iowa
Iowa City, IA 52242
BLAMMGPD@UIAMVS.BITNET

ABSTRACT

A general design strategy for building water resources decision support systems is outlined. The framework uses a modular representation of system functions which must be blended for successful system implementation. Specific emphasis is placed on data organization strategies.

INTRODUCTION

Water is an important resource. Because of its importance, water management practices are commonplace in many arid regions of the world. Increased attention is now being placed on managing water resources in humid regions, as well. This attention has had its genesis in two factors: 1) a growing recognition that water supplies are finite even in areas with seemingly vast stores of fresh water (Cohen, 1986); and 2) an acknowledgement of the role that pollution plays in the long-term viability of both ground and surficial water holdings.

As a result of these concerns, need has increased for timely information upon which to base water management decisions. Researchers have responded by implementing information systems built specifically to address water management issues. Some systems have been designed to support decisions about surface water (Guariso, et al., 1985; Johnson, 1986; Holsapple and Whinston, 1976; Hopkins and Armstrong, 1985), while others deal with ground water resources (Hendrix and Buckley, 1986; Monaghan and Larson, 1985). Although these individual systems are meritorious, the implementations are unrelated, and seem to employ an <u>ad hoc</u> approach to system design. The purpose of this paper is to resolve this problem by describing an overarching strategy for designing and implementing a microcomputer based water resources decision support system (DSS). By exploiting inexpensive microcomputer technology, the strategy may prove palpable to system designers and decision-makers charged with solving water management problems in both developed and developing countries.

The paper is divided into two main sections. The first section is concerned with developing a general framework for designing a water resources DSS. The second provides a consideration of specific issues pertaining to an example application of the framework.

A DESIGN FRAMEWORK FOR WATER RESOURCES DSS

The goal of a DSS is to help decision-makers in the processes of solving structured and, more importantly, semi-structured problems. Many spatial problems are semi-structured or ill-defined (Hopkins, 1984) because all of their aspects cannot be measured or modeled. This aspect of semi-structured problems necessitates human intervention, and therefore, solutions to semi-structured problems are often obtained by allowing a decision-maker to select and evaluate workable solutions from a set of alternatives. These steps are followed in an exploratory and sometimes heuristic fashion until an outcome acceptable to the decision-maker is reached. The system, therefore, must employ feedback loops to allow the user to evaluate the usefulness of solutions, and perhaps, to alter model parameters, or even to choose entirely different modeling strategies. To achieve these objectives, decision support systems normally use a variety of data types, and also rely on graphic displays to convey information to the decision-maker. Many systems also incorporate artificial intelligence principles to make them easy to use.

A DSS can be constructed from a set of linked software modules (Armstrong, Densham and Rushton, 1986). In a water resources DSS, the modules and the data stored within, can be organized in many ways depending upon institutional objectives and the nature of decisions that a system is designed to support. Despite this potential for organizational diversity, common design principles can be adhered to during the construction of any system. The framework described here has been adapted from Sprague and Carlson (1982). Its main components are:

1) Geometric Representations
2) Operations
3) Structure
4) Mechanism for Interaction.

Although each can be considered to be equally important, in this paper, the first three components will be discussed, and particular emphasis will be placed on the third (structure) component. It must be stressed that the ultimate objective of the system designer is to create a seamless, rather than modular, view to the DSS user. These modules, therefore, need not be separated in a real sense. Viewing the system in this way, however, facilitates software development tasks.

Geometric Representations

At the DSS design stage, choices must be made about topological referencing methods, and the degree of spatial precision used for analytical operations and for storing cartographic representations. A two-tiered approach to the organization of these data, such as that described by Hopkins and Armstrong (1985), can provide a flexible means for accommodating the topological and cartographic data. Hopkins and Armstrong, however, were concerned with a stream channel information system. The structure presented here is more general, in the sense that it explicitly accommodates interfluve information that is often critical in water management decision-making. Other relationships such as flow distance, are also readily accommodated in the two-tiered approach.

In this tiered approach, the main design elements are the stream channels, rather than the basins, because of an underlying need to efficiently specify and retrieve flow relationships. This main organizational tier forms a topological skeleton and provides for macro referencing capabilities with respect to the hydrological network. Although the skeleton provides a useful structural mechanism for general representations of database entities, the second, cartographic, tier provides their explicit descriptions; each·topologically referenced entity has coordinate information that is requisite for display and analytical functions.

Operations

The number and types of operations in a water resources DSS are controlled by a need for information upon which to base decisions, a need to select from alternative problem solving strategies, and a need to provide effective representations. Among the analytical operations often needed in a water resources context are:

* Production of summary statistics. These data are used in the course of producing environmental inventories and assessments for basin planning.

* Application of logical decision rules. Operations of this type are used to determine suitabilities from combinations of variables. The results are often used in assessing the impact of proposed development projects, and for basin planning.

* Hydrological modeling capabilities are important components of a water resources DSS, because they provide a mechanism for performing exploratory analyses. For example, by changing runoff parameters, impacts on hydrological characteristics can be determined for various development scenarios.

* From a cartographic standpoint, important operations allow simplification (Douglas and Peucker, 1973) or enhancement (Dutton, 1981) of stream traces for producing thematic maps at various scales.

These functions, and others, are obtained by retrieving and manipulating geometric and thematic information contained in the database. These data are then passed to modules designed to produce cartographic displays, graphs, and formatted reports. Operations are vital to a water resources DSS, because they provide the user with a tangible basis for validating decision-making outcomes.

Structure

The way in which information is organized in any computer system is a critical factor in its success or failure. The chosen structure must provide a means for capturing the fidelity of data relationships that must be accessible to solve either individual problems, or entire classes of problems. The storage structure also influences the user's conceptualization of the database, which in turn, influences the types of problems that a user will attempt to solve. At a most fundamental level, the implementation of the user view plays an important role in system performance.

The structure component of the water resources DSS design framework takes the form of a detailed database design and implementation strategy. An important component of a database is the adoption of a logical model to support representations and operations. Logical models vary in the types of data relationships that they support, and differ in methods for producing efficient linkages among database elements. The major logical database models can be placed into two families: operations-oriented (e.g. relational) and structure-oriented (e.g. network).

Miller (1984) has provided a structure based upon the operations-oriented relational model (Codd, 1982). The relational model, as it is now often implemented for microcomputers, may be unsuitable for DSS application development. Retrieval performance is slow, compared to alternatives, and it requires storage of redundant normalized data domains. Many microcomputer implementations of the relational model also are limited in their joining capabilities when compared to mainframe versions. Other problems with a purely relational approach to data modeling are recounted elsewhere (King, 1981; Sandberg, 1981).

Hopkins and Armstrong (1985) provide a water resources database structure that uses a network design. The network model employs fixed linkages to provide a mechanism for forming relationships among database entities (Olle, 1978). Paths specified by the database designer are used during retrievals. Although the network design is efficient with respect to the relational approach for retrieval types that are known to the database designer, performance may be degraded when alternatives unanticipated by the designer must be explored. The path dependencies then become a liability rather than an asset. Better alternatives are available.

In this paper, I provide a structure based upon the extended network model (Bonczek, Holsapple, and Whinston, 1976; 1984), a hybrid model that exhibits the retrieval performance characteristics of the network model, while providing much of the flexibility of the relational model. The extended network model bears some similarities to the network model; it differs mainly in implementation. Both models use set relationships among record entities in the database. The extended network model, however, provides for a number of advanced logical structuring capabilities that are especially useful for spatial data processing applications: many-to-many sets, recursive sets, and system-owned sets.

Many-to-many sets. The extended network model allows the direct specification of many-to-many (N:M) linkages between database elements. The direct, and thus, efficient, provision of N:M sets is useful, because spatial databases often contain entities and attributes that are linked inherently in many-to-many relationships. For example, coordinate chains constitute the piecewise approximation of more than one polygon (e.g. a shared border), and can be owned directly by both polygons in the extended network structure. The database designer or user, therefore, need not be concerned about the specification of polygon-chain-node pointer structures.

Recursive sets. Extended network structures also provide for recursive relationships, wherein records of a given type can own other records of the same type without having to traverse additional paths in an ownership tree structure. This feature is useful when describing topology, because it obviates the need for separate contiguity, or flow, record structures. For example, a water resources DSS must be able to support a series of data structuration capabilities that will permit rapid retrieval of flow relationships including: upstream, downstream, or tributary determination. Although these relationships could be calculated from three dimensional coordinates, that process is time consuming and error-prone. Because they are often invariant over the life of a database, hydrological relationships are easily determined from maps and stored in a recursive set. A set relationship of this type is formed when data are added to the database. The linkages are not computed "on the fly" (e.g. joins) as they are in operations-oriented approaches to the same problem.

System sets. The extended network model allows independent direct access of any record type by simply declaring it to be system-owned. This obviates the need for chaining through intermediate record types to retrieve information about database entities. If a simple spatial hierarchy, such as streams and sampling stations, is created, stations can be made members of a system-owned set. It is not necessary, therefore, to know the stream on which a station is located to retrieve information about that station. Note, however, that the original hierarchy can also be retained and used if, for example, it is necessary to determine all stations on a single stream.

Ease of retrieval can be gained by declaring many system-owned record types, and comes with only a minor penalty of incurring increased overhead storage (about four bytes per link) for each instance of a system-owned set in the database (Bonczek, Holsapple, and Whinston, 1984:107). This facility helps to provide a tabular, or relational-like view of the database.

DSS DESIGN APPLICATION

In this section an example application is outlined. It draws upon the design strategy from the previous section, and employs capabilities of the extended network model.

First, a general schema diagram is used to illustrate logical relationships in the database. Then a portion of an example schema is specified in a data definition language (DDL).

Schema Diagram

In Figure 1, the main organizational entity is the stream. Each stream, however, can be accessed in many ways to increase flexibility in terms of both jurisdictional referencing (e.g. stream identification for different governmental agencies) and the human interface (e.g. stream name). Note that two recursive sets are present for each stream - one each for tributaries, and when required, distributaries. These sets provide an effective means for encoding flow relationships.

A stream also has precedence over other entities (nodes, lines, areas) that exist either wholly or partially within the areal extent of its basin. Examples of these entities are: wells, transmission lines, and recreation areas. When data are organized in this way, entities are explicitly assigned to basins. Entities that extend across basins, however, can be handled by many-to-many relationships.

Each entity (e.g. a well) also explicitly owns its geometrical description in the form of chains or points. The use of many-to-many sets is a convenient way to structure chain-encoded polygon data. Each polygon owns many chains; each chain is owned by many (two) polygons. Likewise, each chain has two nodes (from, to) and each node, by definition, serves as a terminator for many chains.

In this general structure, each entity can have many attributes. For example, stream entities may have several bridges associated with it; it may also have information about a multitude of gauging stations, historical sites, and recreation areas. In Figure 1, these attributes are sorted by distance along the stream (RMI), by bank (left, right, both, instream) and by date. Of course other strategies exist; these are meant to be illustrative.

Data Definition

After the relationships among database elements have been designed in graphic form, they must be coded in a DDL (Figure 2) prior to implementation. In this example, the DDL syntax of MDBS III (Bonczek, Holsapple and Whinston, 1984) is used. It provides a rich database environment for a variety of microcomputer systems, and supports the extended network model. The intent here, is to provide the flavor of how a DDL specification is constructed; space limitations preclude a total description.

SUMMARY

A design framework for decision support systems can be adapted readily to water resources applications. The logical structuring facilities of the extended network model support the geometrical and operations requirements of the water resources DSS, and provide for data organization in a single, unified repository. Stream flow (topological) relationships are specified using recursive sets. Cartographic representations are stored using many-to-many sets. Attribute information is organized by date, bank, or along the linear dimension of a stream.

REFERENCES

Armstrong, M.P., Densham, P.J., and Rushton, G. 1986. Architecture for a microcomputer based spatial decision support system. Proceedings, Second International Symposium on Spatial Data Handling. Williamsville, NY: IGU Commission on Geographical Data Sensing and Processing.

Bonczek, R.H., Holsapple, C.W., and Whinston, A.B. 1976. Extensions and corrections for the CODASYL approach to data base management. Information Systems, 2: 71-77.

Bonczek, R.H., Holsapple, C.W., and Whinston, A.B. 1984. Micro Database Management: Practical Techniques for Application Development. Orlando, FL: Academic Press.

Codd, E.F. 1982. Relational database: A practical foundation for productivity. Communications of the ACM, 25: 109-117.

Cohen, S.J. 1986. Climatic change, population growth, and their effects on Great Lakes water supplies. Professional Geographer, 38: 317-323.

Douglas, D.H. and Peucker, T.K. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer, 10: 112-122.

Dutton, G.H. 1981. Fractal enhancement of cartographic line detail. American Cartographer, 8: 23-40.

Guariso, G., Rinaldi, S., and Soncini-Sessa, R. 1985. Decision support systems for water management: The Lake Como case study. European Journal of Operational Research, 21: 295-306.

Hendrix, W.G., and Buckley, D.J.A. 1986. Geographic information system technology as a tool for ground water management. Technical Papers, ACSM-ASPRS Annual Convention. Falls Church: American Congress on Surveying and Mapping.

Holsapple, C.W. and Whinston, A.B. 1976. A decision support system for area-wide water quality planning. Socio-Economic Planning Sciences, 10: 265-273.

Hopkins, L.D. 1984. Evaluation of methods for exploring ill-defined problems. Environment and Planning B, 11: 339-348.

Hopkins, L.D., and Armstrong, M.P. 1985. Analytic and cartographic data storage: a two-tiered approach to spatial decision support systems. Proceedings, Seventh International Symposium on Computer-Assisted Cartography. Washington, DC: American Congress on Surveying and Mapping.

Johnson, L.F. 1986. Water resource management decision support systems. Journal of Water Resources Planning and Management, 112: 308-325.

King, J.M. 1981. Evaluating Data Base Management Systems. New York: Van Nostrand Reinhold Co.

Miller, S.W. 1984. A spatial data structure for hydrologic applications. Proceedings, International Symposium on Spatial Data Handling. Zurich: Geographisches Institut, Universitat Zurich-Irchel.

Monaghan, G.W. and Larson, G.J. 1985. A computerized ground water resources information system. Ground Water, 23: 233-239.

Olle, T.W. 1978. The CODASYL Approach to Data Base Management. New York: John Wiley and Sons.

Sandberg, G. 1981. A primer on relational data base concepts. IBM Systems Journal, 20: 23-40.

Sprague, R.H. and Carlson, E.D. 1982. Building Effective Decision Support Systems. Englewood Cliffs, NJ: Prentice-Hall Inc.

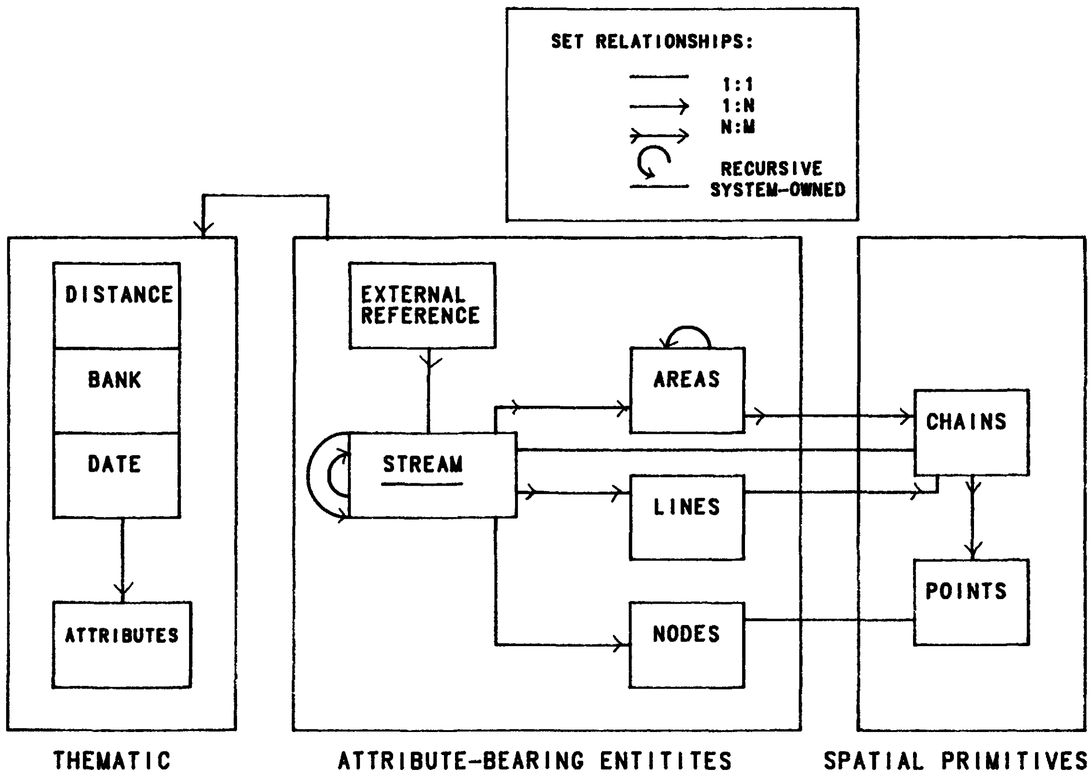

FIGURE 1: DATA ORGANIZATION FOR WATER RESOURCES DSS.

```
/*******  DATABASE  IDENTIFICATION AND SECURITY   *******/
/*                                                      */
database name is STREAMS
user is Granite with ROCK
user is Pillsbury with ROLL
/*                                                      */
/*******             RECORD SPECIFICATION         ********/
/*                                                     */
record name is STREAM
       item name is STREAMNAME string 20
       item name is DRAINAREA real 2
       item name is INFLOWRMI real 2
       item name is STREAMLEN real 2
record name is PSEUDONYM
       item name is STNAME  string 20
       item name is OTHERID  string 20
record name is WELL
       item name is STPLANEX real 3
       item name is STPLANEY real 3
       item name is FIPSCO unsigned 1
       item name is SECTION unsigned 1
       item name is TOWNSHIP string 3
       item name is RANGE  string 3
       item name is TOPELEV real 2
       item name is DEPTH real 2
       item name is H2OLVL real 2
/*                                                       */
/********             SET SPECIFICATION           ********/
/*                                                      */
set name is STREAMS  type is 1:N
       owner is SYSTEM
       member is STREAM order is FIFO
set name is TRIBS type is 1:N
       owner is STREAM
       member is STREAM order is FIFO
set name is EXREFS type is N:1
       owner is PSEUDONYM sorted ascending by OTHERID
       member is STREAM
set name is DEEPSUBJECT type is 1:N
       owner is STREAM
       member is WELL sorted ascending by COFIPS
set name is WELLDIR type is 1:N
       owner is SYSTEM
       member is WELL sorted ascending (TOWNSHIP, RANGE)
end
```

Figure 2. Schema definition in DDL.