MIT/LCS/TM-22

TRANSMISSION OF INFORMATION BETWEEN A MAN-MACHINE DECISION SYSTEM AND ITS ENVIRONMENT

Douglas M. Wells

April 1971

TRANSMISSION OF INFORMATION BETWEEN A MAN-MACHINE DECISION SYSTEM AND ITS ENVIRONMENT

Technical Memorandum 22

Douglas M. Wells

April 1971

PROJECT MAC

Massachusetts Institute of Technology

Cambridge Massachusetts 02139

Work reported herein was supported in part by Project MAC, an M.I.T. research project sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract N00014-69-A-0276-0002.

1. INTRODUCTION

The MacAIMS (MAC Advanced Interactive Management System) project is charged with the task of developing a computer resource to assist in making non-trivial decisions. The project has previously created and maintained a package of interactive programs for use in administering a research organization. Based upon the experience acquired during the implementation and subsequent evaluation of these programs, the project began the design and development of a second, much more powerful information system particularly aimed at assisting in the solution of managerial level problems.

Based upon our experience with earlier systems and an investigation of the coherence of other information system implementation efforts, we set forth the basic design principle that in order to allow and control the complex inter-relationships among vast amounts of data necessary to support non-trivial decisions; simple, logical modules must be sought out and defined throughout the system. Furthermore, we observed that comprehension of the interworkings of a man-machine system is improved through the specification and use of canonical forms for both internal and external interfaces. The purpose of this paper is to report upon our employment of such functional modules and the definition of canonical interfaces at the most observable level -- the overall system specification.

2. OVERVIEW

The implementation of the MacAIMS System is divided into three functional task areas: data base management, man-machine interaction, and modeling. Because it has been our feeling that the first two of these problem areas are general in nature, we have attempted to define, implement, and utilize within these areas programs in which application dependencies reside in an associated data base. Modeling, on the other hand, includes a very large proportion of problems to which general solutions are unknown. For this reason, the level of utilization of high level canonicalization is higher within the areas of data manipulation and man-machine interaction. For this reason also, however, we maintain a forced level of standard form specification in order to obtain the improved comprehension that such specification permits.

Because manipulation of the data base is the foundation and *raison d'etre* for the existence of an information system, the design of the data base management system performs an important role in defining the canonical forms for stored data and the inter-relationships among that data throughout the system. To aid in our own comprehension of the system's functioning, several stages of canonicalization are employed within the data management system. Of interest in the context of this paper are the canonical form for stored data elements and the canonical form for the relations that exist among those data elements. The canonical form for data elements is a fixed-length (36 bits in our implementation) reference number which represents a one-to-one mapping from the written representation of the external world. The canonical form for relations is a set-theoretic relation set consisting of n-tuples of such reference numbers. The implications of this approach and a further description are discussed in an associated paper describing the MADAM (MacAIMS Data Management) System (1).

In considering communication with the MacAIMS System, we have found it useful to distinguish four areas of interest:

Data Collection - the bulk acquisition by the system of previously unknown data elements and/or relation sets.

Process Management - the specification of processing operations and information flow within the system.

Report Generation - the production of a virtual formatted document created from internal data for use external to the system.

Facilities Management - the control and utilization of the various input-output devices and pseudo-devices available to the system user.

Through the use of this separation of external interfaces, we have been able to better understand the requirements of each particular interface. Also because of this specification of distinct interfaces, we have been able to design and implement these external interfaces simultaneously.

3. DATA COLLECTION

Data Collection is the acquisition of bulk data and the subsequent conversion of that data into a form suitable for use within the system. A data collection system must accept the proffered data, perform appropriate validity checks, and then transform the resultant data into the appropriate relation sets and data elements. In defining the implementation of such a mechanism, we find it useful to categorize the input into three classes denoting the primary purpose of the information source of the data:

External Data Sources - systems generating data expressly for transmission to MacAIMS and "Other Information Systems."

Other Information Systems - other computer systems generating data for MacAIMS as a by-product of its primary function of processing information for local use.

Data Entry - facilities for entering data directly into the MacAIMS System through the use of an interactive interface.

Because of the existence of a canonical form for data presented to the data management system, the process of data collection has become a problem separable from the implementation of other parts of the system. Any particular data collection system need only concern itself with producing <u>any</u> relation set(s) that correctly represents the information content of the input data.

The processing of data from a particular "External Data Source" such as a stock ticker or a news service teletype will in general be the easiest collection facility for which to specify an interface because the data is originated with the express purpose of transmission of that data. As such the form of the input data will normally be both consise and consistent. Because such data would generally be received and processed independently of any particular user, however, the mechanism for reception of such data must directly interface the operating system upon which MacAIMS is implemented in order to permit the asynchronous processing of such data. The facility should also incorporate means for recovery of data not processed because of system inavailability. This would normally mean the implementation of separate computing facilities, such as a mini-computer terminal, for the buffering of such data.

The format of input transmitted from "Other Information Systems" will vary according to the conventions of that system. Because the internal data structure format of any other system will generally be quite complex, we envision the necessity of programmers to implement data collection facilities to interface data transmitted in the internal formats of other systems. However, the reports produced by such systems offer a brighter prospect. Using concepts learned during the implementation of FORGE (see Report Generation below), we hope to perform the reverse function of transforming an outputted report into the required internal relation sets as an on-line, user-definable task. With this facility a user could enter the received report through either card-images or optical character recognition and then create the appropriate relation sets by generating a FORGE description of that report. In any case it should be noticed that the organization of any such received reports will normally reflect a close approximation to the structure of such relation sets as are derivable from the input.

Because the design of any "Data Entry" facility should be undertaken with the objective of total data verification and a concern for ease of use by data entry personnel, the implementation of a "Data Entry" facility will almost certainly be a task for a programmer who understands the concepts and implications of man-machine interaction. In the design of such a facility, the programmer must produce an environment in which the data entry personnel are substantially aided in entering verified data and which provides useful tools and checks for them. Because we believe that an on-line immediate response system can substantially aid in monitoring the progress of any operation, it is expected that the use of such data entry would be by normal personnel who would use the system to keep track of the required activity.

4. PROCESS MANAGEMENT

In order to utilize the data that is stored within an information system, some operations and manipulations must be performed upon that data in order to transform it into a structure that represents a response to some inquiry. Because the exact sequence and ordering of these operations will, in general, not be known in advance, the need for a programming and/or command language for controlling the process arises. To avoid confusion within the system between these two types of language we have chosen to consider a command language as an input to an environment in which an interpreter converts all commands into equivalents of standard calls in a programming language and utilizes the interaction factor to provide monitoring of all exceptional and unforeseen conditions. The term "interaction factor" is used here to signify all manner of user-control, including such actions as determining which commands to execute, deciding which relation-sets to use as input, performing checks on data validity, and providing such other controls as might be specified and allowed by the system.

The specification of the canonical form of a process management language has been particularly straight-forward in MacAIMS due to the design of the MADAM (MacAIMS Data Management) System. Because MADAM is based upon a set-theoretic approach, the canonical form of data base manipulation instructions consists of all set operation primitives plus those additional primitives implemented to specify a desired ordering and to allow accessing of the individual elements of relation sets. The canonical form of any operation consists of the primitive and its operands -- exactly the invocation and stipulation of associated arguments of the MADAM subroutine that implements that primitive operation.

Through the judicious use of this specification, we have been able to isolate the specification of the syntax of a command language. Thus it is possible to make the specification of the syntax and even the semantics a separable issue. It is possible to implement any language for which a transformation from acceptable syntax to this canonical form can be defined. Because of this isolation of the characteristics of the process control language, the form and structure of the language can be tailored to the desires of the user of the system. Furthermore, as advances are made in the field of natural language processing, the results can be implemented without necessitating internal changes to the other functional areas of MacAIMS.

For the decision-maker who is cognizant of a computer programming language, the existance of the set primitive operations for data manipulation provides a particularly powerful, yet simple repertoire of operations for manipulation of data. It is expected that interfaces for several common programming languages will be implemented to encourage the use of such programming languages gy decision-makers. The interface for any particular language will need to be defined for that language and will consist of interfacing routines capable of properly interpreting and converting the language data types to the data types of PL/1 (which is the language used in the implementation of MacAIMS). Through the use of such routines, the full facilities of MacAIMS will be made available from that programming language.

5. REPORT GENERATION

The term report is used to denote any output of the system, whether that result be a one sentence answer to a simple question, a three-dimensional plot, or a multi-volume inventory list. Although the MacAIMS system is designed to reduce the number of large reports necessary, we recognize that the production of such reports is a primary function of any information system (at least for the near future) and that the production and preparation of such reports is a large function of such a system.

Within MacAIMS report generation is defined to be the operation of creating a virtual "Page" (see Facilities Management below) containing a human readable representation of the contents of a relation set. To provide the utmost in on-line system flexibility and to eliminate the need for a constantly available system programmer to design and implement such reports generation routines, considerable effort has been expended in attempting to define a report in terms of elements and attributes of those elements, and to allow the on-line description and formulation of such a report.

FORGE (<u>Fo</u>rmatted <u>R</u>eport <u>Ge</u>nerator) is the system used to specify and to generate the reports. FORGE is based on an observation that a formatted report tends to be organized in a tree-like structure of elements. Each leaf-element must then be an attribute, a position specifier, a textual element, or a graphic element. By the choice of a full set of such leaf-elements one can generate a formatted report of arbitrary complexity. Because most output mediums that exist in the real world are of a finite size, FORGE provides attributes that define the mapping function to convert a report of unknown dimensions onto pages of known, finite dimensions. The attributes for controlling this second mechanism constitute the other major grouping of attributes.

Here again, the canonicalization defined by the data management and facilities management systems are utilized to ease the implementation and specification of the FORGE system. Because all data manipulation (sorting, computation, etc.) is expected to be executed by other components of the MacAIMS System, the attributes required by FORGE are limited to specifications of placement and structure.

6. FACILITIES MANAGEMENT

The word facilities is used here to denote those devices (or pseudo-devices) that provide mechanisms for input-output with the user. Typical facilities which will be provided in the MacAIMS System are a visual screen, a typewriter-like keyboard, a function keyboard, hard-copy devices, etc.

The use of input-output facilities has been considerably enhanced through the adoption of a standard interface for all such devices. Thus through proper substitution of internal transformation tables, it is possible to redirect any input or output stream (in the PL/1 sense of the term). Furthermore, the implementation and specification of additional devices is made considerably simpler due to the standard interface.

To aid in the comprehension of implications of changes and design specifications for the management of the facilities, we have adopted the conventions of Sutherland and Sproull (2) for describing the transformation from FORGE output to tangible output. The three concepts accepted in particular are:

Page - a virtual image of the item being considered for input and/or output (possibly three-dimensional).

Window - that section of the "Page" under immediate consideration.

Viewport - the physical representation of a transformation of the "Window.".

Through the use of these concepts, transformations and superpositions can be used to manipulate FORGE output into any desired arrangement. It is expected that standard forms will again be defined for use in this area so that virtual device-independence may be achieved. Thus a user could experiment with various FORGE specifications until he discovered a satisfactory report format. He could then receive an identical looking hard-copy representation of that report by a simple redirection of the output stream from the visual screen to the hard-copy device.

ACKNOWLEDGEMENTS

I would like to express my appreciation to Mr. Robert C. Goldstein for his comments on this paper and especially for his continual suggestions of concepts and goals within the area of the man-machine interface.

REFERENCES

1) Strnad, Alois J., "The Relational Approach to the Mangement of Data Bases," Proc. IFIP Congress (1971).

2) Sproull, Robert F., and Sutherland, Ivan E., "A Clipping Divider," Proc. FJCC (1968), pp 765-775.