An application of a generalized management information system to energy policy and decision making—The user's view

by JOHN J. DONOVAN, LOUIS M. GUTENTAG, STUART E. MADNICK and GRANT N. SMITH

Massachusetts Institute of Technology
Cambridge, Massachusetts

Motivations for flexible systems for energy

As a result of recent disruption in the world petroleum market and rapid price increases, the United States is in the process of developing energy policies that will lead to a greater degree of energy self-sufficiency, and to a reduced level of vulnerability to interruption of supply from abroad.

New England is particularly susceptible to disruption in energy supplies, as we are "at the end of the pipeline."

One advantage of the market system is that public officials can get by without knowing much about the details of the operation of most sectors of the country. Many goods and services are produced, allocated over space and time, and delivered to consumers without government intervention and with no need for a public record of how things are done. When events occur that call for government efforts to influence markets, however, a dearth of public information can be a crucial barrier to effective policymaking.

The need for information, hence an information management system, is obvious in a crisis situation. However, there also exists a need for energy information in a non-crisis situation to aid in a wide set of tasks:

- studies of the economic impact of various events in the energy sector
- studies of the location of major energy facilities (ports, refineries, etc.)
- development of early warning indicators of problems in regional energy supply
- provision of information for special studies of environmental impacts, conservation efforts, price trends, etc.

Our objective is:

To establish a facility (for storing and validating data, retrieving data, interpreting and analyzing data, and constructing and applying models using those data), which will facilitate New England energy policy analysis and decisions.

A system to support the objectives outlined would not be adequately represented by, for example, an accounting system. The accounting system operates on a well-defined set of data in a well-defined way. Neither data nor operations are subject to rapid alternations. Furthermore, the data is relatively "clean", i.e., from consistent, high-quality sources.

For the purposes of the energy information system, the problem area being addressed is not constant. It changes when changes in perception arise, which may be for any number of reasons. This has the effect of changing both the data required and the format of data required far more rapidly than the reporting and data gathering procedures can be altered to reflect the new needs. As such, the already inaccurate data become rapidly less suited to the task at hand.

Furthermore, with change occurring so frequently, it is imperative that the system can be modified to meet the change without incurring prohibitive expenses.

While these requirements are certainly true in the energy information system, they are by no means unique to it. Our approach has thus been to meet the needs of the energy system without actually implementing an energy-specific system. Rather, we have concentrated on constructing a Generalized Management Information System (GMIS) that meets requirements of extreme flexibility, acceptable costs, and simultaneously serving a diverse user group. This paper is addressed to a particular instance of the GMIS, namely, its use in the New England Energy Management Information System (NEEMIS), and more specifically, to the user view of the system rather than the implementation.

DESCRIPTION OF NEEMIS

Keeping in mind the ultimate purpose of NEEMIS—to provide a facility to aid public policymakers in energy decisions in New England—we recognize several classes of users of the NEEMIS facility. In this section we shall briefly explain what facilities each class of user will have. The precise syntax of intermediate languages and implementation details are described elsewhere.
In the NEEMIS facility, we wish to give users increasingly more powerful tools. Figure 1 depicts 4 classes of users as factors:

- Non-technical—e.g., a state energy officer. His objective is to get answers to questions and report.
- Well-trained—e.g., a specialist within a state energy office who has been trained in the use of the system.
- Researcher—e.g., an economist with some computer background who wishes to build a model for a special study.
- Systems analyst/programmer—e.g., a computer professional. He may wish to add a new table to the system or change the protection rights on an existing data series.

Looking across the table in Figure 1, we see the tools available to users of NEEMIS. Although all levels and facilities of the system are available to all users, it is unlikely that users will venture outside of those tools designated by “X.” Increased sophistication on the part of any one user will, of course, qualify him/her for a different category.

The tools of the system have been designed in such a way that the interests of the various user groups are met. Let us proceed to briefly describe the facilities at each level.

**Relational Operator and PL/1 Facility**

At this level, the user sees all data as being stored in relations. This includes not only regular entered data, but also system data, all access control data, etc. The user at this level has at his command thirteen set-oriented relational operators that are used to perform all operations on all data. It is important to note that user data, system data, access control data, etc., are all accessed in a consistent manner via these thirteen operators that are based on the relational model of data, which have their roots in logical systems and predicate calculus. The operators available in NEEMIS are described in detail in Reference 1.

Since these operators appear as PL/1 subroutine calls within NEEMIS, the user at this level also enjoys all the power of PL/1.

Notice that both PL/1 and relational operators require precise use and exhibit low tolerance for error.

**Data definition facility**

A user at this level has facilities to specify and create relations. We call this facility the Data Definition Language (DDL). The DDL will accept a data specification and will produce an appropriate relational data base, which is then incorporated into the system. The DDL also provides a facility for loading bulk data into the newly constructed relational system from punched cards, magnetic tapes, or magnetic disk files.

In the establishment of a new relation, the system tables are modified to include data about this new relation, as well as provision for specification of access control, etc.

Also available at this level is on-line help with commands, and extensive diagnostics.

An example of the use of the DDL facility follows. ("Domain" means a column of the "relation," or matrix.)
Example:

With most information management systems, the design of the system—that is, the design of the data base—is a vital step in the operation. If done incorrectly, it is often impossible, and usually extremely costly in dollars and man years to restructure the data base to more ably suit the needs.

Not so with NEEMIS. In fact, during the summer we experimented with three different designs in the course of a single month. The DDL permits specification of the data base on-line, and extremely rapidly. A sample session might be:

Example:

```
system: ENTER COMMAND:
user: define domains
system: . (. = "ready for input")
user: name character, soc_sec_# numeric (9),
user: supplier choice (gulf, exxon, mobil),
user: address character;
system: ENTER COMMAND
user: create relation
system: employee (name, soc_sec_#, address)
     (primary key: soc_sec_#),
     fuel_data (soc_sec_#, name, supplier)
     (primary key: soc_sec_#, required: supplier);
system: RELATIONS DEFINED
user: define synonym: soc_sec_# = 'ss';
system: SYNONYM ENTERED
user: stop.
```

This session would establish the two relations, and permit data to be entered immediately.

Query facility

At this level a user can specify queries of data stored in relations. The user uses a rigid syntax for his queries that we sometimes call “cryptic” English. More specifically, we call this facility a Data Manipulation Language (DML).

An internal document describes a complete DDL and DML that has been specified at M.I.T.\textsuperscript{16} Other attempts at implementing a query facility based on the relational model include: MACAIMS,\textsuperscript{8} SEQUEL,\textsuperscript{9} COLARD,\textsuperscript{10} RIL,\textsuperscript{11} and M.I.T.'s RDMS.

This facility is available for querying relations established via the DDL or possibly the relational operator facility (see earlier sections of this paper).

The commands, although conforming to a rigid syntax, employ English-like keywords, are quite easy to learn and readily readable. Once again, all data, including system data, are accessed in a consistent manner; and access control specification is an integral part of DML.

Let us give two examples here of our DML query commands.

We assume that the following four tables have been created using the DDL. The first table is named ‘terminal’ and it has six columns: terminal id, name, etc.

```
TERMINAL (TERMINALID, NAME, CITY, STATE, ZIP CODE, AFFILIATION)
SUPPLY CAPACITY (TERMINALID, FUELTYPE, FUELAMI, DATA)
SUPPLIER (SUPPLIERNO, NAME, VOLUME, FUELTYPE, DISTNO)
DISTRIBUTORS (DISTNO, NAME, ADDRESS, CITY, STATE, INVENTORY, FUELTYPE)
```

The following are sample queries against a data base that contains the above tables:

**Question 1**

```
DISPLAY NAME, AFFILIATION, CITY
FOR STATE='MASSACHUSETTS'
```

This question causes the listing of the name, affiliation and city of all terminals in the state of Massachusetts.

**Question 2**

```
DISPLAY NAME FOR FUELAMT>1000, FUELTYPE='GASOLINE', CITY='LYNN'
```

This lists the names of all terminals in Lynn that have over 1000 gallons of gasoline capacity.

The display command is but one of several available. All commands employ consistent syntactic constructs and are equally readable.

There is, again, extensive on-line help with commands available, as well as explanatory diagnostics. No high-level user should have to see “protection exception at location OFE1A3”!

The modeling facility

A user of this facility may construct and activate a model interactively via provision of a set of functions called from APL. These functions include regression routines, plotting routines, time series modeling routines, etc., in addition to the standard APL facilities. The language used for modeling is a superset of APL—i.e., APL with additional facilities. The data that the model uses may be retrieved directly from stored in the relations (see previous section).

This APL-oriented modeling facility is the current standard. Inclusion of additional or different modeling languages, however, poses little problem (see later section).
what are the terminals and their cities for 'kennebec' county?

TRANSLATION:
D TERMINAL.OPNAME,TERMINAL.CITY FOR TERMINAL.COUNTY='KENNEBEC';

TERMINAL.OPNAME TERMINAL.CITY
MOBIL OIL CORP HALLOWELL
NORTHEAST PETROLEUM AUGUSTA
GULF OIL AUGUSTA
AGWAY PETROLEUM HALLOWELL

DISPLAY COMPLETE.

what are the capacities and fuel types for the 'mobil oil corp' terminal in the city of 'hallowell'?

TRANSLATION:
D CAPACITY.CAPACITY,CAPACITY.FUELTYPE FOR TERMINAL.OPNAME='MOBIL OIL CORP', TERMINAL.CITY='HALLOWELL';

CAPACITY.CAPACITY CAPACITY.FUELTYPE
17814 REGULAR GAS
18327 KEROSENE

DISPLAY COMPLETE.

who are the terminal supervisors and what are their telephone numbers and addresses in the city of 'hallowell'?

TRANSLATION:
D TERMINAL.SUPNAME,TERMINAL.SUPPHONE,TERMINAL.SUPADDR FOR TERMINAL.CITY='HALLOWELL';

TERMINAL.SUPNAME TERMINAL.SUPPHONE
ROBERT F CRESSEY 203623873

TERMINAL.SUPADDR
197 CONY STREET

DISPLAY COMPLETE.

Figure 2—Example of computer dialogue
**NEEMIS high-level query facility**

Figure 2 shows an example of the type of query that can be used at this level. For purposes of illustration, we have shown how the requests are translated into DML and passed to that level for further handling. (‘D’ is an abbreviation for “DISPLAY”.)

**NEEMIS interactive query facility**

The user of this facility simply points to a question category he wants answered on a CRT using a “light pen”. If the question needs further specification, the system will flash subsequent choices on the scope, and the user will point to the choice that clarifies his query.

**Prepared packages**

Users of this facility will request standard reports or invoke common models, for example, a monthly forecasting model. All the user at this level needs to know is the name of the report or model. The system will take care of retrieving the requisite data and invoking the appropriate facility to generate a report or run a model.

**NOTES ON IMPLEMENTATION**

The purpose of this paper was primarily to describe the hierarchy of user facilities in NEEMIS as opposed to a description of the implementation of the GMIS. However, there are a number of interesting implementation-related points that bear mentioning.

**Extensions of the relational model**

Just as the user-view of NEEMIS described levels of differing power and flexibility, so the actual implementation of the system was carried out. Software developed for the GMIS has been implemented as a multi-level hierarchy in which each level employs only those facilities implemented in the levels below it. Early explanations and applications of this approach may be found in References 12, 13, and 14.

The GMIS in which NEEMIS is built has paid extensive heed to security of data. Some nineteen types of access have been identified and any owner of data may authorize any user to access those data in any or all of those nineteen ways. The default authority is NO access, rather than the usual approach that allows full access unless otherwise specified. These security specifications are made via facilities in the DML directly.

The relation used to store access control information, as well as all other system relations and descriptors are identical to and accessed in an identical manner to regular user data. Thus all data stored in the system is stored in a consistent fashion making security checking, as well as access consistent for any and all data.

Finally, imbedded in the system code are facilities for monitoring program execution for debugging purposes, as well as timing of operations for system tuning. There is also an ability to log all requests made in the DML and DDL, used mainly for determining optimal data base structure. These facilities may be turned on or off in the DML.

A detailed description of the levels of implementation of the GMIS may be found in Reference 1.

**Role of VM/370**

The capability of running multiple virtual machines at the same time under IBM’s Virtual Machine Facility/370 (VM/370)16 has facilitated a solution to the problem of using NEEMIS as a multiple access system, with different users having varying applications requirements (e.g., report generation, econometric modeling).

In the multiple user environment, the basic requirements for a user are to send a command to NEEMIS, receive a reply that may be in a number of forms (report, single answer, return code) depending on the command, and then either displaying the reply or performing further operations on it.

These requirements are satisfied by using a single virtual machine that contains the NEEMIS data base and command processor. Each user has his own virtual machine, and communicates with the NEEMIS machine through the use of virtual card punches and shared query/reply files. User requests to the NEEMIS machine are stacked in its virtual card reader and are selected one at a time for processing. The NEEMIS machine writes the results of each request in the user’s reply file, and then processes the next user in the queue on a FIFO basis.

Each user is thus provided with a reply file that can be processed by programs written in any language. Currently, programs for flexible report generation have been written in PL/I, and an econometric modeling interface that operates in an APL environment will be implemented.

Using this facility, each user can tailor his interface to NEEMIS to suit his own needs. For example, it is possible to interface TROLL, a popular econometric modeling package, to NEEMIS using programs to convert NEEMIS reply files to TROLL compatible input files.

In summary, the use of multiple virtual machines facilitates increased user isolation and security,17 multiple access to a shared data base without loss of integrity, and the capability of running many different user-dependent application interfaces simultaneously.

**CONCLUSION**

We have presented here a brief overview of some of the user facilities that have been made available in the NEEMIS System. These facilities have been designed
with maximum flexibility and for a wide range of users in terms of both computer sophistication and type of function they perform.

ACKNOWLEDGMENTS

We acknowledge the contributions of Professor Henry D. Jacoby of the Sloan School, M.I.T., for his experience and guidance in the energy policy area.

We would also like to thank Drs. Stuart Greenberg, Paul Comba, and Ray Fessel of the IBM Cambridge Scientific Center for their insight and thoughts, specifically, Paul Comba for his guidance in preserving the mathematical and relational model of data in our DDL and DML, Ray Fessel for his ingenious programming guidance, and Stu Greenberg for his help with the VM concepts of implementation.

Since the writing of this paper, we acknowledge the contributions of members of the IBM Research Laboratory of San Jose who have greatly enhanced the operational aspect of NEEMIS, and we look forward to working with them in the future.

Work reported herein was supported in part by the New England Regional Commission (NERCOM), Boston, Massachusetts.

REFERENCES


