WATER QUALITY MANAGEMENT AND INFORMATION SYSTEMS

by

A. B. Whinston

September 1975

PURDUE UNIVERSITY
WATER RESOURCES RESEARCH CENTER
WEST LAFAYETTE, INDIANA

WATER QUALITY MANAGEMENT
AND
INFORMATION SYSTEMS

by

A. B. Whinston
Krannert School of Industrial Administration

The work upon which this report is based was supported by
funds provided by the United States Department of the Interior
Office of Water Research and Technology, as authorized
by the Water Resources Act of 1964 (PL 88-379)

Period of Investigation: July 1, 1973 to June 30, 1975

Completion Report OWRT-A-031-IND
A DATA BASE SYSTEM FOR POLLUTION CONTROL

Technical Report No. 68

Purdue University Water Resources Research Center
West Lafayette, Indiana
September 1975

FOREWORD

This report provides a general discussion of data base management in the water quality area, and presents salient aspects of the continuing evolution of GPLAN. The motivation behind this evolution is the effort to provide the decision maker with timely, complete, and reliable information and to allow access to such information in a simple, non-technical, straightforward manner. For illustrative purposes, both chapters of this report refer to a system for determining a least-cost solution for raising the water quality characteristics of a river to meet specified standards. The first chapter describes the status of the GPLAN implementation of such a system as it existed a year ago. In order to reflect the new and wider perspective of those attributes which a decision support system should possess, Chapter 2 details the current implementation of a system for determining an optimal treatment strategy.

Current efforts are building towards the implementation of a decision support system which addresses the technical aspects of area-wide water quality planning. Section 208 of the Federal Water Pollution Control Act Amendments of 1972 calls for the 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 source pollution, and on non-structural control methods. The scope of the 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 require 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 must conform to the basin-wide plan, must take existing facilities plans into consideration, and will act as a constraint on the issuance of N.P.D.E.S. permits. The area-wide objectives of technical planning

are to identify the priorities of water quality problems in the area; recognize constraints in the methods for dealing with these problems; formulate alternatives for satisfying water quality goals; develop cost data (including opportunity costs) for each alternative; select the least-cost alternative that is feasible, given existing regulatory authority and qualitative constraints; and periodically update the plan.

The decision support system should have the following capabilities: 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 effect of proposed land use plans on water quality; 6) determination of the least-cost alternative among those generated in 3) above; and 7) retrieval of current construction priorities and time schedules. All of these features are used by the local planning agency and feature 7) can be used by the EPA for monitoring the local agency's activities.

Such a system requires data of the following types: a) an area description; b) quality goals; c) point source data; d) non-point source data; e) cost information for the various types of treatment; f) data on land use plans, population and employment; and g) design flow of municipal facilities.

Application programs needed are simulations to predict water quality (given a treatment strategy for point and non-point sources), to predict waste loads, to demonstrate the effect of a land use plan on water quality and an optimization program to select a least-cost alternative from a set of alternatives.

ABSTRACT

Title: WATER QUALITY MANAGEMENT AND INFORMATION SYSTEMS.

Principal Investigator: Dr. A. B. Whinston

This report presents an approach to a data management system that will aid planners in the implementation of a decision support system which addresses the technical aspects of area-wide water quality planning. The basis is the systematic study of the information flows and their relationships to the goals of the water planning group. The study involved a data processing program for the Technical Information Section of the Indiana Stream Control Board. The data required for the system includes (1) an area description, (2) quality goals, (3) point source data, (4) non-point source data, (5) cost information for the various types of treatments, (6) data on land use plans, population and employment, and (7) design flow of municipal facilities. The study was an application of the GPLAN (Generalized Planning System) developed at Purdue University. The system is capable of supporting data bases and models for a wide variety of planning problems in addition to water quality management. The GPLAN is a valuable planning tool in that it automates the interfacing between data and models, and it offers a non-procedural, English-like interface between the planner and his computerized resources.

Descriptors: *Data Processing, *Decision Making, *Planning, *Water Quality Standards, Basic Data Collection, Data Storage and Retrieval, Management

Identifiers: Generalized Planning System

TABLE OF CONTENTS

	<u>Page No.</u>
Foreword	ii
Abstract	iii
Table of Contents.	v
List of Tables	vi
Chapter 1	
I Introduction.	1
II Designing of a Management Information System.	1
III Information Flow Model.	2
IV Complexity of a System.	6
V Information Flow of the Indiana Stream Pollution Control Board	9
VI Complexity of Data Management Systems	18
VII GPLAN	24
VIII Summary	32
References.	34
Chapter 2	
I Introduction.	35
II Structural Complexity	36
III Water Quality Management.	39
IV An Illustration of Complexity of Data Management Systems.	43
V GPLAN Description	47
VI Interfacing Special Application Programs.	55
VII Attributes of the GPLAN Query Language.	60
VIII Conclusions	62
References.	63

LIST OF FIGURES

	<u>Page</u> <u>No.</u>
Chapter 1	
Figure 1 Information Flow	4
Figure 2 Complexity Levels	8
Figure 3 MULQAL Information Flow	14
Figure 4 NONLIN Information Flow	16
Figure 5 Cost Model Information Flow	17
Figure 6 Modeling Information Flow	19
Figure 7 GPLAN Framework	25
Figure 8 Structure of Data Management System	27
Figure 9 Pollution Data Base Structure	28
Chapter 2	
Figure 1 Levels of Complexity.	37
Figure 2 MUQAL Information Flow.	42
Figure 3 NONLIN Information Flow	44
Figure 4 Overview of Modeling Information Flow	45
Figure 5 Data Base Structure	49
Figure 6 Sample DDL.	51
Figure 7 GPLAN System.	56
Figure 8 Sample Code for MULQAL Extraction Routine	58

WATER QUALITY MANAGEMENT AND INFORMATION SYSTEM

Chapter 1

I. Introduction

In recent years public interest in environmental management has increased significantly. In the field of water quality management the 1972 Water Quality Act greatly increased the powers and obligations of the states to protect and enhance water quality as one part of environmental improvement. The general direction of the legislation has been to establish common federal guidelines with the more specific control and enforcement delegated to the individual states.

The purpose of this chapter is to study the data processing problems of the Technical Information Section of the Indiana Stream Pollution Control Board (ISPCB). We will proceed by stating the broad objectives that such a system is perceived to have and then try to link these objectives to specific outputs that are obtainable. These outputs may be in the form of reports, a listing of data resulting from a specific query or request, or as the result of a complex analysis via models interacting with a data base. Later on we shall suggest a notion of complexity that will measure the degree of difficulty in developing general output.

II. Designing of a Management Information System

The first step in designing an information system involves an analysis of the decision-making process of the group which will be

using the system. Obviously, all the decisions which are required to be made by the group must first be determined. Although this task may seem trivial at first glance, in most situations a fair amount of time is required to actually document every decision which will require input from the information system.

The second step involves analyzing the information requirements for all of those decisions determined in the first step. Information requirements will include both raw data and data processing routines. The data processing routines will perform jobs ranging from simple algebraic and statistical computations to possibly the application of sophisticated models. These models may already exist or may have to be constructed as a part of the information system. The precise flow through the information system can then be determined since each processing routine would have a given set of inputs and a given set of outputs.

The third step involves the actual design of the information processing system and the development of the associated data structures for that system. Since the information flows have already been determined, a system can be designed or selected which will most efficiently operate with the data.

The issue that will be discussed in this paper in detail is the development of the information flow model and analysis of the complexity of a system which is necessary to support that information flow.

III. Information Flow Model

The general information flow through any planning organization can be separated into a group of tasks or functions through which the

information must flow. The most general model of an organization consists of an input of raw data, some processing of that data, and the result in the form of an output of information.

Figure 1 represents a more complex view of the information flow, where the three parts of the general model are broken down into greater detail. Step 1 of the information flow involves the extraction of desired data from the "real world." The data extracted consists of those variables which will be used at a later time for monitoring and controlling the current status of the "world" and for planning for the future of the "world." Although this data will be collected from many different sources both internal and external to the organization all the data should pass through one data collection group which will have the responsibility of insuring against collecting redundant data and classifying the collected data.

The next step involves the encoding of the data to a form which can be used by the computer. The desired goal in this step is to minimize the number of human transcriptions required between the time the data is first observed and when it is actually stored in the computer system, in order to help minimize the errors which can enter into the data.

The third step in the process involves the varification of recorded data to locate obvious errors, and the determination of missing data. This varification can range from simply checking permissible bounds to the extent of running time series regressions or performing clustering analyses to determine which data items should be checked by hand and possibly remeasured. This step is extremely

Figure 1 Information Flow

critical since the rest of the system will assume clean and accurate data exists. Often, the data that is collected will have much information missing. In this case, as in the case where the data is inaccurate, there is little or no benefit to the user.

The first phase of the information flow is completed by step 4 when the data is stored into a data base. The type of data requirements of the organization should determine how the data base will be logically structured as well as where the data will be physically stored. All the data for the organization should be stored into one data base in order to avoid redundancy of data. If each group in the organization retrieves its information from a common base, problems of conflicting information that may arise can be eliminated.

The second phase of the information flow process concerns the extraction of the data from the data base into usable information. A key component in this process is the user of the data. The user determines what types of information is required to answer his desired questions. He must then formulate this request into the form of a query on the data base. The interaction between the user and the computer is a vital link to achieving the desired goals in the decision-making process.

The sixth step involves processing the specific query by a data management system (DMS) which will determine what data is required to be extracted from the data base to answer the query. The DMS should then proceed to extract the data and prepare it for the appropriate processing program. The complexity of the permitted questions will determine the complexity of the DMS and the processing programs which

follow. The question of complexity will be discussed later. In the seventh step, the data that has been extracted is then processed by the DMS, converting it by some specified method into usable information. This processing may involve simple report generators, it may require running one or more programs used for monitoring and determining control information, or it may provide planning modifications to the total system. The result of this step is the information requested by the user for making a particular decision.

The final phase of the information flow involves the feedback of information as a result of analyzing the requested outputs. The action requested might be related to adjusting certain control variables in the system being analyzed in order to obtain the desired results. A second type of activity which might exist is that the actual data being collected might be modified so that variables can be added when an unforeseen problem arises. Without the feedback path for information, there would not exist the capability to control the process being analyzed.

IV. Complexity of a System

Once the information flow is determined we must address the question of designing the information system. The components of the system consist of a data base and applications programs. The programs include various types of models of the environment and routines that carry out operations on the data such as determining optimal regression equations based on elements of the data base or determining optimal values of control variables defined within the information system.

It will be assumed that a DMS will be present for controlling the system and handling the interactions among the components. Different ways of organizing the information system with respect to the data base and application programs imply different levels of generality and versatility. In order to reflect this idea we will characterize alternative structures in terms of complexity levels.

Figure 2 shows several levels of complexity which can exist between models and the data base. The first level is the case when the desired data exists in the data base and the data will only be accessed for listing. A system which can perform this type of information flow possesses the lowest level of complexity and in this case is merely a storage and retrieval system.

The second level of complexity exists when data is collected and prepared for entry into one application program. With this type of system, the decision-making process must be repeated for each model and for each problem being analyzed. The result is merely a report which can be printed each time the model is run. This type of complexity requires no model interaction.

For the third level of complexity, several models are run from a common data base. In this type of system the models must be able to interact with each other as well as interact with the data base. This system possesses the most complexity of the three and provides the ability to analyze the most complex problems with the least effort of the user.

Figure 2 Complexity Levels

Complexity Level 1

Complexity Level 2

Complexity Level 3

V. Information Flow of the Indiana Stream Pollution Control Board

As in any organization, we can view the ISPCB in terms of its information flow. The proper handling and control of this flow is extremely vital to this organization since one of its primary functions is to process collected data. Therefore, much concern is given to define the information flow in order to achieve the most efficient and useful results of accumulated data. The ISPCB is divided into five sections whose responsibility is the collection of data:

- 1) Industrial Waste Disposal
- 2) Municipal Sewage Disposal
- 3) Surveys
- 4) Agricultural Waste Disposal
- 5) Semi-Public Facilities

All industries that put waste materials into Indiana waters and all industries that have some type of sewage treatment plant must report to the Industrial Waste Disposal Section. The reports are submitted monthly, but the data reported is collected either daily, biweekly, weekly or monthly; this is dependent upon the size of the treatment plant and the quantities of effluent.

Initially, each affected industry must complete an application form for a permit by the Army Corps of Engineers. This completed form is made available to the ISPCB. The information on the form includes the industry's name, latitude and longitude, major and minor basin names, and other identifying numbers; the collection of application data for all industries makes up a masterfile. After the permit has been issued, monthly reports will be submitted stating the industry's name, city, date, receiving stream (or treatment plant), flow, pH,

suspended solids, temperature, BOD, DO, etc. The specific variables reported by each industry will vary according to the contents of their effluent. The prime use of Industrial Waste Disposal data is for the checking of levels of pollutants.

The Municipal Sewage Disposal Section is responsible for monitoring data from the nearly 300 sewage treatment plants in the state. The monthly reports include seven variables for the inflow into the disposal plant, fourteen outflow variables, and thirteen heavy metal variables (inflow and outflow). Although the monthly report contains daily data, only the maximum, minimum, and mean values of all variables are retained.

A masterfile is created for Municipal Sewage Disposal. Information for each plant consists of the station number, city, population, year building completed, designed flow capacity, types of treatment, and receiving stream. The Municipal Sewage Disposal Section uses its data primarily to check for proper treatment of waste.

The Survey Section is responsible for collecting data pertaining to the streams of the state. Two types of surveys are made: water quality and intensive surveys.

Water quality surveys are made biweekly at 84 specific locations. The testing locations are spread out over the state, and the tests do not represent a thorough investigation of the rivers. An intensive survey, however, investigates a particular area very thoroughly, but there is no regular schedule for the survey. An intensive survey is conducted only when the need arises, and it might not be repeated in the same location for several years. There are approximately fifty intensive surveys made each year with perhaps fifteen data records for each one. The water quality surveys are made to keep a check on the

condition of the selected waterways. Data from an intensive survey is used to find the cause of severe pollution in specific areas.

Farms with a given number of livestock (as specified by law) are required to apply for permits to control animal waste. The information on the application is the major source of data for the Agricultural Waste Disposal Section. Name, city, county, address, types of sewage treatment, types and numbers of animals, adequacy of the facilities, permit status, and date of permit issuance are among the data comprising this masterfile. The data is mainly used to enforcement of permit issuance.

The Semi-Public Facilities Section collects the monthly data concerning the small waste treatment methods as provided by schools, mobile home parks, shopping centers, etc. This section, like the other, has a masterfile consisting of data from permit applications.

Another section, The Technical Information Section, is primarily concerned with analysis of the data collected by the other sections. Its goals revolve around another, the state's primary role in developing plans for the 1972 Water Quality Act. The basic goal of the information system should be to enable this section to efficiently carry out its stated objectives. In a general sense we may establish the information needs--and more specifically what data must be stored, which application programs would be interfaced, and what type of query language capability should be provided--from the basic goals. We shall argue that no systematic study of goals and resulting information needs to support these goals has been made. Rather, the system currently being used evolved mainly from hardware considerations

and software feasibility requiring the user organization (the state planning agencies) to conform.

In order to limit the scope of analysis for the purpose of this paper, the information flow model will be applied to one aspect of the state water pollution board, that aspect being the waste load allocation problem. This area of study is of special interest since recent legislation has required the ISPCB to assign an allocation of waste loads from a cost effective point of view and assigned in a manner which will guarantee that the minimum water quality standards will be maintained.

The first step in designing a system which will meet the above objectives involves defining the specific goals and the information required to meet those specific goals. The specific goal is to determine a load allocation for each polluter for each of the following quantities:

- 1) Carbonaceous BOD (lbs/day)
- 2) Nitrogenous BOD (lbs/day)
- 3) Temperature ($^{\circ}\text{C}$) of Effluent
- 4) Conservative minerals (lbs/day)

These pollutants must be limited such that the following goals are met for each reach of a river:

- 1) Carbonaceous BOD < Goal
- 2) Nitrogenous BOD < Goal
- 3) Temperature of river < Goal
- 4) Conservative mineral concentration < Goal
- 5) Dissolved Oxygen > Goal

With these information requirements defined, the next step is to determine what kinds of models will be required to generate this

information, and what input data will be required to run those models.

The first requirement will be a model which can simulate a river basin and determine the water quality for a given set of treatment levels. A number of models are available which perform this task, such as MULQAL, QUAL I, QUAL II, DOCAL, and DOSAG. Although each of these models has a different data requirement, they all generally require the same core of basic data. The MULQAL model will be selected for the analysis since it allows for each pollution level to be analyzed individually. The types of data required by the model can be grouped into the following categories:

- A) Description of basin structure
(junctions, by-pass-piping, etc.)
- B) River reach statistics (length,
reaeration parameters, velocity
parameters, benthic demand)
- C) Polluter statistics (BOD concentration,
temperature, flow rate, conservative
minerals)
- D) Incremental flows statistics
- E) Headwater statistics
- F) Treatment levels of polluters

Figure 3 shows the information flow for this model.

The second model which is required is an optimization package for determining a good (optimal) solution of assigning waste load allocations to individual polluters. One solution is for a planner to make many runs of the simulation program trying different combinations of treatment levels until a good solution is found. A much more economical method involves determining a criteria for the 'goodness', and then try to maximize (or minimize) that criteria given the constraints mentioned previously. This particular problem requires a nonlinear

Figure 3 MULQAL Information Flow

optimization routine which, when interfaced with the river quality model through the data base, can select a choice of treatment levels which minimizes the criterion function of cost of treatment. The data required for this model is 1) the output generated by MULQAL, 2) cost information about the various pollution removal rates, and 3) derivatives of the equations used in MULQAL. Figure 4 shows the information flow of NONLIN, a routine having the above capabilities.

The third model generates the desired derivatives of the MULQAL equations. The data required for this model is the same as the data required for the MULQAL model, and the output of this model is the derivatives evaluated at a particular point (treatment level).

The fourth model is required to generate the cost functions and their derivatives. The cost information is usually available as a set of cost curves or as a few discrete points on a curve, in which case the user must fit a known function to this curve and then evaluate the parameters of that function. This model might consist of a regression package if it were felt the data were linear. Other models might be used to fit log-linear curves or exponential curves to the data. The input to the fourth model would be the collected cost data, and the output would be the coefficients for cost functions and their derivatives. Figure 5 shows the information flow for this model.

These four models will perform the information generation phase of a specified problem. Before looking at the total data flow one additional data processor should be mentioned which deals with data validation. This routine is required to analyze the data which has been collected and determine which data values might be in error and which

Figure 4 NONLIN Information Flow

Figure 5 Cost Model -Information Flow

data values are missing. Techniques for this include the use of discriminant analysis [4] and time series analysis. Another technique which is not as satisfactory involves determining permissible bounds on each data type.

Figure 6 shows how the total information flow would look for this particular problem. By performing an analysis of flow in this manner, starting with the desired output and working back through the models required to generate that output, the required inputs can then be determined. If this type of analysis is then performed on each of the remaining subsystems of the state pollution board, the total data requirements can then be determined. Using this technique a much more useful and economical data base can be constructed than if one include in the data base every possible variable that could exist or could be measured.

VI. Complexity of Data Management Systems

The simplest operation that would be expected of a data management system is merely the fetching and displaying of data from the data base. Within this level of complexity there exists a further subset of complexities. This involves the manner by which the user designates the data items to be displayed. The range goes from an encoded format of data names to a natural-language request, and from a mere specification of data names to a full expression of qualifications for selecting the item values to display. For example, the simplest may appear as:

LIST/BOD, DO/WABASH/

Figure 6 Modeling Information Flow

The other end of the range is exemplified with
LIST THE BOD AND DO FOR REACH 1 AND REACH TWO OF THE WABASH RIVER
ON 12/31/72.

The Federal Environmental Protection Agency has made available the use of a centralized data storage and retrieval system, STORET. The STORET system, currently being used by the ISPCB, represents an inquiry system of the first level of complexity in terms of the ease of use of the query language. The system operates out of Maryland and is accessible by all state Environmental Protection Agencies through the use of remote terminals. Acting as a computer utility, the reason for establishing the system is twofold. First, the remote terminals provide computer capabilities to those agencies which do not currently have or cannot afford their own computer facilities. Second, the system provides a common format for storing data and a standard method of retrieving this data so as to enable the interstate transfer of the data.

The data collected by the five sections of the ISPCB is stored in the system. A request for data is made from a remote terminal, and, normally, the response (the data listing) is made at that same terminal. As the system is currently operating, there is a 24-hour delay between the request and the response. This is due to the large volume of requests made from the numerous state agencies. The agency may also have the response printed or punched in Maryland and mailed to the agency, especially when the listings are quite lengthy. Calcomp plots are made available by certain STORET routines.

The STORET system is divided into two subsystems--Water Quality Data File and the General Point Source File. The Water Quality File

represents the historical data collected by the several sections of the ISPCB, and acts as a simplified report generator. A request is made of the Water Quality File by specifying 1) the geographical area covered by the request, 2) the data items (parameters of variables) desired to be listed, and 3) certain formatting information for the response listing.

The geographical area may be specified by giving the state, the county, or latitudinal and longitudinal coordinates. To use the third method (which is the method most often used), four pairs of latitudinal and longitudinal coordinates will specify a quadrangle, and all desired items falling within this outlined area will be returned.

There are a few thousand data items which may be specified in the request. The Indiana agency, however, has reduced this to 35 variables which are most frequently used. The reason for the large number of variables available is that, for each type of variable, there will be several corresponding to how that variable was tested and in what units it was reported. (For example, temperature is broken down into fahrenheit, centigrade, summer, winter, air, surface,...) The parameter may be stated to the system by specifying either a five-digit code or an alphabetic synonym.

The formatting capability allows for a page headings, spacing of columns, and grouping of lines.

In addition to specifying which data is desired (through the use of geographical area specification, variable selection, and formatting requirements), the user may instruct the system to calculate types of statistics on the data. For example, the mean and standard deviation

of the variables can be requested. Another routine will produce a 2-dimensional plot of the specified variables against time. Other routines give variance, standard error, maximum and minimum. There are twelve data-processing routines available.

A sample query in the STORET system is given below. This query requests the mean, minimum, and maximum values of several parameters for the specified stations during a given period. "A" specifies the agency code, "S", the station numbers, and "P", the desired parameters (10: temperature; 60: flow; 300: DO; 400: pH). "BD 58" and "ED 59" specify the years 1958 and 1959. "SORT AU, DG S" are parameters specifying how the output is to be sorted and grouped.

```
PGM MEAN,  
A 1110NET,S 230040,S 230041,S 230075,  
P 10,P 60,P 300,P 400,  
BD 58,ED 59, SORT AU,DG S,
```

There is a limited amount of editing as the data is first entered into the system. This primarily includes verification of keypunching and plausibility of values (for example, certain data values must be between stated amounts). However, this data validation is inadequate as it currently stands and must be checked by other means.

The General Point Source File (GPSF) subsystem of STORET consists of a collection of data elements and values arranged in groups and records to describe potential or actual point sources of pollution. It is currently being used to store and access data pertaining to permits. Compared to the Water Quality subsystem, GPSF offers extended formatting capabilities and a more precise method of selecting data.

The most notable extension that GPSF has over the Water Quality subsystem is that qualifiers may be put on the parameters or data items to be listed. For example, the user could request to see the DO and BOD for all rivers with flow between 10 and 20 MGD. Qualifiers may be put on several parameters, and logical AND among the conditions is performed in selecting the records with which to respond. The conditions AND, OR, NOT, equal to, less than, and greater than are available.

A sample GPSF query is given below showing only the clause which specifies which parameters are to be selected. It can be interpreted as "Select values such that parameter 90122XP equals "RIV," "LKE," or "OTH," such that parameter 90145XP is less than 7, and such that parameter 90128XP is between 6 and 10, inclusively.

```
$$SLCT GPSF      ,90122XP=RIV/LKE/OTH,90145XP<07,  
                  90128XPR6/10
```

The extended formatting capabilities allow for the grouping of lines by types of data, and the printing of column totals.

The STORET system is just a data storage and retrieval system, through. No routines are provided for modeling. Also, no facilities are provided for interfacing the data to models if a state agency has access to such a model. If a model is desired to be run, the state agency would have to make a request for data and, after the response has typed out on the terminal, repunch this data in the format needed by the model.

VII. GPLAN

A higher level of complexity exists in a data management system which will perform the interfacing of the data values contained within the data base to certain application programs. The extreme of this level of complexity would be for the system to actually select the appropriate application program to execute. In the area of water quality control, the user is most interested in static modeling of a stream. It would be desirable for the system to select the proper model and the necessary data in order to reply to a user's query.

A proposed solution to this problem of software-data interface is currently being developed at Purdue University. The primary components for the GPLAN (Generalized Planning System) [5] project are:

- a) Data Management System
- b) Query Analyzer
- c) Data Extractor
- d) Control Program
- e) Application Programs.

Figure 7 shows how these various components are interrelated. The following is a brief description of each of the components of GPLAN.

GPLAN/DMS (GPLAN/Data Management System) [2] was developed as a FORTRAN implementation of the CODASYL DATA BASE TASK GROUP (DBTG) Report of 1971 [1]. The primary reason for selecting FORTRAN over COBOL as the host language was that most of the application packages used in the planning area were scientific in nature and written in FORTRAN. The four basic components of GPLAN/DMS are:

Figure 7 GPLAN Framework

- a) Data Description Language
- b) Schema
- c) Data Manipulation Language
- d) Data Base.

Figure 8 shows how these various components interact with each other. The following is a brief description of each of these components.

The Data Description Language (DDL) is used to describe the logical structure into which the data will be stored within the data base. The first section of the DDL describes each record type, which consists of a group of item types. The second section describes how each of the record types are linked together to form various set structures. Each set can have one or more record types as its member records. The data structure defined by the DDL should closely resemble the actual data structure in the real world, and not the specific data structure of a given application program. Figure 9 shows the structure used for the pollution data.

The DDL analyzer produces two tables known as the schema. These two tables describe the record and set structure within the data base. The information in these tables is generated only once by the DDL analyzer, and does not change during use of the data base (except currency pointers). These tables perform the mapping from the DMS routines to the data base, and are stored in core during execution. The first of these tables is the record table which provides all the information for each record type (item names, item types, etc.). This information is similar to the record description section of a COBOL program. The other table is the set table which describes the possible set structures that can exist within the data base. Within this table

Figure 8 Structure of Data Management System

Figure 9 Pollution Data Base Structure

are pointers which indicate which records are the current owners and current members of each set type.

The Data Manipulation Language (DML) provides the link between the users program and the data base. The DML consists of about 60 FORTRAN subroutines which perform various functions. These commands can be grouped into the following general categories:

- a) Open and Close Data Base
- b) Set Current Indicators (owner and member)
- c) Create and Delete Records
- d) Add and Remove Records from Sets
- e) Fetch and Store Data from Current Records
- f) Search through Sets for Records
- g) Provide Statistics.

These commands use the schema as their map into the data base. The commands currently implemented make up a subset of the commands proposed in the DBTG report.

The data base is where all the record occurrences are located along with their various set pointers. The data base consists of a number of fixed length pages which are stored on a direct access file. The DML can store one page of data in core at any time. A page of data consists of a number of record occurrences. Each record might be of a different length. Each record within the data base is assigned a unique data base key. This key tells what page the record is stored on and where in the page the actual record occurs.

The designed goal of the Query Analyzer (QA) is the capability of processing the non-procedural description of problems specified in the users terminology. The current version works only on a limited set of

queries, but is designed for expansion. The Q/A is a table driven program, which builds its tables from the data base. Any portion of the Q/A which is dependent of a specific data application, is stored with the data base associated with that application. By using this technique the same Q/A can have one set of queries for water pollution control, and another set for air pollution. The Q/A is designed to be triggered from a request for information, rather than a request for specific operations. The control program determines which operations (or application programs) will be required to satisfy that request.

The data extractor searches through the data base to locate the data required for running the specified application program. The data extractor then prepares this data in the form for input into the application program. The data extractor can detect missing data, check type information (real, integer), as well as verify basic statistics of data (eg. input must be continuous or must be 0 or 1).

The current implementation has an extraction program for each application. A generalized data extractor is currently being developed which will automatically generate a schema for the application program and then map this into the schema for the data base. This process could also be reversed so that output from an application program could be stored in the data base.

The control program determines which application programs are required to answer a specific query, directs their data extraction, and prepares Job Control Language for executing those programs. This program also performs the appropriate scheduling if more than one program execution is needed for answering the desired query. The goal

of the control program is to provide the desired result while trying to minimize the resources required to generate that result.

Two sample queries that could be made through GPLAN are given below. The first is an example of a data retrieval type of query, and the second requires that models be run to derive the proper response.

```
LIST REACH,DO,AND BOD SUCH THAT  
DO < DO-GOAL OR ( DO < 5 AND  
BOD ≥ 7 ) $
```

```
LIST REACH AND DO SUCH THAT DO < DO-GOAL  
WHEN BOD IS INCREASED BY .10 $
```

The basic version of GPLAN is currently being tested in the environmental planning area of water pollution control [3]. The application programs being used to perform this test include several river simulation models, a non-linear optimization package, and various report generators [6], [7]. The river simulation programs predict the quality of the river based on the river characteristics, and selected values of control variables such as sewage treatment levels. The non-linear optimization package selects a least cost solution for raising the quality of a river to meet specified standards. The river basin currently being studied is the Grand Calumet River and Indiana Harbor Ship Canal. The results of this study have not only provided useful information to the Indiana Stream Pollution Control Board, but have also provided the GPLAN project with much information useful for future development.

VIII. Summary

The purpose of this chapter has been to present some of the problems and approaches to developing an information system that will aid planners in coping with the complex problems of water quality management. The basis of our approach is the systematic study of the information flows and their relationships to the goals of the water planning group. Basing the design of the system on such a user oriented approach, it is our opinion that a more useful system will be developed.

Furthermore, our analysis of the information needs suggest that a useful system would have a fairly high complexity level. This suggests that consideration be given to the extension of the STORET system to handle a model - data base interface as described in this paper. Higher complexity will also exist in systems having a query format approaching that of a natural language.

We would also like to draw attention to the current design of STORET which involves one central computer complex serving all state pollution agencies. The original centralized design was motivated by two concerns; the importance of having a common data format that would allow for convenient data exchange among adjoining states, and secondly, the provision of computer capability to state pollution agencies. At the time of the original design of STORET, relatively few state pollution agencies had computer capabilities. This situation has now changed and probably is no longer a major consideration in design evaluation. Data portability between states is obviously critical but can be achieved without the centralization of equipment. There is a

common core of data items that is relevant for each state, such as BOD and DO measurements, but many other items are specific to only a few states. A common data base that is accessed by all states, such as STORET, must provide storage and retrieval capability for all data items that could be of relevance to any state. On the other hand, a data base system for a particular state would, while using a common format with other states, be a much smaller and more manageable system.

Our aim in this chapter has been to raise some issues in the area of water quality management and information systems, and outline some useful approaches to the problem. Implementation of the water quality information system described herein, resulted in a broadened perspective of those attributes which a decision support system should exhibit in order to facilitate the planner's task. The second chapter examines the content and effects of these insights as they relate to the water quality problem.

Appendix I. References

1. CODASYL Committee, Data Base Task Group Report, Association for Computing Machinery, April, 1971.
2. W. D. Haseman, A.Z. Lieberman, J.F. Nunamaker, A.B. Whinston, Generalized Planning System/Data Management System (GPLAN/DMS), Users Manual, Krannert School of Industrial Administration, December, 1973.
3. W.D. Haseman, J.F. Nunamaker, Jr., and A.B. Whinston, Data Base Management System For Pollution Control, Proceedings of the Seventh Annual Symposium on the Computer Science and Statistics Interface, October, 1973.
4. James Marsden, D.E. Pingry, and A.B. Whinston, Large Scale Data Analysis - The Identification of Outliers, Krannert School of Industrial Administration, June, 1974.
5. J.F. Nunamaker, Jr., Donald Swenson, and A.B. Whinston, Specifications for the Development of a Generalized Data Base Planning System, Proceedings of the National Computer Conference, New York City, AFIPS Press, June 1973.
6. D.E. Pingry, and A.B. Whinston, A Multi-Goal Water Planning Model, Journal of Environmental Engineering Division, ASCE, December, 1973.
7. D.E. Pingry, W.D. Haseman, and A.B. Whinston, MULQAL - Multiple Quality River Simulator, Program Documentation and Users Manual, Krannert School of Industrial Administration, November, 1973.

Chapter 2

I. Introduction

In view of the magnitude and complexity of contemporary-societal problems and the importance of finding solutions to these problems, it is imperative that decision-makers have in their hands those tools which can facilitate decision-making processes. Typically, a decision is the result of the decider's analyses of available, pertinent data. In this presentation, we focus attention on a planner's utilization of formal analyses (models) which make use of large volumes of logically inter-related data. The needs to collect, store, and access large quantities of data and to subject that data to modeling procedures suggest that computerized information systems may be potentially valuable tools for the planner. The manner in which a computerized information system handles the functional interfaces between the planner, the models and the data are of particular significance. Inherent in the very nature of the planning activity is the dynamic quality of the interface between model and data; for even though the set of data may be somewhat stable over time, both the problems and the models used in solving those problems are continually changing. It should be noted that each model operates on a distinct subset of the total set of data. And it is submitted that the task of managing this dynamic interface should be handled by a computerized data management system.

Another interface that deserves attention is that between the planner and the data management system that supports the models and data. The nature of this interface may range from interaction through procedural programming languages to control of the data management system with a non-procedural English-like query language, according to the needs of the planner.

II. Structural Complexity

Alternative methods of organizing an information system with respect to the models and data have varying implications for the interfaces under consideration here. These implications relate to the factors of generality, versatility, speed, and ease of use. We characterize the methods of system organization in terms of structural complexity. Several levels of complexity that can exist between models and data are depicted in Figure 1. The first level is the lowest level of complexity and refers simply to the retrieval of stored data. The second level of complexity exists when data are collected and prepared for entry into a single application program. In a typical planning situation, the planner has a group of models available along with a voluminous set of data. Operating at complexity level two, the following tasks are required for each model that is pertinent to the planner's current problem:

- a) Identification of an appropriate model.
- b) Determination of the data requirements of that model, given the context of the current problem.
- c) Location and verification of the existence of this germane data.

Complexity Level 1

Complexity Level 2

Complexity Level 3

Figure 1. Levels of Complexity

- d) Reorganization of the data into a configuration that is acceptable as input to the model.
- e) Determination of job control language commands.
- f) Execution of the model.

The result of this procedure is a report that can be printed each time the model is executed. This level of complexity permits no interaction among models, requires direct user involvement in managing the total data set and in setting up proper data configurations, and is characterized by high inflexibility of the interface between data and model. Though such a scheme may be workable in a non-planning environment where the data-model interface is repetitive and constant over time, it limits the effectiveness of the planner whose time and talents are presumably too valuable to be employed in the tedious task of data management.

The third level of complexity is the case in which all models are executed from a common data base. There is also the capacity for models to interact with each other. In this type of system, the function of data management is automated so that most, or even all, of the six tasks required at complexity level two are handled automatically. The extent of this automation depends upon the nature of the interface between the planner and the data management system. If flexibility is permitted with regard to this interface, it is possible for users who are programmers to interact with the system via a programming language, whereas non-programmers may submit a single non-procedural, English-like command which results in tasks c) through f) being performed. Depending upon the sophistication of the query language, tasks a) and b) may also be carried out automatically. A data base management and query system

which supports the third level of complexity is the Generalized Planning System (GPLAN) developed at Purdue University. In order to illustrate the value of GPLAN as a planning tool, we consider the problem of water quality management.

III. Water Quality Management

The 1972 Water Quality Act greatly increased the powers and responsibilities of governmental agencies to maintain and restore water quality as one part of environmental enhancement. The specific planning problem considered here concerns the assignment of waste loads to polluters in a river basin. This assignment of waste load allocations must be made from a cost effective viewpoint and in such a manner that minimum water quality goals are met.

The water quality problem is examined on a basin-wide basis, since waste discharge from an outfall may affect not only local water quality, but downstream qualities as well. Also, the result of waste discharge into a given river can vary significantly according to the river characteristics at (and downstream from) the actual outfall point. For instance, if the outfall is located at a point where the river has a large assimilative capacity, the value of further waste treatment facilities could be negligible; however, if the discharge occurs in an area with little assimilative capacity, then the introduction of additional treatment facilities can make a definite contribution towards the improvement of water quality. In view of these considerations, both the imposition of uniform discharge restrictions applicable to all outfalls and the ad hoc decisions as to treatment plant construction

at pollution sources can be seriously questioned with regard to efficiency and long-term effectiveness. In addition, basin-wide analysis may indicate that economies can be achieved through methods such as treating wastes from several sources at a regional facility, bypass piping, instream aeration, or flow augmentation.

Thus, given some set of water quality goals, there are a number of alternative basin-wide strategies which satisfy the goals; but one strategy may have significantly different implications for a particular polluter than another strategy. A decision rule for the selection of a particular feasible basin-wide strategy can be to choose that strategy which requires the smallest basin-wide cost. Of course, there may be certain non-quantitative criteria which also influence the planners' final decision. Though these criteria are not discussed here, the technique which is examined provides participants in the planning process with information about the costs and water quality effects of alternative treatment strategies. In summary, this technique has two aspects:

1. Simulation. This involves the capacity to determine the water quality characteristics implied by a given treatment strategy.
2. Optimization. Minimize the basin-wide treatment costs, subject to satisfaction of water quality goals.

The first stage in implementing the water quality information system is the specification of objectives and of information needed for fulfilling those objectives. The ultimate aim is to assign to each discharger a maximum daily discharge amount, in pounds, of each of the following quantities: 1) Carbonaceous biochemical oxygen demand (BOD);

2) nitrogeaneous BOD; and 3) conservative minerals. Also to be assigned is a maximum temperature for discharger's effluent. These assignments must be made, for each river reach, such that carbonaceous BOD, nitrogeaneous BOD, heat concentration, and conservative mineral concentrations do not exceed their respective water quality goals and such that the measure of dissolved oxygen in each reach is at least as great as some set standard. The term 'reach' refers to a section of the river throughout which certain state characteristics, such as reaeration parameters, velocity parameters, and deoxidation rates, are relatively stable.

There exist several simulation models which predict the water quality for a given basin-wide treatment strategy; these include DOSAG, DOSCAL, QUAL I, QUAL II, and MULQAL. The basic core of data needed by each of these is identical, though there is variation in required data configurations. MULQAL [1], [2] is examined here, since it is the most comprehensive of the models, in the sense that it takes into account the effects of heat concentration on both the dissolved oxygen level and the rates of oxidation and reaeration. This model uses data of the following types: 1) Basin structure description, including by-pass-piping plans and junction configurations and locations; 2) river reach information; 3) specifications of all discharge activities, such as flow rate, temperature, and concentrations of BOD, ammonia, and conservative minerals; 4) descriptions of the water quality of all incremental flows; 5) characterizations of headwaters in terms of BOD, dissolved oxygen, temperature, and conservative minerals; and 6) piping configurations and discharge treatment levels. Figure 2 depicts the

Figure 2. MULQAL Information Flow

MULQAL information flow. The result of the MULQAL simulation is a description of the water quality of each reach in terms of BOD, temperature, conservative minerals, etc. along with the load allocations derived from polluter and treatment statistics.

* The planner could use MULQAL iteratively, varying the treatment levels with each iteration, until a treatment strategy is found in which water quality goals are met on every reach. Alternatively, an optimization model may be interfaced with the simulation model and data base in order to determine the least cost basin-wide treatment strategy. The nature of this planning problem demands the use of a nonlinear optimization algorithm (NONLIN) which has the following data requirements: 1) Cost data for pollution removal rates; 2) simulation results of MULQAL; and 3) derivatives of equations used in MULQAL. The NONLIN information flow is given in Figure 3. The cost data is input to a routine that generates both the cost functions and their derivatives that are used by NONLIN. Another routine, using the same inputs as MULQAL, evaluates the derivatives of the MULQAL equations at a particular treatment level. Figure 4 presents an overview of the information flow which commences with the validation of data and culminates with the load allocation that is implied by the optimal treatment level strategy indicated by NONLIN.

IV. An Illustration of Complexity of Data Management Systems

We now consider the attributes of each of the levels of data management complexity in relation to water quality management. It should be noted that the functions of each level of complexity can be

Figure 3. NONLIN Information Flow

Figure 4. Overview of Modeling Information Flow

handled by all higher complexity levels. The most elementary level consists of retrieval of stored data. Since no modeling is involved here, there is no model-data interface; but there are wide variations possible in the interface between the user and the data. First of all, there are variations in the methods by which the user specifies the data to be retrieved. At one extreme, there is use of a procedural language such as COBOL or FORTRAN, while at the opposite extreme, data may be retrieved by issuing a single non-procedural command such as LIST BOD FOR RV = WABASH. A second range of variations on the user-data interface is concerned with the selectivity of retrieval. This spectrum ranges from unconditional retrieval to highly selective retrieval. For instance, LIST ID, BOD, TEMP FOR RV = WABASH lists the discharger ID, BOD, and temperature for all dischargers on the Wabash; there is considerably more flexibility when compound conditionals operating on any mix of data items is permitted, eg. LIST THE BOD AND TEMP FOR DISCHARGER 38 ON THE WABASH ON 11-1-75 or LIST DISCHARGER FOR TEMP < 21 AND BOD < 5.67.

The result of adding a modeling capability to the function of retrieval is a system of complexity level two. When the user desires to use MULQAL to make a simulation run for the Wabash river, he must locate all data that are required and then reorganize the data into the format needed by MULQAL. If the user needs a simulation run for the Wabash using polluter, incremental flow, and headwater data that have been collected in a different time period or if he needs a simulation for the White River, he must once again go through the process of data manipulation. The same situation holds for the use

of NONLIN, of statistical packages, regression analyses, plotting routines, etc. In view of the cumbersome nature of this process, it is proposed that a system which automatically sets up the proper model-data interface in response to a simple command (eg., RUN MULQAL), which allows the user to access the data base with a high degree of selectivity through either an English-like query language or a procedural data manipulation language, and which allows interaction among models would be beneficial to users whose information needs are constantly changing. The Generalized Planning System meets all of these criteria.

V. GPLAN Description

GPLAN [3] 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) [4], while the latter two are fundamental components of the query system (GPLAN/QS) [5].

GPLAN/DMS was designed around the specifications of the CODASYL DBTG report of 1971 [6] 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 5 presents a pictorial representation of the water quality data base schema. Though the schema shown here defines a hierarchical set structure by means of which the data is organized, GPLAN/DMS and GPLAN/QS are also capable of supporting the more complex network set structures. The schema may be thought of as being a template which is utilized in building the data base and navigating through it.

In Figure 5, each box represents a record-type. Each record-type is composed of one or more data item types. For instance, the record-type BASN is composed of the data item types BSID, BSNA and E. Each arrow 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, BASN is the owner of the set RVIB and RIVR is the member. For each record-type RIVR defined in the schema, there may be a data base record

Figure 5. Data Base Structure

occurrence for all rivers in a state. A set denotes a one to many relationship between each occurrence of the owner record type and many occurrences of the member record-type. Each occurrence of river owns many occurrences of reach. 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 6. Each data item has the attributes of type (INTEGER, REAL, CHARACTER, BINARY, COMPLEX), size (number of characters or 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 (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 the 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

	<u>NAME</u>	<u>TYPE</u>	<u>SIZE</u>	<u>MAX</u>	<u>COMMENTS</u>
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

⋮

	<u>NAME</u>	<u>ORDER</u>	<u>KEY</u>	
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
OWNER
MEMBER

⋮

MPIB
BASN
MODL

SORT

MOID

MODEL PARAMETERS BY DATE
BASIN
MODEL PARAMETERS

Figure 6. Sample DDL

of the data bases; 2) creation and deletion of record occurrences; 3) setting of currency indicators; 4) addition and removal of record occurrences from sets; 5) retrieval and storage of data from current record occurrences; and 6) searching through sets for particular record occurrences. 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), histogram, regression, statistics (number, minimum, maximum, average, standard deviation, variance), and add, change and delete routines for altering the data base contents. 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 reformatting 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 items 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 7. 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.

VI. 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 the 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

Figure 7. GPLAN System

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 8. It can be noted that this extraction routine makes extensive use of the Data Manipulation Language. The FMSK command finds the member record occurrence 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 occurrence 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 and deposits this information in the first five words of the FORTRAN variable IDATA. Subsequent uses of GFM fill the array IDATA with appropriate fields from the current

```
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 8. Sample Code for MULQAL Extraction Routine

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 input, 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.

VII. 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 base [7], [8], [9] 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, Whinston [10] 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 5, each record type is owned by the system; the parenthesized identifier at the upper right corner of each record type is the name of the system-owned set of which that record type is a member. Hence in this example, there are eighteen 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 [11], [12]. 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.

VIII. Conclusion

In conclusion, it must be emphasized that the Generalized Planning System is quite capable of supporting data bases and models for a wide variety of planning problems, and not merely in the specialized area of water quality management. Examples range from transportation problems to urban planning to economic planning. But regardless of the problem under consideration, the Generalized Planning System is a valuable planning tool in that it automates the interfacing between data and models, and it offers a non-procedural, English-like interface between the planner and his computerized resources.

- [1] Pingry, David E. and Whinston, Andrew B., "A Multi-Goal Water Planning Model," Journal of Environmental Engineering Division, ASCE, (December, 1973).
- [2] Pingry, David E., Haseman, William D., and Whinston, Andrew B., MULQAL - Multiple Quality River Simulator, Program Documentation and User's Manual, Krannert School of Industrial Administration, (November, 1973).
- [3] 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).
- [4] Haseman, William D., Lieberman, Arthur Z., and Whinston, Andrew B., GENERALIZED PLANNING SYSTEM/DATA MANAGEMENT SYSTEM (GPLAN/DMS), Users' Manual, Krannert Graduate School of Industrial Administration, (September, 1974).
- [5] 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).
- [6] CODASYL Committee, Data Base Task Group Report, Association for Computing Machinery, (April, 1971).
- [7] Codd, E.F., "Normalized Database Structure: A Brief Tutorial," Proceedings 1971 ACM SIGFIDET Workshop on Data Description, Access and Control, ACM.
- [8] Codd, E.F., "Further Normalization of the Database Relational Model," Database Systems, Vol. 6, Prentice Hall (1972).
- [9] Date, C.J., An Introduction to Database Systems, Addison Wiley (1975).
- [10] Bonczek, R.H., Haseman, W.D., and Whinston, A.B., "Structure of a Query Language for a Network Data Base," (July, 1975).
- [11] Chamberlain, D.D. and Boyce, R.F., "SEQUEL - A Structured English Query Language," Proceedings ACM SIGFIDIT Workshop, Ann Arbor, Michigan, (May, 1974).
- [12] Bonczek, R.H., Haseman, W.D., and Whinston, A.B., "Network vs. Relational Data Base Implementation," (August, 1975).