AN APPLICATION OF A GENERALIZED MANAGEMENT INFORMATION SYSTEM TO ENERGY POLICY AND DECISION MAKING: THE USER'S VIEW

John J. Donovan
Louis M. Gutentag
Stuart E. Madnick
Grant N. Smith

May 1975

Energy Lab in Association
with Sloan School
Report No. MIT-EL-75-008

ABSTRACT

This paper presents an approach to the development and use of management information systems that is particularly applicable to systems with the following characteristics:

- several classes of users, each of which has a different degree of sophistication
- complex and changing security requirements
- data exhibits complex and changing inter-relationships
- changing needs to be met by information system
- must be built quickly and inexpensively
- complex data validation requirements

The approach is hierarchical from the user's view in that he may access the system at distinct levels, corresponding to his degree of computer sophistication. A casual user has high level primitives to work with, while an experienced user has more flexible but more detailed low-level primitives.

We also have advocated that such systems be implemented in a hierarchical fashion, because this technique provides for ease of debugging, independence of hardware, and a basis for investigating properties of completeness, integrity, correctness, and performance.

1. 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 recore of how things are done. When events occur that call for government efforts to invluence 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.

2. 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 details of the precise syntax of intermediate languages and implementation details are described elsewhere [1].

In the NEEMIS facility, we wish to give users increasingly more powerful tools, Figure 1 depicts four classes of users as follows:

FIGURE 1

NEEMIS - FUNCTIONS AVAILABLE IN EACH INTERFACE FACILITY.

INTERFACE FACILITY USER CATEGORY	PREPARED PACKAGES	NEEMIS INTERACTIVE QUERY FACILITY	NEEMIS HIGH LEVEL QUERY LANGUAGE	MODELING FACILITY	RELATIONAL QUERY LANGUAGE (DML)	DATA DEFINITION FACILITY (DDL)	RELATIONAL OPERATORS & PL/1 FACILITY
NON-TECHNICAL (No computer training)	X	X	X				
WELL TRAINED	X	X	X	X	X		
RESEARCHER	X	X	X	X	X	X	
SYSTEMS ANALYST & PROGRAMMERS	X	X	X	X	X	X	X

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, 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.

2.1 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 all system data, all access

* For our purposes, can be thought of as matrix of values; each column a domain, each row an entry. See [2] for more details.

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 [2,3], which have their roots in logical systems and predicate calculus [4, 5, 6, 7]. The operators available in NEEMIS are described in detail in [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.

2.2 Data Definition Facility

A user at this level has facilities to specify and create tables. 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:    .
user:      employee (name, soc_sec_#, address)
              (primary key: soc_sec_#),
           fuel_data (soc_sec_#, name, supplier)
              (primary key:  soc_sec_#, required:  supplier);
system:    RELATIONS DEFINED
           ENTER COMMAND
user:      define synonym:  soc_sec_# = 'ss';
system:    SYNONYM ENTERED
           ENTER COMMAND
user:      stop.
```

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

2.3 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. [18]. Other attempts at implementing a query facility based on the relational model include: MACAIMS [8], SEQUEL [9], COLARD [10], RIL [11], and M.I.T.'s RDMS.

This facility is available for querying relations established via the DDL (see 2.2) or possibly the relational operator facility (see 2.1).

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, FUELAMT, 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 name and affiliation 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"!

2.4 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 additional 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 that stored in the relations (see 2.3).

This APL-oriented modeling facility is the current standard. Inclusion of additional or different modeling languages, however, poses little problem (see 3.2 below).

2.5 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".)

2.6 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.

2.7 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.

```
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 fueltypes 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
adresses in the city of 'hallowell'?

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

TERMINAL.SUPNAME                        TERMINAL.SUPPHONE

ROBERT F CRESSEY                        2036233873


TERMINAL.SUPADDR

197 CONY STREET


DISPLAY COMPLETE.
```

FIGURE 2.

EXAMPLE OF COMPUTER DIALOGUE

3. 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.

3.1 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 [12, 13, 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 that 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 infomration, as well as all other system relations and descriptors are identical to 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 [1].

The capability of running multiple virtual machines at the same time under IBM's Virtual Machine Facility/370 (VM/370)[15] has faciliated a solution to the problem of using NEEMIS as a multiple access system, with different users having varying applications requirements (e.g., report generation, economietric 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 [16], 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.

ACKNOWLEDGEMENTS

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 Greenburg, 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 ingeneous 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

1. Donovan, John J. and Henry D. Jacoby: "A Hierarchical Approach to Information System Design," Report CISR-5, M.I.T. Sloan School Working Paper 762-75, Janaury, 1975.

2. Codd, E. F.: "A Relational Model of Data for Large Shared Data Banks," CACM, Vol. 13, No. 6, June, 1970, p. 377-387.

3. Codd, E. F.: "A Data Base Sublanguage Founded on the Relational Calculus," PROCEEDINGS 1971 ACM/SIGFIDET Workshop.

4. Post, E. L.: "Formal Reductions of the General Combinatorial Decision Problem," AMERICAN JOURNAL OF MATHEMATICS, Vol. 65, 1943, p. 197-215.

5. Church, A.: "The Calculi of Lambda-Conversion," ANNALS OF MATHEMATICS STUDIES, No. 6, Princeton University Press, 1941.

6. Smullyan, R.: "Theory of Formal Systems," Study 47, Princeton University Press, 1961.

7. Donovan, John and Henry Ledgard: "A Formal System for the Specification of the Syntax and Translation of Computer Language," PROCEEDINGS 1967 FJCC.

8. Goldstein, I. & Alois Strnad: "The MACAIMS Data Management System," M.I.T. Project MAC TM-24, April, 1971.

9. Chamberlin, D. D. and R. F. Boyce: "SEQUEL: A Structured English Query Language," PROCEEDINGS 1974 ACM/SIGFIDET Workshop.

10. Bracchi, G. et al.: "A Language for a Relational Data Base Management System," PROCEEDINGS, 5th PRINCETON CONFERENCE ON INFORMATION SCIENCE, 1972.

11. Fehher, P. C.: "The Representation of Independent Language," IBM Technical Report RJ1121, November, 1972.

12. Dijkstra, E.: "T.H.E. Multiprogramming System," CACM, May, 1968.

13. Madnick, S. E.: "Design Strategies for File Systems," M.I.T. Project MAC TR-78, October, 1970.

14. Donovan, John J.: SYSTEMS PROGRAMMING, McGraw-Hill, New York, 1972.

15. IBM Virtual Machine Facility/370: Introduction, No. GC20-1813, IBM, Burlington, Mass.

16. TROLL REFERENCE MANUAL, National Bureau of Economic Research, 1972.

17. Donovan, John J. and Stuart E. Madnick, "Application and Analysis of the Virtual Machine Approach to Computer System Security and Reliability," M.I.T. Sloan School Report CISR-2, May, 1974.(to be published in the IBM SYSTEMS JOURNAL, May, 1975).

18. Smith, Grant N., "Internal Intermediate Language, Version 2", M.I.T. Sloan School Report CISR-6, November, 1974.