Database System Approach to Management Decision Support

JOHN J. DONOVAN
Massachusetts Institute of Technology

Traditional intuitive methods of decision-making are no longer adequate to deal with the complex problems faced by the modern policymaker. Thus systems must be developed to provide the information and analysis necessary for the decisions which must be made. These systems are called decision support systems. Although database systems provide a key ingredient to decision support systems, the problems now facing the policymaker are different from those problems to which database systems have been applied in the past. The problems are usually not known in advance, they are constantly changing, and answers are needed quickly. Hence additional technologies, methodologies, and approaches must expand the traditional areas of database and operating systems research (as well as other software and hardware research) in order for them to become truly effective in supporting policymakers.

This paper describes recent work in this area and indicates where future work is needed. Specifically the paper discusses: (1) why there exists a vital need for decision support systems; (2) examples from work in the field of energy which make explicit the characteristics which distinguish these decision support systems from traditional operational and managerial systems; (3) how an awareness of decision support systems has evolved, including a brief review of work done by others and a statement of the computational needs of decision support systems which are consistent with contemporary technology; (4) an approach which has been made to meet many of these computational needs through the development and implementation of a computational facility, the Generalized Management Information System (GMIS); and (5) the application of this computational facility to a complex and important energy problem facing New England in a typical study within the New England Energy Management Information System (NEEMIS) Project.

Key Words and Phrases: decision support systems, management applications, database systems, modeling, virtual machines, relational, networking
CR Categories: 2.1, 3.5, 4.33

1. CHARACTERISTICS OF DECISION SUPPORT SYSTEMS

Traditional intuitive methods of decision-making are no longer adequate to deal with the complex problems faced by the modern policymaker in both the public and private sectors. The difficulty in addressing these problems is further complicated by the interrelations, immediacy, and far-reaching implications of actions

This work was supported in part by the New England Regional Commission under Contract 10630776, in part by IBM Corp. through an IBM/M.I.T. Joint Study, in part by the Federal Energy Office under Contract 14-01-001-2040, and in part by M.I.T. Institute funds.
Author's address: Center for Information Systems Research, Alfred P. Sloan School of Management, Massachusetts Institute of Technology, 50 Memorial Drive, Cambridge, MA 02139.

taken. Decision support systems [54] have been developed to provide the information and analysis necessary for the decisions that must be made. While database systems lie at the heart of decision support system tools, the characteristics of the problems associated with decision support are different from those to which database systems and other computational technologies have usually been applied in the past. The modern-day problems which the policymaker must face are such that it is likely that the policymaker's perception of the problem will change over time and that the inherent nature of the problem itself may change. Hence additional technologies, methodologies, and approaches are needed to make existing database and computational systems effective in addressing problems of such a nature.

Consider two examples that characterize some of the information system needs of decision support systems. The first example is a fairly detailed one, the second a more general and broad one.

1.1 Leading Indicator Example

In recognition of the need to forecast demand for energy in the future, the Federal Energy Administration asked the M.I.T. Energy Laboratory and the Center for Information Systems Research of the Sloan School of Management to investigate the development of a set of leading indicators which would indicate energy demand [42]. One such indicator is simply a plot of the average miles per gallon (MPG) of all new cars sold within a month over a period of months. Figure 1 is such a plot for the months January 1974 through January 1975, the months in which the "energy crisis" was perhaps most severe—that is, gasoline lines were the longest.

It was expected that people would buy smaller cars in response to the energy crisis, and therefore the curve should go up. Since the results shown in Figure 1 were contrary to expected behavior, the policymaker's immediate reaction was to try to explain the plot. In doing this, questions were raised, and it became apparent that a slightly different problem needed to be addressed.

Were the results that Figure 1 illustrates obtained because heavier cars had rebates, and thus more large cars were being bought? Or was it the case that during the time under consideration the United States was in a recession and only people in upper income levels were buying automobiles? Hence sales of luxury cars might have remained at a high volume, whereas the sales volumes of all other cars were reduced. Total sales would thus be weighted toward the heavy luxurious cars with lower MPG.

To determine which hypothesis is correct, we must look at the *types* of cars sold and the volumes of those cars. When we plotted such data, it confirmed the second hypothesis that in fact luxury car sales remained relatively constant while the sales of cars that may have appealed more to lower or middle income families dropped off considerably (see Figure 5 for output example).

Note that because of the policymaker's attempt to account for and explain the original plot, questions were asked that changed his perception of the problem. In the mind of the policymaker the problem had changed from one of simply producing an indicator of energy demand to one of explaining why that indicator behaved as it did. The problem we at M.I.T. had to solve was thus changed, and therefore the use of the data required to answer the problem was changed, as were the keys by which the data was accessed.

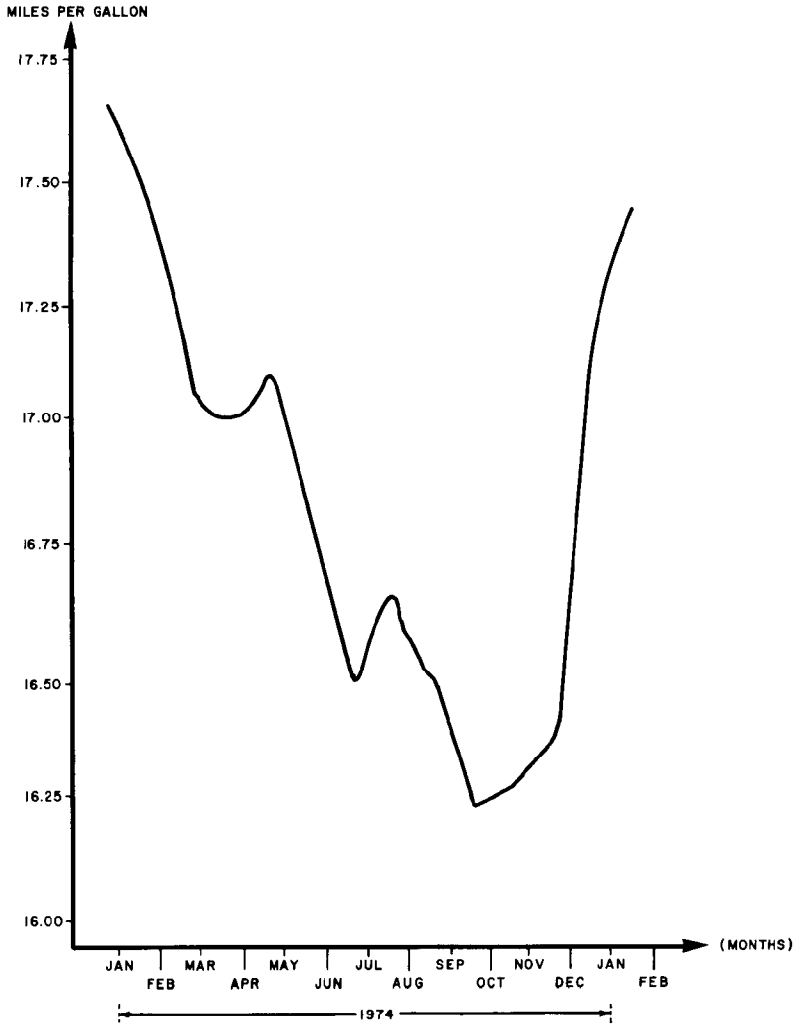

Fig. 1. Average MPG of cars sold in a month

1.2 Regional Energy System Example

Let us cite an even more dramatic example, one in which the actual nature of the problem itself changed. This involves a request that the M.I.T. Energy Laboratory and Center for Information Systems Research received at the height of the energy crisis during the winter of 1973–1974 from the New England Regional Commission. The request was to develop an information system to assist the region in managing the possible distribution of oil to minimize the impact of shortages throughout the region [18]. A considerable amount of effort was spent in designing and developing a prototype of just such a shortage information system. But less than six months later, before the system became fully operational, the problem had changed completely. New England was no longer in a shortage situation, as there was a backlog of full tankers in Boston Harbor. Instead the region was beset by a

new series of problems, namely price increases. Prices of energy had gone up by over 50 percent in that three-month period (winter of 1973–1974). Certain industries and sectors within the region were thus adversely affected. As the region realized its vulnerability to price fluctuations in energy, the problems of the policymaker shifted from ones of handling shortages to ones of analyzing methods to conserve fuel, the impacts of tariffs, decontrol, and natural gas or oil prices on different industrial sectors and states within the region, the merits of refineries, and the impacts of offshore drilling on New England's fishing industries. These are but a few of the problem areas which New England policymakers faced and on which they needed immediate support.

1.3 Summary of Characteristics

Let us summarize the characteristics of the problems associated with the two decision support system examples given above:

(1) The problems are continually changing, either because the policymaker's conception of the problem changes or because the problem itself changes. This certainly was the case when New England shifted from a shortage situation to a price impact situation. This is unlike payroll, marketing, and other types of information systems which have been applied in the past—systems that deal with areas in which the problems can be fairly well defined and do not change as dramatically.

(2) The answers to the problems are needed quickly. The state energy officers, as well as legislators and governor's officers, must be able to respond rapidly to energy problems and to initiate or evaluate regional as well as federal legislation.

(3) The data necessary to perform the analyses are continually changing. They come from many different sources, and a large part was obtained by independent organizations.

(4) Because of the complex nature of the problem, more than raw data is needed. Sophisticated analyses, transformations, displays, projections, etc., are required.

(5) Because the problems are constantly changing, the mechanisms for solving those problems are less concerned with long term efficiency and more concerned with rapid implementation and robustness.

Point (5) is expanded upon in Figure 2, which depicts the characterization of the cost of an information system. The solid line depicts typical expenditures in inventory control or payroll systems, where the fixed costs (i.e. the cost of developing the system) are not as important as the variable costs (i.e. the cost of operating such a system). Hence in these traditional systems much more emphasis is placed on the tuning of the system for performance. However, decision support systems (DSS) have different requirements. The literature describes [13] two types of DSS: (1) *institutional*, which support decisions of a recurring nature (e.g. financial planning system); (2) *ad hoc*, which support one-time decisions (e.g. to merge a corporation). For the institutional DSS, the computational system often is revised many times during its development, hence the need for "breadboarding" facilities (i.e. the focus is not initially on tuning such a system but in adapting it to fit user needs). For the ad hoc system, the emphasis is to get the system operational and running as soon as possible and not on fine tuning. Hence, either during the breadboarding process of an institutional DSS or during the entire life of an ad hoc DSS the systems are seldom operating in a stable mode long enough to make

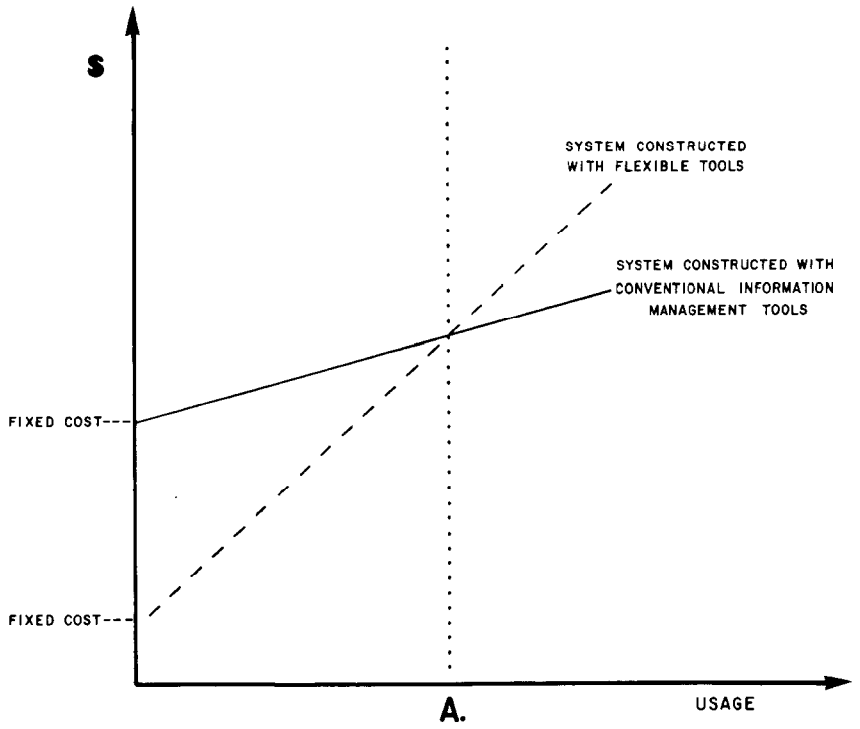

Fig. 2. Fixed costs and variable costs versus usage

the operational costs paramount (i.e. a decision support system is typically operating in the region below A in Figure 2). The dotted line in Figure 2 characterizes the types of expenditures that are necessary in decision support systems, namely, low fixed costs. Of course it is desirable to have low variable costs as well, but if there exists a tradeoff, low fixed costs are preferable to low variable costs.

1.4 Other Examples

Others have noted similar characteristics in applications areas, e.g. Siegel [55] in systems for assisting in collective bargaining; Little [36] in systems for marketing; Altshuler [1], Plagman and Altshuler [49], Scott Morton [54], and Rockart and Scott Morton [52] in corporate decision support systems.

2. EVOLUTION AND NEEDS OF DECISION SUPPORT SYSTEMS

Most applications of database systems and computer-based information systems have been aimed at operational control or management control in organizations, and therefore the major concerns of such systems have been at low levels, dealing primarily with raw data. With a decision support system, data analysis needs are more important. Furthermore, mechanisms must be included for quickly adapting to the changing nature of problems, for assimilating new data series, and for integrating existing models and programs in the effort to save time in responding to a particular decision maker's request. Hence computational technology as applied to

decision support systems needs a new approach, not simply a better, faster database management system.

2.1 Types of Information Systems

To draw this distinction between the traditional use of computational systems and uses in the field of decision support systems, we refer to Figure 3. Depicted in this figure is a framework for information systems that was developed by Gorry and Scott Morton [24]. This framework combines characterizations of Anthony [2] and Simon [56]. Anthony's characterization is based on the proposed purpose of management activities (listed across the top of the matrix), while Simon's classification is based on the way that management deals with problems (listed along the side of the matrix).

The unshaded areas of Figure 3 represent the types of information systems in which computers and computer technologies have been most effectively used to assist management in the past. Inventory control packages are widely available, as are accounts receivable packages, budget analysis, scheduling packages, and tools to perform PERT and cost analysis.

However, information systems depicted in the shaded area (called decision support systems) demand more than the traditional database system can offer, and very little attention has been given to the technologies needed by such systems.

2.2 Evolution of Awareness of Decision Support Systems

2.2.1 *Decision-Making Process.* A theory of human decision-making was developed by Newell et al. [45] and applied in the work of Clarkson and Pounds [7]. Pounds [50] focused on the problem of identifying the managerial problems to be solved, and he constructed a theoretical structure which a manager can use for solving problems. However, these works placed little emphasis on the computer capabilities necessary to assist in the decision-making process. Rather, they focused on the more basic problem of understanding the decision-making process.

2.2.2 *Computer's Participation.* Licklider [35], as well as others (e.g. Zannetos [61]), advocated both the need for computer participation in formulative and real-time thinking and the need for cooperation between man and computer in decision-making. He discussed some of the computer technologies which at that time (1960) were felt to be prerequisities for the realization of man-computer symbiosis. That technology has proven to be inadequate.

2.2.3 *Characterization of Problems To Be Addressed in Decision Support Systems.* Scott Morton [54], Rockart and Scott Morton [52], and others (e.g. Davis [10]) articulated the characteristics of the problems which decision support systems address. Scott Morton's points emphasize the "unprogrammed" nature of problems in decision support areas. The characteristics we recognized in Section 1.3 further substantiate these earlier observations of Scott Morton's about decision support area problems, namely, dynamic environment, high requirements for data manipulation, complex interrelationships, and large databases. He discusses the comparative advantage of interactive display system technology for management decision-making.

2.2.4 *Models and Databases.* The role of models in the decision-making process has been discussed by Little [36] as well as by Urban [60]. Plagman and Altshuler

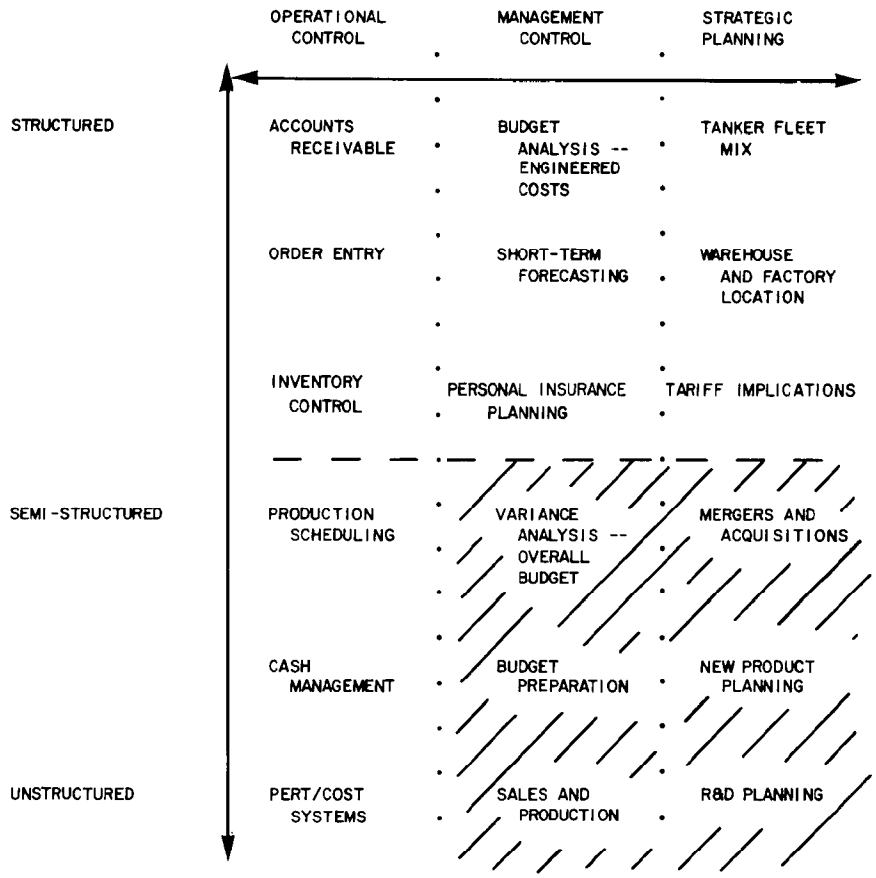

Fig. 3. Framework for management information system (from Gorry and Scott Morton [24])

[49] have articulated the role of databases in decision support. In this paper we build on the realization of the importance of both models and databases for decision support.

Plagman and Altschuler consider some of the questions concerning the structure of the database in decision support systems (which they call Corporate Level Systems). In particular they suggest what data about data should be maintained in these systems in order to help solve such recognized problems of decision support systems as validation and open-ended design.

2.2.5 *Why So Slow?* Even with the recognition of some of the needs of decision support, wide scale use of computers for decision support systems has lagged behind the use of computers in other application areas. Siegel [55] has explained why this is so. Siegel discusses the potential for computers in the decision support area of collective bargaining. With the availability of employee data, trade data, corporate data, and facilities to test and explore possibilities of reconciliation, the computer appears to offer a logical aid. Yet the application of computers in the field of collective bargaining is in its infancy.

Siegel has suggested the following reasons for the slow advancement of computer

technology in such applications. Currently computer technology is best adapted to problems where advanced understanding, advanced structure, and unified data exist. However, the problems addressed in collective bargaining are not of this type. Rather, the problems in this field keep changing (hence there is not enough time to reprogram), they are loosely structured, and despite the ready availability of the data involved, it is in a variety of forms. Others, as noted earlier, have also recognized these problems, but until recently the technologies were not available to address these problems.

This paper proceeds from a recognition of the characteristics of problems to which decision support systems must be applied. We formulate these characteristics in terms of the technologies that can address them and we present an instance in which those technologies were applied in building several decision support systems that are now operational.

2.3 Technical Requirements To Support Decision Support Systems

From our work and from that of others we recognize the following technical requirements that are needed to meet the characteristics (cited in Section 1.3) of the problems that decision support systems must address:

(1) Data management capability. Because the problems are continually changing and the answers are needed quickly, the data management capability must have an interactive component that can quickly introduce new data series and validate, protect, and query them. Furthermore, the data associated with these problems is constantly changing and must be obtained from many different sources. Since many of the data series exist under different and potentially incompatible database management systems, and these data series are maintained by people using those systems, it is important to have facilities for integrating data from diverse general purpose database management systems into such a form that a single-user application program may access them.

(2) Analytical capabilities. Coupled with the capabilities of a database it is important to have sophisticated computational facilities for analyzing the data since the data are continually changing and the complexity of the problems addressed demand more than raw data. Sophisticated analyses, projections, etc., are needed. Required capabilities include modeling languages, statistical packages, and other analytical facilities.

(3) Transferability. Because of the short time available for responding to the policymaker's needs, it is important to be able to build upon existing work, such as existing models or existing programs. It is also necessary to work with data coming from many different sources. Hence it is valuable to bring up in an integrated framework any programs, computational aids, or data series that are applicable, even though these programs and data series may currently be operating on seemingly incompatible computer systems.

(4) Reliability, maintainability, flexibility, and capabilities for incorporating new technologies. Software associated with any decision support system must be reliable and maintainable, and as new technologies develop, those technologies should be able to be incorporated quickly into a flexible decision support system.

The third requirement, transferability, is important for two reasons: First, as was mentioned above, it allows for the incorporation of existing models, data-

base systems, and other software into an integrated framework quickly; second, it minimizes the need to retrain users of existing systems. Economists and support personnel (in a project like an economic impact study) should be able to operate a computational facility using data management tools and languages with which they are familiar. They would not be required to learn new tools at a loss of time and expense.

Examples of existing analytical and data management tools are: econometric languages such as TROLL [43], TSP [27], and EPLAN [31, 53], analytical tools such as PL/1, Fortran, APL [46], and DYNAMO [51]; statistical tools such as MPSX [34] and APL STATPACK II [57]; editors such as VM/370-CMS's EDIT [30]; retrieval systems such as DIALOGUE [59]; and database systems such as IMS [33], SEQUEL [6], and Query by Example [62]. A user should be able to activate any one of these tools even though many of the tools are seemingly incompatible in that they operate under different operating systems.

3. GENERALIZED MANAGEMENT INFORMATION SYSTEM

In response to the technical requirements mentioned in Section 2.3, a prototype facility called the Generalized Management Information System (GMIS) has been implemented [16] with VM/370 [29]. It is not the purpose of this paper to describe GMIS completely, and furthermore, GMIS is only a first attempt. However, let us briefly discuss some of the technologies developed and used in GMIS which have met some of the needs presented above. We hope that this discussion wil stimulate further work on these and other techniques.

The GMIS system is operational and currently is being applied to decision support systems to aid energy impact analysis and policymaking.

3.1 Virtual Machine Approach To Transferability

Primarily in response to the transferability need outlined above, our emphasis has been to find techniques for accommodating different database systems and analysis systems in one integrated framework. Rather than force the conversion and transport of application systems to one operating system, we advocate the use of the virtual machine concept [23, 39, 47] and the networking of virtual machines [5, 16]. A virtual machine (VM) may be explained simply as a technique for simulating one or more real machines on an existing computer. This technique is essentially accomplished by programs that timeshare the resources of the single physical machine among different operating systems [41].

3.2 GMIS Evolution

GMIS has evolved through its actual use in real decision support systems, especially the New England Energy Management Information System (NEEMIS) Project which started at M.I.T. in 1973 [17]. In 1974 additional resources (personnel, programs, computational facilities) became available from IBM (Cambridge Scientific Center and San Jose Research Center) as a result of an IBM/M.I.T. Joint Study Agreement, which has greatly enhanced the development of the present system.

Early configurations of GMIS focused on the data management needs of decision

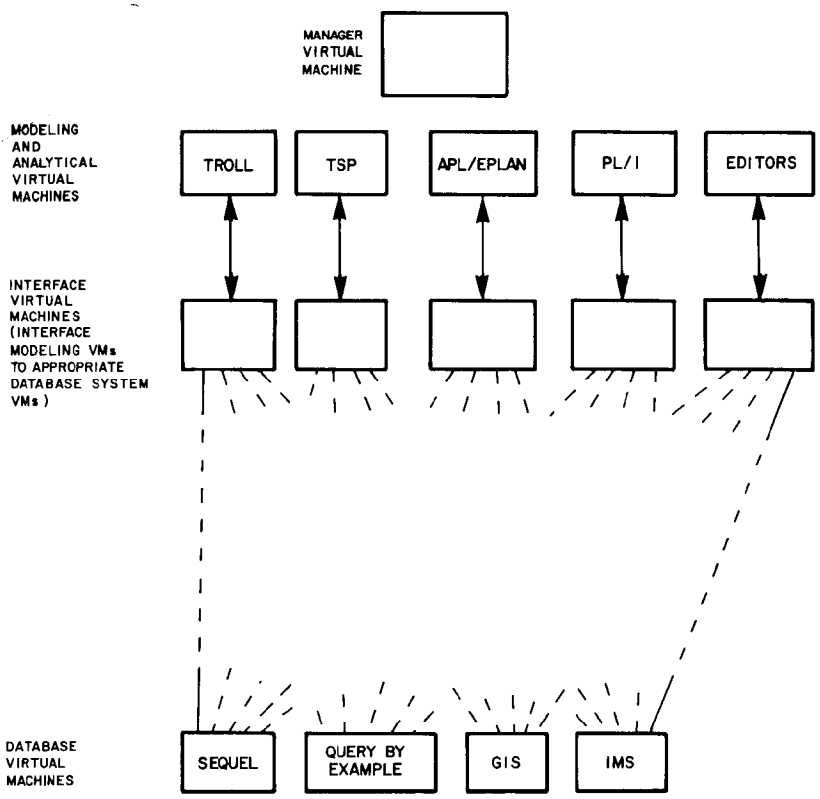

Fig. 4. Present GMIS configuration

support systems since NEEMIS was initially concerned with energy shortages in New England [17], and thus it was necessary to keep track of data on fuel flows in New England. This early system used an M.I.T.-developed prototype relational data management system [58], which was later replaced by the IBM experimental SEQUEL system [6]. SEQUEL is an experimental interactive data management and data definition language based on a relational model [8], which has been made available through an IBM/M.I.T. Joint Study. SEQUEL was modified and enhanced to make the experimental code more effective in an operational environment, and a user-oriented interface was added that permitted communication with most terminal devices and that provided report-writing capabilities [21, 26]. As the NEEMIS problem areas changed and requirements for data analysis increased, an interface between VMs running PL/1 and APL programs and VMs running SEQUEL was developed [26]. Furthermore, it became evident that modeling, transportability, and multiuser access to the same database system were important; hence a configuration of several VMs was developed [22]. In that configuration several different modeling facilities, each running in its own VM, could communicate with an interactive data management facility running on a different VM [16]. In that same paper it was proposed that multiple database facilities be made accessible. That need became more evident, and such a configuration has now been implemented in the present GMIS.

3.3 Present GMIS Structure

The configuration of VMs used in the present GMIS is depicted in Figure 4, where each box denotes a separate virtual machine. The blocks across the top of the page represent different user-oriented programs (modeling and analytical systems, editors, etc.), and the blocks across the bottom of the page denote different data management systems, each running on its own VM. A user may access any modeling system and request a connection to any virtual machine. An interface VM associated with the user's machine provides the necessary communications interface between the user's analytical capability and the desired database system. With this configuration it is possible for a user to access the modeling or analytical capability with which he is most familiar, even though it may be running under an operating system different from the other available modeling or analytical capabilities. Thus the user is not required to learn new analytical capabilities.

In addition, since each VM may run any existing model or program under its normal operating system, such a configuration eliminates the need to devote resources to translating application packages and programs between operating systems.

Furthermore, the GMIS configuration permits interaction between application languages and programs not originally envisioned by their developers. For example an analytical package is greatly enhanced by having its data management capabilities extended. Hence a user of the APL/EPLAN analytical capabilities, for example, may request data that are stored and managed by SEQUEL database management capabilities.

3.4 GMIS Operation

A user of this system wishing a connection to another VM sends a message to the VM manager (depicted in Figure 4). The VM manager will automatically log in an interface VM. The interface VM loads into its address space the appropriate programs which can transfer commands from the chosen modeling machine to the chosen database management machine and can transfer data back from the database management machine to the modeling machine. The user may then access the appropriate database machine, which waits for an "external interrupt" to be initiated by the interface VM. The user, for example, may activate an APL model which could pass to the interface machine a SEQUEL command which could pass that command on to the SEQUEL database machine.

Figure 5 depicts a user console session that demonstrates such an interaction. In Figure 5 the user has previously configured an APL/EPLAN machine connected to a SEQUEL machine. This example is taken from the early discussion of the resolution of Figure 1: Why did the average miles per gallon of cars sold per month go down? Note in Figure 5 the user is communicating with an APL VM. The QUERY command is an APL function written for GMIS which sends the SEQUEL command to obtain Cadillac sales information from the SEQUEL machine (via the interface VM). The SEQUEL machine returns the requested data in a vector VOLUME.

To facilitate plotting, the user from the APL machine then converts the vector VOLUME into a time series using the EPLAN function *DF*. The user repeats this process to obtain data on the sales volume of Valiants. The user then activates the

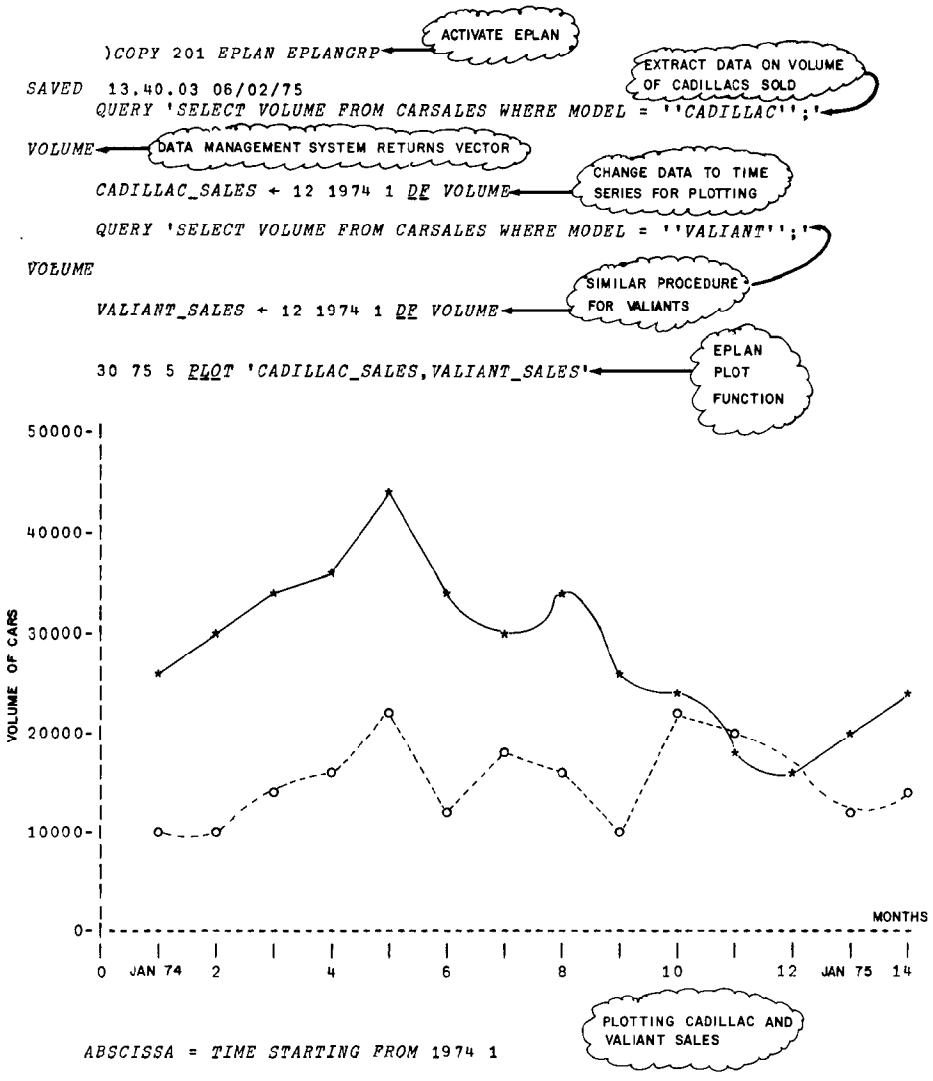

Fig. 5. Using the plotting function for reporting data
(Underlined symbols correspond to italic symbols in the text.)

EPLAN *PLOT* function which plots the sales volumes of Cadillacs and Valiants. (EPLAN [53] is an econometric modeling language consisting of a set of APL functions.)

Note that the solid line in Figure 5 represents Valiant sales, which appear to fall off during the energy crisis, while the dotted line represents Cadillac sales, which appear to have remained constant. This gives evidence to support a hypothesis discussed earlier in this paper which attempted to explain why average miles per gallon of cars sold seemed to go down during the energy crisis. The hypothesis

suggested that the affluent were buying big luxurious cars while others in lower income levels were simply not buying cars at all.

The modeling and analytic systems which are presently active on the GMIS configuration are TROLL, EPLAN, TSP, PL/1, MPSX, BMDP, DYNAMO, STATPACK II, and APL. The database systems that are presently running are SEQUEL and Query by Example. The APL Data Language [32] and VSAM [30] are presently being added.

3.5 Functions of the Virtual Machines

3.5.1 *Functions of Manager Virtual Machine.* The primary function of the manager VM is to respond to user requests to create the connections between the VMs servicing that user. The other function of the manager is to disconnect and automatically log out the appropriate interface VMs once the user has logged out.

To accomplish these functions several procedures were added to the user VM and the manager VM. When a user logs into his user machine he makes a request through his interface machine to connect to a database machine by sending a message to the VM manager. The message is sent by calling a message sending routine:

CALL IXESEND (UMID, message-address, message-length, message-code)

The IXESEND procedure uses the VM/370 experimental intermachine communications facility SPY [28] for sending a message to the manager VM. The user-initiated action causes the VM manager to receive an external interrupt. The external interrupt handlers which have been added to the manager VM perform the following: (1) check ID of sender for authorization; (2) look at the message located at message-address. If the message is to log in an interface VM then it will check to see if such a VM is already running. If not, it automatically logs one in (note that the VM manager has operator privileges, which permit it to log in other VMs). The manager VM then sends a message to that interface machine for it to load the appropriate interface module. The manager VM then sends a complete code message-code to the user VM. If the message at message-address were a terminate message, the manager would automatically log off the user's interface VM. Furthermore, the manager periodically checks all interface VMs to see if they have "parents," i.e. if the user VMs are currently logged in. If an interface VM does not have a parent, the manager VM automatically logs off the interface VM.

3.5.2 *Functions of Interface Virtual Machines.* The interface VMs provide mechanisms for user VMs to communicate with database VMs. When a user VM signals the manager VM to activate its interface VM, this user VM also indicates in which modeling or analytical environment it is currently running and to which database machine it wishes to send transactions. The manager VM uses this information to signal the interface VM to load the appropriate interface routines for the particular user environment/database system combination desired.

Since each interface routine is custom built to permit communications between a specific user environment and a specific database system, each user environment (e.g. PL/1, TROLL, APL) has a single standard communications interface written that uses the most efficient communication capabilities available in that environment. Likewise, each database system has a standard communication front end

build for it which receives transactions in a format that requires a minimal amount of preprocessing by the database machine. Any reformatting of transactions from the user or replies from the database system is handled by the interface routine which resides in the user's interface machine.

For user environments and database systems which can be interfaced through front ends written in PL/1 or Fortran, the SPY intermachine communications mechanism [5, 28] is used for highly efficient signaling and message sending. Some of the user environments which may use SPY include user-written PL/1 or Fortran applications programs, TSP, and BMDP [11]. Some of the database systems which may communicate using SPY include SEQUEL and Query by Example. Some other user environments, such as TROLL and APL, do not normally permit the user to call external routines directly, and thus one must use a spooling mechanism utilizing virtual card readers and punches for sending and receiving messages [16].

One example of this communications facility is the APL to SEQUEL interface (see Figure 6). A user in the APL environment first signals the manager VM to activate the user's interface VM and load the APL/SEQUEL interface routines. The user VM may then send a transaction to the interface VM by writing it to a Conversational Monitor System (CMS) file and spooling a card from its virtual card punch to the interface VM's virtual card reader, which generates an interrupt. The interface VM is alerted to the user's request by the interrupt, reads the transaction from the CMS file, reformats it for the SEQUEL database system, and sends the transaction to the SEQUEL VM via the SPY mechanism. After processing the transaction, the SEQUEL VM sends the reply to the interface VM via SPY again; the interface VM reformats the reply for APL, writes the reply to a CMS file, and signals the user VM running APL that the transaction is complete by spooling a card to its virtual card reader. The user VM may now read the reply from its CMS file and process it in any manner desired. This entire sequence is illustrated on the right-hand side of Figure 6.

3.6 Communication Mechanisms Betwen the Virtual Machines

The original philosophy of the VM concept was isolation [19]; that is, each virtual machine should be unaware that other VMs exist. Until recently, applications of VM technologies were consistent with this philosophy. Fortunately, with respect to technologies needed for decision support systems, researchers have recently developed mechanisms to facilitate communication. These include: the page swap method and the data move method [5, 28]; segment sharing [25]; channel-to-channel adaptor and virtual card punch and reader [16] available with standard release of VM/370 [29]. The page swap method has been implemented by IBM by using a VM enhancement of the IBM 370 DIAGNOSE instruction. This implementation, called SPY, can be thought of as a core-to-core transfer between the two communicating VMs. This is a very efficient mechanism for communicating between VMs. However, it requires the receiving VM to be capable of handling an external interrupt. Hence this mechanism is best used between VMs running programs that can be modified to call external PL/1, Fortran, or BAL routines, which would handle the interrupt and communications mechanisms. Under VM/370 the CMS [30] provides an operating system environment to modify, recompile, and reload these programs for use in GMIS. The communication mechanisms

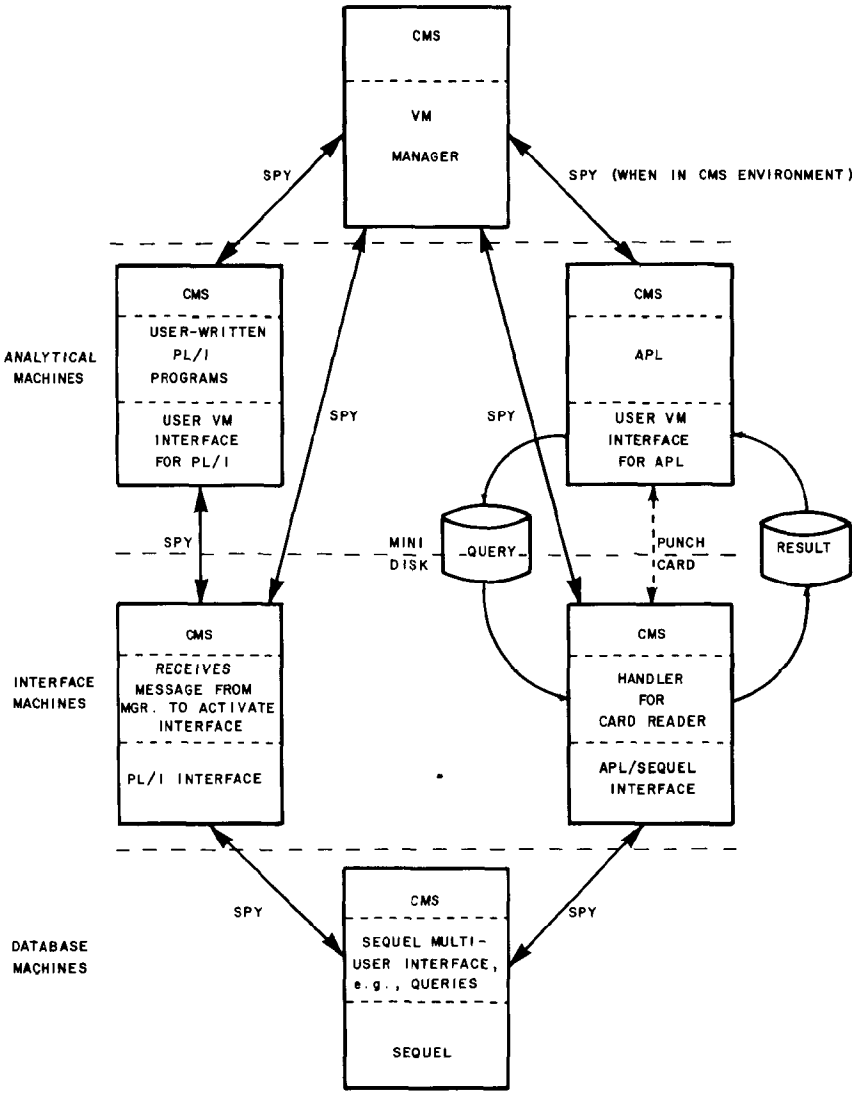

Fig. 6. Example of communication mechanisms used

used between the different classes of virtual machines in GMIS, as described in Section 3.5, are depicted in Figure 6 and summarized here:

Between the User Analytical Facility and the VM Manager. Since some modeling facilities would be difficult to modify to communicate directly with the manager VM, a separate communication program which runs under CMS is invoked before the modeling facility is activated. This program sends the necessary messages to the Manager. The user may then activate a modeling facility under CMS or other operating systems.

Between the Modeling VM and the Interface VM. For PL/1, TSP, and other modeling facilities running under CMS, the communication to the interface machine is via SPY (note that we modified TSP to run under CMS). However, for

systems like APL and TROLL that run under their own environments, communication is via minidisks, since standard versions of these systems have the capability of reading and writing disks, as well as punching and reading cards. The message is written on a shared minidisk. The interface VM is notified that such a message is waiting by punching a card on a virtual card reader. The interface VM reads that card and then reads the minidisk.

Between the Interface VM and a Database VM. SPY is used when the database VM is running in a CMS environment (e.g. in the case of SEQUEL and Query by Example). However, communication is via minidisk, virtual card readers, and punches for data management systems that do not run in a CMS environment (e.g. IMS in an OS/VS1 environment).

These communication facilities are explicitly shown in Figure 6, which depicts an example configuration of two analytical machines (user-written PL/1 programs running under CMS and an APL environment running under CMS) interfaced to a SEQUEL machine.

3.7 Functions of the Database Management Systems

The GMIS configuration has allowed the implementation of a data management capability which meets many of the requirements of decision support systems outlined in Section 2.2.

GMIS provides the user with access to an interactive relational data management system SEQUEL. This relational system has particular advantages for decision support systems, as it is able to provide a simple view of data. Policymakers and analysts have found that viewing data in the form of a table (relation) is conceptually simple. Furthermore, as we have discussed in Section 1, the structure of the data, the ways a user will access it, and addition and deletion of data all change frequently in such applications. The relational system provides mechanisms for facilitating these changes; thus there is no need to define a more complex data structure. The follow-on experimental system to SEQUEL called System R [4] holds even greater promise for use in decision support systems.

Although we have found that relational database systems are advantageous in certain public policy applications, most papers concerned with relational database technologies take their examples from inventory control areas or other unshaded areas of Figure 3 (e.g. [8, 9]). It is our feeling that applications like inventory control are areas where relational systems appear to have little advantage. In fact they may be at a disadvantage owing to performance problems over traditional systems.

Decision support systems have requirements not only for data manipulation but also for facilities for data analysis. Systems like TROLL, TSP, and APL/EPLAN are good data analysis facilities but have poor database facilities. Facilities like SEQUEL, IMS, etc., have good database capabilities but poor analytical or statistical capabilities.

The implementation of database systems in the particular VM environment of GMIS allows the enhancement of any data management facility by extending its analytical and statistical capabilities at minimal cost. This enhancement is accomplished by running additional analytical systems which communicate with the database machine.

A common requirement of a decision support system is to allow different groups working on similar problems access to the same database. Each group may be familiar with a different analytical system. The data needed by all groups may be maintained by one group. The GMIS-type VM configuration allows different users (each using different analytical systems) to access the same data management system.

Another requirement of decision support systems resulted from the fact that many data series needed by the decision-maker may be maintained in several different data management systems, and there is often not time to transport these data series to a common data management system. The GMIS configuration allows many data management systems to exist simultaneously. Any user or analytical system can access data stored in a variety of data systems.

In decision support applications it is often desirable for different (and often incompatible) application programs (e.g. models) to be able to interact with each other frequently. For example there may exist an operational national supply model for natural gas consumption in the United States (e.g. [37]). At the same time, there may exist a regional demand model for energy by sector (e.g. [3]). If a decision-maker wishes to study supply and demand of natural gas in New England, he may find it helpful to use these two models. The output of the first model (forecasted supply for the region) would be compared to the sum of output of the second model (demand by sector). Iterations of each model would then be performed until a balance occurred. However, the first model is written in TROLL, which operates under its own operating system and thus cannot be run on the same system as the second model, which is written in Fortran, running under OS. By bringing these two models up on the GMIS configuration (where each could access data generated from the other), their interactions would be facilitated through a common data management system.

3.8 Other Issues

We found the relational view of data particularly attractive for interactive public policy type applications. However, we recognize both the experimental nature of these relational systems and the existence of many data series in more standard widely used data management systems, e.g. IMS. Hence we have provided for the availability of systems like IMS.

Additional advantages of the GMIS approach include increased security among users of such a system; that is, security is improved over the more conventional method of operating different modeling capabilities that were compatible in a multiprogrammed environment underneath the same operating system. This increased reliability of GMIS is discussed elsewhere [19, 40] and is an intuitive result of the fact that malicious or unintentional violations by the user must not only subvert the protection mechanisms of the operating system under which it is running, but also must subvert the protection mechanisms of the virtual machine monitor (VMM) if these violations are to affect another user. Hence this hierarchical protection mechanism can provide much higher security. This concept is still somewhat controversial [20].

Since VM/370 software has been developed in such a way that each VM can be accessed via a console, programs that were previously batch oriented behave much

as though they were interactive; that is, a program can be created online, edited, and submitted for processing via a console.

Our experience with the GMIS approach in several application areas to date has been very productive. The performance implications of this configuration are discussed elsewhere [14]. We feel that further studies on cost benefit analysis and on increased effectiveness of users of this sort of system will quantitatively confirm our observations of the benefits of this approach [38].

4. NEW ENGLAND ENERGY MANAGEMENT INFORMATION SYSTEM EXAMPLE APPLICATION

In this section we present an example to show the interaction necessary in a decision support system between a database system and an analytical system. More importantly this example was chosen to show that the computational capabilities advocated in this paper have a large comparative advantage.

4.1 Purpose

This is a very detailed example. Its purpose is to show in a real setting the importance of:

(1) the interaction between an analytical system and a data management system, like that of GMIS;
(2) a flexible data management system for real decision support applications.

More specifically, this example shows:

(1) the amount of data manipulation required for validation;
(2) that queries had to be made to the data which were not originally planned for;
(3) that the interaction between an analytical capability and a database capability is frequent and is best accomplished in a user interactive mode;
(4) that other data series (not originally planned for) had to be introduced long after the study started;
(5) that an interactive analytical facility was helpful if not absolutely necessary for quickly responding to a problem.

In spite of the fact that the entire study was a complex one (requiring sophisticated data manipulation and complex analytical functions), it was accomplished in one week, largely because a computational facility (GMIS) with many of the features advocated in this paper was available.

4.2 New England Energy Management Information System

The example chosen here is taken from the NEEMIS Project [13, 18], which uses GMIS as its computational facility. The NEEMIS Project has been sponsored by the New England Regional Commission, and its primary focus is on assisting the individual New England states and the overall region with energy policy decision-making. The Project consists of four thrusts: making the NEEMIS computational facility available to the states; maintaining relevant energy data series; maintaining energy-related computational models; and providing a group of regional energy specialists accessible to regional policymakers.

4.3 Example Study Within the NEEMIS Project

Let us explore the use of the GMIS computational capability in one recent study [15]. One goal for the policymaker with respect to energy is to increase the supply of petroleum or to reduce the demand for petroleum in the region. Since it is unlikely that oil will be discovered quickly in the New England region, if at all, considerable focus in the region has been directed toward reducing demand and thus toward conservation efforts in energy. Residential space heating (home heating) consumes over 20 percent of all energy used in New England [3] and comprises over 10 percent of all energy consumed in the United States [12]. Oil is the source of over 70 percent of New England's home heat, and virtually all of this oil is imported into the region [44]. Hence even a small reduction in home heating oil consumption could result in a considerable economic improvement in the region. The question for the policymaker is how can he reduce consumption of fuel for home heating, using the handles over which he has some control, namely, price and awareness (e.g. by raising the price of oil or by an advertising campaign).

To assist the decision-making process, the NEEMIS Project performed a study to determine the relationship of price, awareness, and consumption. To establish this relationship consumption data was gathered from a sample of homeowners in New England. Scott Oil Company made available delivery data, specifically delivery data on 8000 individual homes within the suburban Boston area. The data are associated with the years 1973 to 1975, a period in which marked price changes, shortages, and behavioral changes occurred, hence providing an opportunity to study the effects of these changes. The delivery data of the sample cover a period in which there were perhaps the greatest price changes in recent history (for instance 1973–1974 shows a 50-percent increase of price of oil to homeowners). It was also a period in which awareness of energy use, shortages, and expected price increases was great. Thus the data afford an unusual opportunity to calculate short-term elasticities. Weather data were gathered from 38 weather stations in New England. Additional data were gathered as they were needed.

Let us examine the computational steps that were required to calculate the short-term elasticity of consumption to price. That is, if the policymaker raises price by a certain amount, by how much could he expect consumption to be reduced? The purpose of this exercise is to give the reader some feeling for the operations needed in a decision support system.

To analyze consumption as a function of factors that vary over time, a regression model [48] was established that related change in consumption per degree-day (CPD) to a function of price and awareness. To normalize the effects of weather, consumption of households is expressed in gallons of oil consumed per degree-day. Degree-days are a weighted average of daily temperatures as they vary from a mean of 65 degrees. Because price data were available on a monthly basis, we may write this expression as

$$\overline{\mathrm{CPD}}_m = A_1 + \sum_{i=2}^{n} A_i X_i .$$

The dependent variable ($\overline{\mathrm{CPD}}_m$) is the average consumption per degree-day month of all consumers of the Scott Oil sample who received frequent oil deliveries (five or more each season) during the three heating seasons.

The independent variables (X_2, X_3) used in the model were price and awareness.

The price variable was set equal to the average price (in cents per gallon) of the oil company involved during the corresponding month. (A_1 is a constant term. A_2 and A_3 are coefficients of the independent variables.) After much discussion, the awareness variable chosen was the number of frontpage headline columns of energy-related articles in the Thursday and Sunday *Boston Globe* accumulated over the corresponding month. In this manner a monthly data series for this variable was compiled.

4.4 Computational Steps

The computational steps involved in developing the model and preparing the data were as follows: (1) validate the data; (2) select the applicable data; (3) analyze the biases involved in such a selection; (4) make computations on the data for creating the dependent variable of the model; (5) run the model and introduce various mathematical alternatives to the model to improve its statistical properties; and (6) use the most representative version of the model to compute the elasticities.

Because of the advantages of the computational facility chosen, all the above steps were accomplished and analyzed in less than one week.

4.4.1 *Validate Data.* In the first step the data had to be validated. The data were provided by the oil company in a form that associated with each customer the amount of each delivery which that customer received in the years 1972–1975. Many simple computations were performed on those data to check their validity. For example, for each customer, we added all deliveries in a year and compared them to the average yearly deliveries. Wide variations were examined more closely. *Note that while the computations were simple, often the types of accesses to the data were quite selective. Furthermore, the number of accesses and tests was large.*

4.4.2 *Select Applicable Data.* The second step was selecting the desired data. Consumption data were needed for the proposed model to calculate $\overline{\text{CPD}}_m$ (the average monthly consumption per degree-day for the entire sample). However, with our source of consumption data (the oil company delivery records), consumption can only be measured whenever a delivery occurs. Consumption in a period is equal to the quantity delivered, where the period is defined as the time between this delivery and the previous one. Hence those customers with frequent deliveries provide more reflective data on consumption, since consumption is monitored more often. Therefore data used to calculate the average CPD per month were accessed by selecting only those customers with frequent oil deliveries. *Note that here is a query on the data that was not envisioned beforehand.*

4.4.3 *Analyze Biases.* Taking a subsample from the entire sample that includes only customers with frequent oil deliveries introduces a problem for the policymaker, that is, biases. Therefore it is necessary to make a bias analysis (step 3). To perform this analysis, we need to test the hypothesis that the subsample generated from step 2 has the same consumption habits as the sample as a whole. We may compare the CPD averaged over an entire year ($\overline{\text{CPD}}_y$) for both the subsample and the entire sample to test this hypothesis. In order to do this in the analytical facility, a program was written that calculates consumption for the entire year by summing all the deliveries made to an individual customer and dividing by the number of degree-days that occurred during that year. (Degree-day data are ac-

cessed from the database.) This is then done for all customers to get the average $\overline{\text{CPD}}_{y,i}$ for each customer (i). $\overline{\text{CPD}}_{y,i}$ is then averaged over all customers in the entire sample and in the smaller sample, giving $\overline{\text{CPD}}_y$ for the sample and for the subsample. A statistical routine (written in another language) is then invoked to perform statistical tests to determine the significance of any differences if they exist. The analytical system was used to perform the calculations. The data management system was used to access the data by the criteria of all customers with frequent oil deliveries. *Note the interaction between the analytical system and the data management system.*

4.4.4 *Compute Dependent Variable.* Step 4 involved the computation of the data used in the dependent variable $\overline{\text{CPD}}_m$, and in this step even more elaborate interaction between an analytical facility and a database facility was necessary. The following procedure was used: (a) Consumption for individual delivery periods was calculated for each customer from delivery data; (b) CPD for each customer for each delivery period was calculated by dividing the degree-days for each delivery period into the consumption of that period; (c) the average CPD for all customers $\overline{\text{CPD}}_d$ for a particular day was obtained by averaging CPD for each customer for that day; and finally (d) the average CPD for each day of a month $\overline{\text{CPD}}_m$ was calculated by summing CPD for each day of a month and dividing by the number of days.

Note that from a computational point of view (using Figure 4, GMIS) for substep (a), it was necessary to access the data associated with the subsample for the amount of oil that was delivered during a period to each customer and for the dates of that period. For substep (b), it was necessary to access the weather data to determine the number of degree-days in that same period. The calculation of CPD for a delivery period was performed in the analytical system and then had to be stored back in the data management system. The computational aspects of substep (c) involved creating 365 individual pieces of data that correspond to the average consumption (over all of the customers) per degree-day for a particular day ($\overline{\text{CPD}}_d$). To compute $\overline{\text{CPD}}_d$ one must access for a particular day each customer's CPD (as calculated in substep (c)) and then sum $\overline{\text{CPD}}_d$ for a particular day over all customers and divide by the number of customers to produce the resulting data series $\overline{\text{CPD}}_d$ (average CPD for all customers for each day in the three heating seasons under consideration). The computation involved in step (d) involved accessing this $\overline{\text{CPD}}_d$ series and summing it for each day in a particular month, then dividing by the number of days in the month to obtain $\overline{\text{CPD}}_m$. *Note that other data series, e.g. weather, had to be introduced, accessed, and used. Furthermore, this entire step was accomplished in a matter of hours owing to the interactive characteristics of the computational facility.*

4.4.5 *Run and Adjust Model.* Step 5, the computation associated with running the model, involved activating a standard regression package that existed in our facility as the EPLAN package. In such a regression package one specifies the dependent variables and all the independent variables. Data for those variables must be obtained from the data management system and passed back to the analytical system where the regression is performed. The user receives output statistics as to the significance of each of the coefficients in the regression, as well as overall statistics as to the goodness of the model.

The first such regression resulted in a relatively poor r^2 statistic, and so a slight modification of the model was made. Specifically it was felt that it would be more reflective to take the log of the awareness variable since the first article in a newspaper would have the most effect, with each article in subsequent issues having less effect. After this modification was made, the resulting statistics improved (that is the r^2 statistic as well as each of the coefficients). We then felt it would be more accurate to lag the awareness variable by one month, as perhaps a customer's reaction to the shortage situation would not occur until some time after this customer was made aware of the situation. Lagging the awareness variable by one month again improved the r^2 statistic as well as the significance of each of the coefficients.

It was then noted that the price data should be adjusted for inflation. Hence another data series was established in the data management system, containing a set of inflation indicators for home heating fuel. The modeling system then used these standard inflation indicators to adjust the price data series. The model with the best statistics used adjusted prices, a lagging of awareness by one period, and a log of awareness. *Note the comparative advantage that an interactive system gave in allowing for quickly interacting and changing the model. Note that yet another unexpected data series was introduced. Furthermore, each new version of the model was made and examined in a matter of minutes.*

4.4.6 *Compute Elasticities.* With respect to step 6, the result to the policymaker was the calculation of the elasticity of price with respect to demand; an APL program was used. That program calculated the ratio of percentage change in price to percentage change in demand. The elasticity calculated from the best model was $-.17$; that is, a 1-percent increase in price would produce a decrease in consumption of .17 percent. This value of elasticity applies to the New England region and applies in the short term. A policymaker should be aware of this important number.

The steps required to compute this elasticity further support the need for the close interaction of analytical capability and database management systems and for the capability to quickly incorporate new data series such as inflation indicators, incorporate existing programs such as those used in the bias analysis, and incorporate different data series and access them at the same time (such as accessing the Scott Oil data series as well as weather data series supplied by the weather bureau). *Note that we were able to accomplish this entire computation in under one week. That is not to say that others could not now duplicate that computation (now that the problem is defined, the data defined, etc.) in such a short time using any number of computational facilities, but we do say that it would have been nearly impossible or very difficult to have accomplished that task by using a traditional system owing to the need for haste and the changing perception of the problem.*

5. CONCLUSION

Common to many of the problems facing our country is the necessity to support decisions. Central to the process of supporting decisions is information. Database system technology as it now stands lies at the heart of the technology necessary for decision support systems but is not adequate in itself for such systems. Decision support systems, as has been shown in this work, are different from the traditional management and operational control systems to which database systems have been

successfully applied in the past. The differences are due primarily to the problems being addressed by decision support systems. That is, the nature of these problems is such that they are constantly changing, the data needed to solve them are not always known, solutions to these problems are needed quickly, and attention must be given to the cost of developing solutions to these problems. Our experience in using database systems, analytical systems, and other software in decision support systems for energy-related areas (the NEEMIS work discussed in this article) and other private and public sector areas only further confirms our realization of the inadequacies of existing database management systems and technologies as far as the needs of decision support systems are concerned. The approach explained here alleviates some of the deficiencies of traditional database systems. By developing a framework in which different database systems and different analytical and modeling systems may be integrated together within the same system, the transfer costs and time loss that would necessarily be involved in integrating existing programs and existing data series to solve particular decision problems have been reduced. This paper is a call for further attention to be given to perhaps the most promising technology available for dealing with an ever more complex world. Attention must be given within an application framework to the development of complementary technologies and the extension of existing database system technologies for the development of effective decision support systems.

ACKNOWLEDGMENT

Special recognition is given to the members of the San Jose Research Center who made their SEQUEL system available to us and assisted us in adapting it for GMIS and for the NEEMIS energy use. Members of the IBM Cambridge Scientific Center have played key roles in the design and implementation of GMIS. Ray Fessel designed and implemented the APL interface to GMIS, participated in the design and implementation of the multiuser interface, helped to correct bugs in the experimental code, and participated in the initial design of the present GMIS configuration. We also acknowledge Stuart Greenberg for coordinating the Joint Study and Richard MacKinnon for overseeing it.

It is difficult to acknowledge all the M.I.T. personnel who worked on the GMIS system; more than forty students, five staff members, and three faculty members all played important roles. I would like to recognize the following people: Jenny Rankin for editing and Marvin Essrig for assistance with the development of the leading indicators example. Recognition is given to Louis M. Gutentag, of M.I.T., who supervised the GMIS implementation and designed much of the system. He specifically designed and implemented the user interface to SEQUEL known as TRANSACT. He played a major role in the design of the present GMIS configuration and supervised several students who implemented various parts of the system. Some of those students include Robert Selinger who implemented the PL/1 interface routines and Chat Yu Lam who implemented the VM manager and communication schemes for the present GMIS.

Walter Fischer, an IBM visiting scientist on leave from the University of Munich, and Peter DiGiammarino and Richard Tabors, both of M.I.T., assisted with the price elasticity example. Peter DiGiammarino implemented the models, Richard

Tabors helped in the formulation of the models, and Walter Fischer performed the validation, the bias analysis, and the computation of $\overline{CPD}_m$. We acknowledge the Scott Oil Company for making available the data used in the computation of elasticities.

We thank Louis M. Gutentag, Peter DiGiammarino, David Hsiao, and J.F. Rockart for their helpful comments.

REFERENCES

1. ALTSHULER, G., AND PLAGMAN, B. User/system interface within the context of an integrated corporate data base. Proc. AFIPS 1974 NCC, Vol. 43, AFIPS Press, Montvale, N.J., pp. 27–33.
2. ANTHONY, R.N. *Planning and Control Systems: A Framework for Analysis.* Graduate School of Business Administration, Harvard U., Boston, Mass., 1965.
3. ARTHUR D. LITTLE, INC. Preliminary Projections of New England's Energy Requirements. The New England Regional Commission, Boston, Mass., 1975.
4. ASTRAHAN, M.M., ET AL. System R: Relational approach to database management. *ACM Trans. on Database Systems 1*, 2 (June 1976), 97–137.
5. BAGLEY, J.D., FLOTO, E.R., HSIEH, S.C., AND WATSON, V. Sharing data and services in a virtual machine system. Proc. ACM SIGOPS Fifth Symp. on Operating Systems Principles, Nov. 1975, pp. 82–88.
6. CHAMBERLIN, D.D., AND BOYCE, R.F. SEQUEL: A structured English query language. Proc. ACM SIGFIDET Workshop, Ann Arbor, Mich., May 1974, pp. 249–264.
7. CLARKSON, G.P.E., AND POUNDS, W.F. Theory and method in the exploration of human decision behavior. *Indust. Manage. Rev. 5*, 1 (Fall 1963), 17–27.
8. CODD, E.F. A relational model of data for large shared data banks. *Comm. ACM 13*, 6 (June 1970), 377–387.
9. DATE, C.J. *An Introduction to Database Systems.* Addison-Wesley, Reading, Mass., 1975.
10. DAVIS, G.B. *Management Information Systems: Conceptual Foundations, Structure, and Development.* McGraw-Hill, New York, 1974.
11. DIXON, W.J., Ed. *Biomedical Computer Programs.* University of California Press, Berkeley, Calif., 1975.
12. DOLE, S. Energy use and conservation in the residential sector: Regional analysis. Rep. No. R-1641-NSF, RAND Corp., Santa Monica, Calif., 1975.
13. DONONAN, J.J., AND MADNICK, S.E. Institutional and ad-hoc decision support systems and their effective use. Working Paper, Center for Information Systems Research, M.I.T., Cambridge, Mass. Oct. 1976.
14. DONOVAN, J.J. A note on performance of VM/370 in the integration of models and databases (to appear in *British Comptr. J.*, Aug. 1977).
15. DONOVAN, J.J., AND FISCHER, W.P. Factors affecting residential heating energy consumption. Working Paper No. MIT-EL-76-004WP, M.I.T. Energy Lab., M.I.T., Cambridge, Mass., July, 1976.
16. DONOVAN, J.J., AND JACOBY, H.D. GMIS: An experimental system for data management and analysis. Working Paper No. MIT-EL-75-011WP, M.I.T. Energy Lab., M.I.T., Cambridge, Mass., Sept. 1975.
17. DONOVAN, J.J., AND JACOBY, H.D. New England energy management information system. Research proposal submitted to the New England Regional Commission, Boston, Mass., 1973.
18. DONOVAN, J.J., AND KEATING, W.R. NEEMIS: Text of governors presentation. Working Paper No. MIT-EL-76-002WP, M.I.T. Energy Lab., M.I.T., Cambridge, Mass., Feb. 1976.
19. DONOVAN, J.J., AND MADNICK, S.E. Hierarchical approach to computer system integrity. *IBM Systems J. 14*, 2 (1975), 188–202.
20. DONOVAN, J.J., AND MADNICK, S.E. Virtual machine advantages in security, integrity, and decision support systems. *IBM Systems J. 15*, 3 (1976), 270–278.

21. DONOVAN, J.J., FESSEL, R., GREENBERG, S.A., AND GUTENTAG, L.M. An experimental VM/370 based information system. Proc. Int. Conf. on Very Large Data Bases, Framingham, Mass., Sept. 1975, pp. 549–553.
22. DONOVAN, J.J., GUTENTAG, L.M., MADNICK, S.E., AND SMITH, G.N. An application of a generalized management information system to energy policy and decision making: The user's view. Proc. AFIPS 1975 NCC, AFIPS Press, Montvale, N.J., pp. 681–686.
23. GOLDBERG, R.P. Survey of virtual machine research. *Computer 7*, 6 (June 1974), 34–35.
24. GORRY, G.A., AND SCOTT MORTON, M.S. A framework for management information systems. *Sloan Manage. Rev. 13*, 1 (Fall 1971), 55–70.
25. GRAY, J.N., AND WATSON, V. A shared segment and inter-process communication facility for VM/370. Res. Rep. RJ 1579, IBM Res. Lab., San Jose, Calif., Feb. 1975.
26. GUTENTAG, L.M. GMIS: Generalized management information system—an implementation description. M.S. Th., Sloan School of Management, M.I.T., Cambridge, Mass., June 1975.
27. HALL, R.E. TSP manual. Harvard Tech. Rep. No. 12, Harvard Inst. of Economic Research, Cambridge, Mass., April 1975.
28. HSIEH, S.C. Inter-virtual machine communication under VM/370. Form No. RC 5147, IBM Thomas J. Watson Res. Ctr., Yorktown Heights, N.Y., Nov. 1974.
29. IBM (1). IBM virtual machine facility/370: Introduction. Form No. GC20-1800, IBM, White Plains, N.Y., 1976.
30. IBM (2). IBM virtual machine facility/370: CMS user's guide. Form No. GC20-1819, IBM, White Plains, N.Y., 1976.
31. IBM (3). APL econometric planning language. Form No. SH20-1620 (Product No. 5796PDW), IBM, White Plains, N.Y., 1975.
32. IBM (4). APL data language. Form No. SB21-1805, IBM, White Plains, N.Y., 1975.
33. IBM (5). Information management system: General information manual. Form No. GH20-1260, White Plains, N.Y., 1975.
34. IBM (6). Mathematical programming system extended/370 (MPSX/370): Basic reference manual. Form No. SH19-1127, IBM, White Plains, N.Y., 1974 & 1975.
35. LICKLIDER, J.C.R. Man-computer symbiosis. *IRE Trans. on Human Factors in Electron. 1*, 1 (March 1960), 4–11.
36. LITTLE, J.D.C. Models and managers: The concept of a decision calculus. *Manage. Sci. 16*, 8 (April 1970), B466–B485.
37. MACAVOY, P.W., AND PINDYCK, R.S. Alternative regulatory policies for dealing with the natural gas shortage. *Bell J. of Econ. and Manage. Sci. 4*, 2 (Fall 1973), 454–498.
38. MADNICK, S.E. INFOPLEX: Hierarchical decomposition of a large information management system using a microprocessor complex. Proc. AFIPS 1975 NCC, AFIPS Press, Montvale, N.J., pp. 581–586.
39. MADNICK, S.E. Time-sharing systems: Virtual machine concept vs. conventional approach. *Modern Data 2*, 3 (March 1969), 34–36.
40. MADNICK, S.E., AND DONOVAN, J.J. Application and analysis of the virtual machine approach to information system security and isolation. Proc. ACM Workshop on Virtual Comptr. Systems, Harvard U., Cambridge, Mass., March 1973, pp. 210–224.
41. MADNICK, S.E., AND DONOVAN, J.J. *Operating Systems*. McGraw-Hill, New York, 1974.
42. M.I.T. ENERGY LABORATORY. Information systems to provide leading energy indicators of energy sufficiency: A report to the Federal Energy Administration. Working Paper No. MIT-EL-75-004WP, M.I.T., Cambridge, Mass., June 1975.
43. NATIONAL BUREAU OF ECONOMIC RESEARCH. *TROLL Reference Manual*. Technology Square, Cambridge, Mass., 1974.
44. NEW ENGLAND FUEL INSTITUTE. *Yankee Oil Man*. Fuel trade and fact book, Boston, Mass., March 1974.
45. NEWELL, A., SHAW, J.C., AND SIMON, H.A. Report on a general problem-solving program. Proc. Int. Conf. on Inform. Processing, UNESCO House, Paris, 1959, pp. 256–264.
46. PAKIN, S. *APL/360 Reference Manual*. Science Research Associates, Chicago, Ill., 1972.
47. PARMELEE, R.P., PETERSON, T.I., TILLMAN, C.C., AND HATFIELD, D.J. Virtual storage and virtual machine concepts. *IBM Systems J. 11*, 2 (1972), 99–130.

48. PINDYCK, R., AND RUBINFELD, D. *Econometric Models and Economic Forecasts*. McGraw-Hill, New York, 1976.
49. PLAGMAN, B.K., AND ALTSHULER, G.P. A data dictionary/directory system within the context of an integrated corporate data base. Proc. AFIPS 1972 FJCC, Vol. 41, AFIPS Press, Monvale, N.J., pp. 1133–1140.
50. POUNDS, W.F. The process of problem finding. *Indust. Manage. Rev. 11*, 1 (1969), 1–19.
51. PUGH, A.L., III. *DYNAMO User's Manual*. M.I.T. Press, Cambridge, Mass., 1961.
52. ROCKART, J.R., AND SCOTT MORTON, M.S. *Computers and the Learning Process*. Prepared for Carnegie Commission on Higher Education, McGraw-Hill, New York, 1975.
53. SCHOBER, F. EPLAN: An APL-based language for econometric modelling and forecasting. IBM Scientific Ctr., Philadelphia, Pa., 1974.
54. SCOTT MORTON, M.S. *Management Decision Systems: Computer Based Support for Decision Making*. Div. of Res., Graduate School of Business Administration, Harvard U., Boston, Mass., 1971.
55. SIEGEL, A.J. Unions, employers, and the computer. *Technology Rev. 72*, 2 (1969), 36–41.
56. SIMON, H.A. *The New Science of Management Decision*. Harper & Row, New York, 1960.
57. SMILLIE, K.W., Ed. STATPACK II, 2nd ed. Publication No. 17, Dep. Comptng. Sci., U. of Alberta, Edmonton, Alberta, Canada, 1969.
58. SMITH, G.N. Internal Intermediate Language, Version 2. Manage. Sci. Group, Sloan School of Management, M.I.T., Cambridge, Mass., Nov. 1974.
59. SUMMIT, R.K. DIALOG II User's Manual. Rep. No. 6-77-67-14, Lockheed Res. Lab., Palo Alto, Calif., March 1967.
60. URBAN, G.L. Building models for decision makers. *Interfaces 4*, 3 (May 1974), 1–11.
61. ZANNETOS, Z.S. Toward intelligent management information systems. *Indust. Manage. Rev. 9*, 3 (1968), 21–38.
62. ZLOOF, M.M. Query by example. Proc. AFIPS 1975 NCC, Vol. 44, AFIPS Press, Montvale, N.J., pp. 431–437.

Received May 1976; revised July 1976