OUTLINE OF A DECISION SUPPORT SYSTEM FOR AREA-WIDE WATER QUALITY PLANNING

Clyde W. Holsapple
Andrew B. Whinston

November 1975 WP-75-146

Outline of a Decision Support System for Area-wide Water Quality Planning

Clyde W. Holsapple* and Andrew B. Whinston**

Introduction

This working paper outlines requirements for an implementation of a computerized decision support system which addresses the technical aspects of area-wide water quality planning. The framework for this work is in the context of the environmental law adopted in the United States during 1972. This law, known as the Federal Water Pollution Control Act Amendments of 1972, specifies various requirements that both municipal and industrial discharges must eventually conform. By 1977 municipal waste treatment plants must have in place secondary treatment facilities and for industry it is necessary to utilize what is referred to as "best practical technology" for waste treatment. Under certain circumstances as described in section 303 of the law further treatment may be required to meet water quality standards. Section 208 of the Federal Water Pollution Control Act Amendments of 1972 calls for area-wide implementation of technical and management planning, with the objectives of meeting 1983 water quality goals and establishing a plan for municipal and industrial facilities construction over a twenty year period. Emphasis is placed on locally controlled planning, on dealing with non-point sources as well as point sources, and on consideration of both structural and non-structural control methods. The scope of present examination is limited to those aspects of technical planning which are amenable to implementation within the framework of a computerized decision support system.

In order to be designated under section 208, an area must be in need of a complex control program and must exhibit either impairment (water quality limited segments) or a need for preclusion of desired uses. An area-wide plan is constrained in that it must conform to the basin-wide plan, must take existing facilities plans into consideration, and must be capable of implementation from the managerial perspective. The plan may eventually act as a constraint on the issuance and conditioning of effluent discharge permits. By law each discharger must possess what is referred to as an N.P.D.E.S. permit that specifies the types and amounts of allowable discharge. The area-wide objectives of technical planning are to identify the priorities of water quality problems in the area; recognise constraints in methods for dealing with these problems; formulate alternatives for satisfaction of water quality goals; develop cost data for each alternative; select the least-cost alternative that is

---

*Graduate Student, Purdue University.

**Research Scholar, IIASA, Laxenburg, Austria feasible, given existing regulatory authority and qualitative restrictions; and periodically update the plan.

In order to meet these objectives, the planner must be able to store, manipulate, and retrieve large volumes of data. It is important that the data be organized in a manner that accounts for their interrelationships. The planner must be able to utilize a collection of mundane application programs which provide selective retrieval of a multitude of dara configurations; which offer statistical analyses, plots and projections; and which perform large scale simulations and optimizations. Each of these programs requires a particular configuration of data as input. In a comparatively unstructured decision making process, the types of reports that are needed tend to change rapidly; in traditional file oriented systems this necessitates the writing of a new report generator every time a new report type is called for. The typical local planner is not a computer programmer; indeed, the planner's time is presumably too valuable to be concerned with writing report generators, maintaining data files, and interfacing large scale application programs with data files. It is therefore proposed that the planner be provided with a decision support system which automatically handles the tedious, cumbersome task of data management and program interface, which provides organized data storage capable of representing all pertinent data relationships while obviating redundancy, and which can be controlled by the planner through the use of a non-procedural, English-like query language. The Generalized Planning System (GPLAN) described below provides the framework for the implementation of such a decision support system.

Before describing the attributes of GPLAN, it is useful to enumerate those capabilities which are needed for technical planningand the resources required to support those capabilities. In order to assist area-wide water quality planners, a decision support system should include the following capacities:

1) Selective retrieval of any configuration of stored data values
2) Prediction of water quality for all area reaches, given a particular area-wide treatment strategy for both point and non-point sources
3) Generation of alternative strategies that meet a specified set of water quality goals.
4) Production of waste load projections based upon land use plans, population statistics and employment statistics
5) Prediction of the effects of proposed land use plans on water quality
6) Determination of least cost alternative among those generated in 3) above

7) Permit basin-wide simulations in order to determine the effects of area-wide plans on each other
8) Permit regional monitoring of local planning activities

Such a system requires data of the following types:

1) Basin area descriptions
2) Quality goals
3) Point source and non-point source information
4) Cost information for the various types of treatment
5) Data on land use plans, population, and employment
6) Municipal facilities information (including design flow data)

Major application programs which the system utilizes are simulations to predict water quality (given a treatment strategy for point and non-point sources), models to predict the effects of alternative land use plans on water quality, and optimizations to select a least cost alternative from a set of alternatives.

GPLAN Description

GPLAN [1] is characterized by four primary features: 1) a data base for retaining all data, relevant to the planning problem, in an organized structure; 2) a data manipulation language which provides the means for manipulating data in the data base; 3) a set of application programs for processing the data; and 4) a query language with which the user can control the entire information system according to his requests. The first two features form the basis of the data management system (GPLAN/DMS) [2], while the latter two are fundamental components of the query system (GPLAN/QS) [3].

As an example of an application programs we will use MULQAL. MULQAL is a program that simulates the water quality characteristics of a stream. The types of data required by MULQAL can be grouped into the following categories:

1) Description of the river basin structure (junctions, bypass-piping, etc.)
2) River reach statistics (length, reaeration parameters, velocity parameters, etc.)
3) Pollution statistics (pollutant concentration, temperature, flow rate, conservative minerals)
4) Incremental flow statistics
5) Headwater statistics
6) Treatment levels of polluters

The outputs of MULQAL are water quality statistics and load allocations.

GPLAN/DMS was designed around the specifications of the CODASYL DBTG report of 1971 [4] and was implemented using FORTRAN as both the host language and as the language of implementation. FORTRAN was selected as the host language because many of the problems encountered in a planning environment involve the use of operations research techniques such as modeling, simulation, and optimization, and most programs written to solve such problems are currently implemented using the FORTRAN language. The high level FORTRAN language was chosen as the implementation language for the much needed machine independence quality. The GPLAN/DMS is currently being used on a CDC-6500 and an IBM 370/155, and is being installed on a Univac 1108. This feature of portability is critical to organizations with changing needs and changing computer environments.

The data base is stored on a random access device and contains all collected data that is pertinent to the user's needs. The description of a schema, detailing the logical structure of the data base, is prepared by the data base administrator using a Data Description Language (DDL). Figure 1 presents a pictorial representation of the water quality data base schena. Though details of the schema are omitted from the figure, it is useful for illustrative purposes. Note that GPLAN/DMS and GPLAN/QS are quite capable of supporting the complex network structures. The schema may be thought of as being a template which is utilized in buildings the data base and navigating through it.

In Figure 1, each box represents a record-type. Each record-type is composed of one or more data item types. For instance, the record-type labeled 'Reach' is composed of the data item types which identify the reach and give its state characteristics. Each record connecting two record-types indicates a set relationship, i.e., a set is merely a relation between two record-types. The arrow points from the owner of the set to the member. As an example, 'Basin' is the owner of the set of which 'River' is the member. For each record-type in the schema, there are many occurences of actual data of that type in the data base. For the record-type 'River' defined in the schema, there may be data base record occurences for all rivers in a state. A set denotes a one to many relationship between each occurence of the owner record-type and many occurences of the member record-type. Each occurence of river owns many occurences of reach. Similarly, each occurence of a river may own several segment occurences. A standard requirement is that the system owns all record-types (this is not depicted in Figure 1). In general, a schema is set up so as to reflect hierarchical or network structures that exist in the real world. An abbreviated sample of the DDL description is shown in Figure 2. Each data item has the attributes of type (INTEGER, REAL, CHARACTER, BINARY, COMPLEX), size (number of characters of words of storage), and replications. For instance, the data item BSID is an integer whose length is one word which occurs only once; the data item NACO is character in type, has a length of ten characters, and can be replicated up to eight times

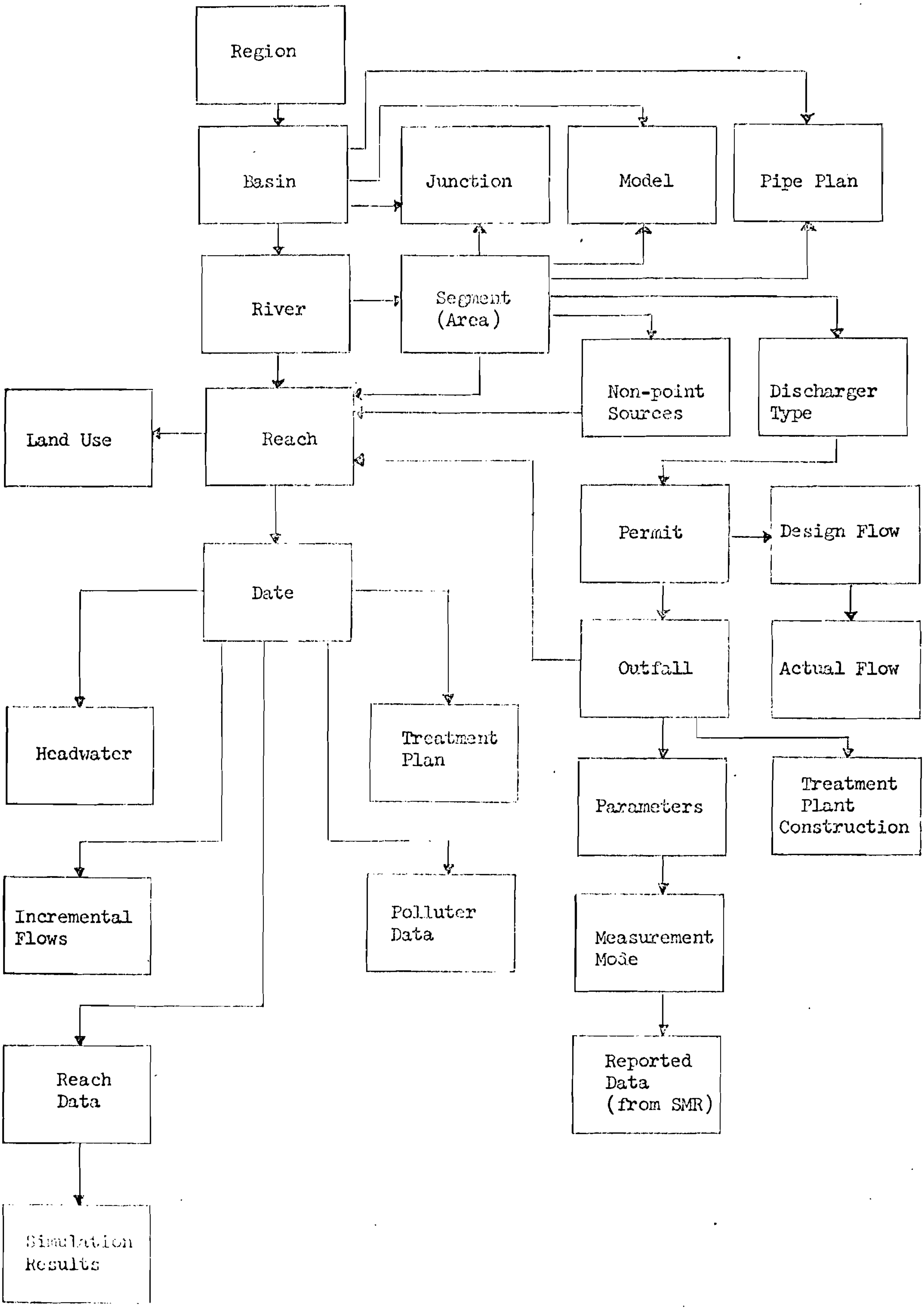

Figure 1. Overview of Logical Structure

(since MULQAL can handle up to eight conservative minerals). Since this DDL is used primarily for the FORTRAN language, arrays (such as NACO) are permitted, but the repeating group capability is not yet implemented. Each data item is given a unique four character name; however, the GPLAN system allows the user to define synonyms of up to sixteen characters. The data items are grouped together to form record-types, each of which has a unique four character name. The logical data structure is completed by defining various set relationships among record-types. This DDL description is used by a Data Description Language Analyzer in order to generate the actual schema that is used in creating and accessing data base.

The other main component of the data management system is the Data Manipulation Language (DML) which consists of about sixty commands which perform the following kinds of functions: 1) opening and closing of the data bases; 2) creation and deletion of record occurences; 3) setting of currency indicators; 4) addition and removal of record occurences from sets; 5) retrieval and storage of data from current record occurences; and 6) searching through sets for particular record occurences. Each DML command is a FORTRAN subroutine call. This permits direct user interaction with the data base via programs in the FORTRAN host language. In general, this is the level at which the data base administrator interacts with the data base,whereas the planner's interface is through a high-level query language. The DML provides the link between the data base on the one hand and the query language and application programs on the other, thus allowing the planner to interact by means of non-procedural, natural language commands that require no knowledge of the physical or logical data base structure.

The set of application programs consists of a library of programs that are standard for the GPLAN query system plus a library of special programs that are exclusively applicable to a particular data base. The standard library includes a standard report generator, plot routine (allowing up to five variables on the y-axis), regression, statistics (number, minimum, maximum, average, standard deviation, variance). An example of a special library routine for the water quality data base is MULQAL. The user may request execution of any application program through the query language, with the system providing the appropriate data extraction from the data base and reformmatting of the data for use by that program. The methods for interfacing special application programs with the data base is discussed later.

The GPLAN query system allows the simple, but effective, utilization of the data base and the automatic interfacing of models with data. The query system may be used in either interactive or batch mode. In batch mode, the queries are entered on cards. In interactive mode, queries are entered at a terminal and output is given at the terminal or at a high-speed line printer, according to the user's needs. GPLAN/QS allows both selective and unconditional retrieval. The query syntax is:

<COMMAND> <FIND clause> <CONDITIONAL clause>

or alternatively,

<CONDITIONAL clause> <COMMAND> <FIND clause> .

The COMMAND denotes which application program the user desires to have executed. The FIND clause specifies what data are to be retrieved from the data base for use by the application program. This retrieval is subject to all conditions satisfied in the CONDITION clause. Any combination of data otems in the data base may appear in the CONDITION clause. The query system allows the use of synonyms and noise words in the query language. This feature makes a query easier and more natural to use; it also enables interested observers to understand the meaning of the query. Typical queries are:

```
LIST REACH. NAME, REAERATION. PARAMETER AND REAERATION.
EXPONENT FOR DATE = 11-1-75 AND REACH. LENGTH < .9
PLOT DATE VERSUS BOD. CONCENTRATION, AMMONIA. CONCENTRATION
AND DO. CONCENTRATION FOR REACH. NUMBER = 53
STAT NITRO. DEOXIDATION WHEN TEMPERATURE > 22
```

A more detailed description of the GPLAN/QS attributes is presented in a separate section.

A general overview of GPLAN's system flow may be seen in Figure 3. The GPLAN query system prompts the user for a query. When the query has been read and parsed, a set of control routines initiates DML commands to retrieve data through the data management system from the data base, as specified by the query's FIND and CONDITIONAL clauses. The retrieval data is written on an extraction file in a format that is usable by the application program specified in the COMMAND. At this point, execution passes to the designated application program which reads data from the extraction file. Output is returned to the user's terminal, to an auxiliary file, or to the data base. The user is then prompted for another query, at which time he can request supplemental processing of the data collected by the previous query, or retrieve and process a new set of data.

Interfacing Special Application Programs

There are several ways to add a special application program to the standard GPLAN library. These range from the insertion of DML commands into the code of the application program to the automatic generation of additional control code that governs the data extraction procedure for the particular application.

One way to add MULQAL to the system is to replace all READ statements in the MULQAL program by appropriate DML commands for retrieving needed data from the data base. Using this method, a query to run MULQAL results in the control routine transferring control to the application program which performs its own extraction of data from the data base.

A second method involves writing a control module that specifically handles data extraction for MULQAL. This module is added to the control routine. A request for a MULQAL run results in this control module writing required data on to the extraction file and a subsequent transfer of control to MULQAL. A sampling of code from this module is shown in Figure 4. It can be noted that this extraction routine makes extensive use of the Data Manipulation Language. The FMSK command finds the member record occurence of the set ALLB whose key (BSID) is equal to the contents of the FORTRAN variable IBASIN. The contents of this variable is determined by the user's query. SOM is a DML command that sets the current owner record occurrence of one set based upon the current member occurence of another set. For instance, the third use of SOM makes the current member occurrence of the set ALLB, the current owner record occurrence of the set MPIB. Once again FMSK is used, this time to find the member record occurrence of the set MPIB whose key (MOID) is equal to the content of the FORTRAN variable MDLID which is set according to the user's query. Having located the proper model for the proper basin, data items can be extracted from the current occurrence of the record-type BASN and the current occurrence of the record-type MODL. This is accomplished using the GFM command. The first use of this command gets the data item field BSNA from the current member of the set ALLB abd deposits this information in the first five words of the FORTRAN variable IDATA. Subsequent use of GFM fill the array IDATA with appropriate fields from the current member of the set MPIB. The argument IERR is a check for inconsistency between the data base and the command. Finally, the data is written from IDATA onto the extraction file. The extraction routine then repeats the same type of process in order to extract the next kind of information that is needed as input to MULQAL, and so forth. When the extraction file has been completed, it serves as the input medium to MULQAL.

The third and fourth methods are automated versions of the first and second. The third relies on a routine which, given a schema description and MULQAL code as inputs, produces a version of MULQAL in which FORTRAN input statements have been replaced by the proper DML commands. The fourth method depends on an algorithm which uses a description of MULQAL's data needs and a description of the schema, in order to generate an appropriate control module. The automated addition of special application packages is currently under development by the GPLAN group.

No matter what method is used, the interfacing of a special model with the data base is a one-time operation. A sample query for executing MULQAL is:

```
RUN MULQAL FOR BSID = 1, DATE = 11-1-75, MOID = 1,
RNUM = 1, JCOD = 1, PLAN = 1, TPLN = 1 AND SIMR = 4
```

This will run MULQAL for basin number one, for measurement number (RNUM) one taken on the date 11-1-75 and using model number one, piping plan one and treatment plan one. If JCOD is one, then this is to be a simulation only; if it is two, then an optimization run, using NONLIN, is being requested. Results of the simulation are stored in the data base and identified as being the fourth simulation of this type.

Attributes of the GPLAN Query Language

Perhaps the most outstanding feature of the GPLAN query language, aside from its ease of use, is its capacity to support queries directed not only at tree structured data bases, but at network structured data bases as well. The initial version of the query system was able to handle tree structured data bases only. In the process of implementing various information systems within GPLAN framework, it became apparent that in order to express all needed interrelationships among data items, a network was sometimes needed. Since the GPLAN data management system was implemented using doubly linked lists to represent sets, the extension to complete networks was entirely compatible with the existing DML commands. Though the problems of navigation through a network structure were more complex for the data base administrator and other persons desiring to use DML commands, the critical problem was that the initial algorithm used by the query language to determine a path through the data base was designed strictly for a tree structure. Development of a new algorithm led to the extension of the query system to its current capacity for searching through networks.

As the question of querying a network data base was being considered, it became apparent that the work being done on relational data bases [5], [6], [7] by Codd et al might be of some use in resolving this problem. Careful analysis indicated that an efficient method of implementing a relational data base capability would be through the use of a network structure. A paper by Bonczek, Haseman, and Whinston [8] discusses how such an implementation has been performed. The network implementation differs in two major ways from the original tree implementation. Whereas a tree structure possessed only one entry point, the root node, from which to begin a search for data items, the network has multiple entry points. For instance, in Figure 1, each record type is owned by the system; hence there are twenty-six possible points of entry into the data base. A second and related distinction is that there exist many more paths to answer a given query in a network, than exist in a tree structure.

Further investigation of the relational approach resulted in the development of an extension to the existing query system that allowed querying of network structures and which was shown to be relationally complete [9], [10]. The outcome of this excursion into the relational data base area was the demonstration that the GPLAN query language does indeed have all of the retrieval power of any of the languages proposed for relational data bases, and is more natural for the non-programming user while it operates on a network data base. This allows the user to view the data base as either a network or as a relational structure. In addition, the GPLAN system is operational (not just theoretical) and does not become ineffective as the data volume becomes large. It should be noted that the approach of allowing the user to view the network data structure as a relational data base is consistent with the CODASYL view that the data base should be accessible by alternate classes of users with different languages, functions, and capacities.

Implementation of a 208 Decision Support System

Regardless of the type of system used, whether manual or GPLAN, data and models must be available. Given these, a decision support system can be designed. The perspective presented here is that this design is an ongoing process; the system must be adaptable, flexible and capable of handling volatility in the types of models use, in the types of data required and in the values of data occurences.

The data and models are highly interdependent. There are physical technical and monetary constraints on the collection of data, with regard to both types and data detail. Data is usable according to the models which are available to analyze it. On the other hand the use of models is limited by the availability of data and to some extent by the complexity and size of the model and by the system available for supporting the model.

Given a local planning board's decision regarding the resolution level from which it intends to attack the area-wide water quality problem, it can be determined what data must be collected, what methodologies are to be used in collection, and what application programs are necessary. Parenthetically it must be pointed out that the level of resolution may depend upon such factors as the technical expertise of the local planners, the availability of funds, the nature of existing data, and the characteristics of the existing (or anticipated)decision support system. Having settled managerial issues and hydrological methodology problems to some extent, attention can focus upon the development of the decision support system. Given data and models, these can be integrated into the GPLAN framework.

The initial step involves devising the appropriate logical data structure. Collected data, after passing through data validation routines, is then loaded into the data base. At this juncture the user may query the data base, using any command in the standard library. MULQAL, the Multiple Quality River

Simulation model, is currently interfaced into the GPLAN/QS as a special library. Thus the user may also request this simulation run, by typing a single command at the computer terminal. Other applications may be added to the special library in the ways already indicated. It should be noted that the addition of applications may be an ongoing process. For instance, if after using the system for a period of time the planners decide to augment the special library, the additional models can be interfaced with the GPLAN/QS in the one-shot operation already described; subsequent to this a new program may be executed by submitting a single query.

Another important area of system adaptability relates to changes in date. Basically there are two kinds of modifications. First there are additions, changes, and deletions of values of existing data item types (i.e., those defined in the schema). Second there are modifications to the schema itselg which include such contingencies as the addition of data item types, the addition of data item types, the addition of record types and the alteration of set relationships. There are two methods for handling the first variety of modification. The data base administrator can write a DML program to perform the necessary updating; this procedure is generally used when an update problem is relatively large in scale or is repetitive in nature. Small, non-repetitive modifications are best handled through the query language bu usage of ADD, CHANGE and DELETE commands; these commands are currently in the process of implementation. These two methods of revision provide the data base administrator with flexibility in treating the problem of keeping data values up to date.

In order to change the logical structure of an already existing data base there are two major approaches. The first involves destruction of the existing data base. The new schema is defined with DDL, the programs which load the data are revised to reflect the new schema, and the entire data base is loaded using this new logical structure and the revised laod programs. For a large data base this is a cumbersome, time-consuming procedure and may necessitate the data base being dumped before reloading. The second approach, which is currently under investigation, permits a dynamic restructuring of the schema without destruction of an existing data base and subsequent reloading. The data base administrator can modify the existing logical structure through the use of either DML or query commands and adjust data values associated with restructured schema with ADD, CHANGE and DELETE commands or with the DML.

In order to obviate the necessity of devising specialized load programs for loading data into the base, a single generalized load program is being developed. This will facilitate data base creation and modification. The generalized loader accepts data files, logical descriptions of the data file formats, and the schema of a network data base as inputs and proceeds to load data values from the data files into the network data base.

Another important factor to consider is the scope of the 208 system with the GPLAN framework. In this connection there are many possible alternatives, ranging from a collection of relatively small data bases (one data base for each area), to moderately sized data bases (one data base for all designated areas in a basin), to one or a few very large data bases (e.g. one data base for each EPA region). Another dimension of scope involves the degree to which N.P.D.E.S., construction grant and land use information is integrated into the data base. Figure 1 presents a data base that is moderately broad in scope, in that it permits basin-wide analyses, in addition to area-wide (segment) analyses; it includes permit data and treatment plant construction information. Suppose that through the use of MULQAL and various cost and projection models, all area-wide planning agencies have arrived at respective local treatment strategies. Then regional administrators can monitor their activities by means of usage counters built in to the data base and by submission of queries that retrieve local plans. Furthermore, the water quality of the entire basin predicated upon current local strategies, can be simulated in response to a single query. This could permit the identification of the effects of various local strategies upon each other and upon the basin as a whole.

In this example there are two levels of data base entry: the segment level and the basin level. Typically, a group of area-wide planners is concerned with its own particular segment and maintains control over the data base information for that segment. The issue of restricting a user's access to certain portions of the data base in order that the user need not be confronted with data that is irrelevant to the local problem and so as to preserve data integrity, is an issue of data security. A method for treating this issue within the GPLAN framework is detailed in a paper by Cash et.al. [11].

Some examples of possible queries have already been presented. It is important to emphasize that GPLAN/QS allows retrieval of any configuration of data items using conditions on any combination of data items as the criteria of retrieval. Also GPLAN/QS is quite capable of accomodating special application programs in such a manner that submitting a query causes them to be executed, using any specified pertinent set of data from the data base.

Summary

This paper has addressed factors involved in the development of a decision support system for area-wide water quality planning. In particular, the Generalized Planning System has been presented as a framework for such a system. Given data and a special group of models for analyzing the data, these can be integrated into the GPLAN framewotk in order to provide timely, reliable and complete information; at the same time we recognize the need for a natural easy-to-use planner/system interface and the need for system flexibility that allows adaptation to changing conditions.

	NAME	TYPE	SIZE	MAX	COMMENTS
RECORD	BASN				BASIN
ITEM	BSID	INTE	1	1	BASIN ID
ITEM	BSNA	CHAR	20	1	BASIN NAME
ITEM	E	REAL	1	1	MEAN ELEVATION
RECORD	JUNC				JUNCTION IN BASIN
ITEM	JUID	INTE	1	1	JUNCTION ID
ITEM	UPR1	INTE	1	1	UPSTREAM RIVER ID - 1
ITEM	USC1	INTE	1	1	UPSTREAM REACH ID - 1
ITEM	UPR2	INTE	1	1	UPSTREAM RIVER ID - 2
ITEM	USC2	INTE	1	1	UPSTREAM REACH ID - 2
ITEM	DSRN	INTE	1	1	DOWNSTREAM RIVER ID
ITEM	DSCN	INTE	1	1	DOWNSTREAM REACH ID
RECORD	RIVR				RIVER IN BASIN
ITEM	RVID	INTE	1	1	RIVER ID
ITEM	RVNA	CHAR	20	1	RIVER NAME
RECORD	MODL				MODEL PARAMETER IN BASIN
ITEM	MOID	INTE	1	1	MODEL ID
ITEM	TEQ	REAL	1	1	BASIN TEMPERATURE
ITEM	TH1	REAL	1	1	THETA 1
ITEM	TH2	REAL	1	1	THETA 2
ITEM	CKT	REAL	1	1	TEMPERATURE EXPONENT
ITEM	NACO	CHAR	10	8	NAMES OF MINERALS
ITEM	CGO	REAL	1	8	GOALS OF MINERALS
ITEM	NIN	INTE	1	6	OUTPUT PARAMETERS
ITEM	NOUT	INTE	1	10	OUTPUT PARAMETERS
ITEM	JCOD	INTE	1	1	JCODE FOR MULQAL
ITEM	NCOM	INTE	1	1	NUMBER OF MINERALS
ITEM	COST	REAL	1	1	OPTIMAL COST
ITEM	E1	INTE	1	1	EXPANSION
ITEM	E2	INTE	1	1	EXPANSION

	NAME	ORDER	KEY	
SET	ALLB	SORT	BSID	ALL BASINS
OWNER	SYST			SYSTEM
MEMBER	BASN			BASIN
SET	JUIB	SORT	JUID	ALL JUNCTION IN BASIN
OWNER	BASN			BASIN
MEMBER	JUNC			JUNCTION IN BASIN
SET	RVIB	SORT	RVID	ALL RIVERS IN BASIN
OWNER	BASN			BASIN
MEMBER	RIVR			RIVER IN BASIN
SET	MPIB	SORT	MOID	MODEL PARAMETERS BY DATE
OWNER	BASN			BASIN
MEMBER	MODL			MODEL PARAMETERS

Figure 2. Sample DDL

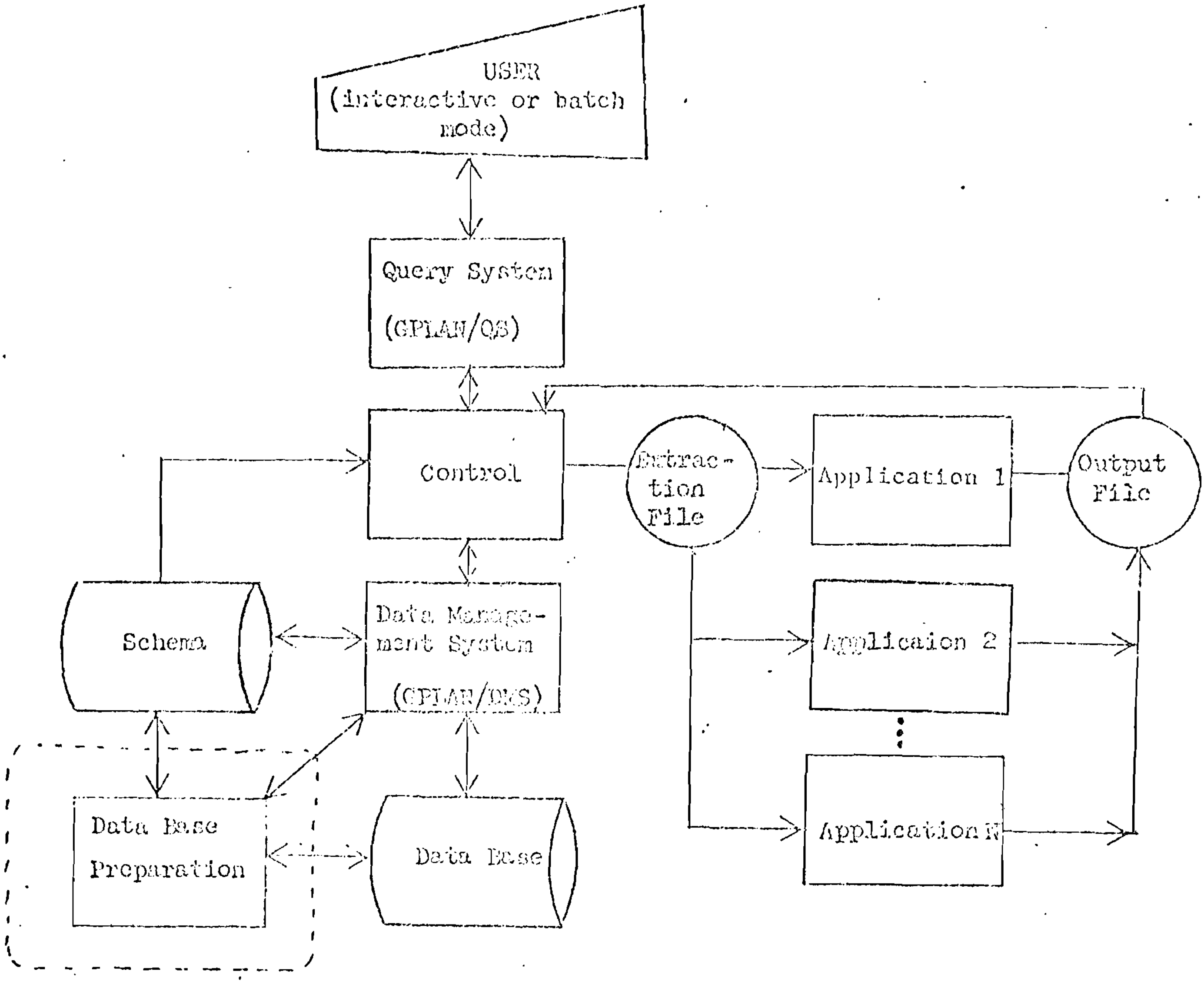

Figure 3. GPLAN System

```
          IERR = 0
          CALL FMSK (4HAUB, IBASIN, IERR)
          CALL SOM (4HJUIB, 4HALLB, IERR)
          CALL SOM (4H RVIB, 4HALLB, IERR)
          CALL SOM (4HMPIB, 4HALLB, IERR)
          CALL SOM (4HPIIB, 4HALLB, IERR)
          CALL FMSK (4HMPIB, MDLID, IERR)
          IF (IERR.NE.0) GO TO 900
          CALL GFM (4HBSNA, 4HALLB, IDATA, IERR)
          CALL GFM (4HE   , 4HALLB, IDATA(6), IERR)
          CALL GFM (4HTHI , 4HMPIB, IDATA(7), IERR)
          CALL GFM (4HIEQ , 4HMPIB, IDATA(8), IERR)
          CALL GFM (4HCKT , 4HMPIB, IDATA(9), IERR)
          IF (IERR.NE.0) GO TO 900
          .
          .
          .
900 error message
          .
          .
          .
```

Figure 4. Sample Code for MULQAL Extraction Routine

References

[1] Nunamaker, Jr., J.F., Swenson, Donald, and Whinston, A.B., "Specifications for the Development of a Generalized Data Base Planning System," Proceedings of the National Computer Conference, New York City: AFIPS Press, (June, 1973).

[2] Bonczek, R.H., Cash, J.I., Haseman, W.D., Holsapple, C.W., and Whinston, A.B., Generalized Planning System/Data Management System (GPLAN/DMS), User's Manual, Version 3.o, Krannert Graduate School of Industrial Administration, (September, 1975).

[3] Haseman, William D., Lieberman, Arthur Z., and Whinston, Andrew B., Generalized Planning System/Query System, User's Manual, Krannert Graduate School of Industrial Administration, (January, 1975).

[4] CODASYL Committee, Data Base Task Group Report, Association for Computing Machinery, (April, 1971).

[5] Codd, E.F., "Normalized Database Structure: A Brief Tutorial," Proceedings 1971 ACM Sigfidet Workshop on Data Description, Access and Control, ACM.

[6] Codd, E.F., "Further Normalization of the Database Relational Model," Database Systems,Vol. 6, Prentice Hall (1972).

[7] Date, C.J., An Introduction to Database Systems, Addison Wiley (1975).

[8] Bonczek, R.H., Haseman, W.D. and Whinston, A.B.,"Structure of a Query Language for a Network Data Base"(July, 1975).

[9] Chamberlain, D.D. and Boyce, R.F., "SEQEL - A Structured English Query Language," Proceedings ACM Sigfidit Workshop, Ann Arbor, Michigan, (May, 1974).

[10] Bonczek, R.H., Haseman,W.D., and Whinston, A.B., "Network vs. Relational Data Base Implementation," (August, 1975).

[11] Cash, J.I., Haseman, W.D. and Whinston, A.B., "Security for the GPLAN System," submitted to International Journal of Information Systems, (August, 1975).